



# Graph transformations expressed in logic and applications to structural graph theory

*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>

*History* : Four independent research directions now intimately related :

1. **Polynomial algorithms** for NP-complete and other hard problems on particular classes of graphs, and especially hierarchically structured ones : series-parallel graphs, cographs, partial k-trees, graphs or hypergraphs of tree-width  $< k$ , graphs of clique-width  $< k$ .
2. **Excluded minors** and related notions of forbidden configurations (matroid minors, « vertex-minors »).
3. **Decidability of Monadic Second-Order logic** on classes of finite graphs, and on infinite graphs.
4. Extension to graphs and hypergraphs of the main concepts of **Formal Language Theory** : grammars, recognizability, transductions, decidability questions.

Two key words :

**Graph structure** (main notions) :

hierarchical decompositions (tree-decomposition, modular decomposition,...)

embedding on surfaces

exclusion of a minor, a **vertex-minor** or an induced subgraph

existence of homomorphism into a fixed graph (generalized coloring)

**Logic** : First-order, second-order, **monadic second-order (MS)**

for expressing graph properties (i.e., defining graph classes)

for defining **graph transformations**, and structures of above types

## Graph structure and monadic second-order logic are related in many ways

- 1) Monadic Second-order (MS) logic and hierarchical decompositions yield **fixed-parameter tractable** algorithms (for **tree-width** and **clique-width / rank-width** ).
- 2) Decidability of MS logic implies **bounded** tree-width or clique-width.
- 3) Planarity, tree-width  $< k$ , **exclusion of minors or vertex-minors** are MS properties
- 4) Modular decompositions, planar embeddings can be **defined by MS formulas**.
- 5) Tree-width bounded and clique-width bounded classes have characterizations in terms of images of **trees under MS graph transformations** that are independent of the initial combinatorial or algebraic definitions :  
this shows the robustness of definitions and yields useful stability properties under MS graph transformations.

## Summary

1. Logical expression of graph properties by MS formulas : graph minors and related properties.
2. **MS transductions** : Graph transformations specified by MS formulas.
3. Characterization of tree-width and clique-width bounded graph classes in terms of trees and MS transductions.
4. **New result** : Comparing *encoding powers* of graph classes via MS transductions. (*Uses graph minors*)
5. Decidability of monadic second-order (**MS**) logic is strongly connected with bounded tree-width or bounded rank-width (hence bounded clique-width). (*Uses minors and vertex-minors*).

# 1. Monadic Second-Order (MS) Logic

= First-order logic on power-set structures

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

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

*3-colorability :*

Let  $\varphi$  be the formula

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

$G$  is 3-colorable  $\Leftrightarrow G \models \varphi$  (read :  $\varphi$  is true in  $G$  )

For each  $k$ , a formula expressing  $k$ -colorability can be constructed.

*Non connectivity (whence connectivity by negation, and transitive closure) :*

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

One can write a formula  $\gamma(Y)$  with a free set variable  $Y$  such that, for every set of vertices  $Y$  of  $G$  :

$$G \models \gamma(Y) \iff \text{the subgraph of } G \text{ induced on } Y \text{ is connected}$$

For each simple and loop-free graph  $H$  one can write a formula  $\mu_H$  such that

$$G \models \mu_H \iff G \text{ contains } H \text{ as a minor.}$$

*From the Graph Minor Theorem :*

Every minor-closed class of graphs is characterized by a monadic second-order formula (the obstruction set must be known for an effective construction).

Edge set quantifications increase the expressive power

*Incidence graph of*  $G$  undirected,  $\text{Inc}(G) = (V_G \cup E_G, \text{inc}_G(\cdot, \cdot))$

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

Monadic second-order ( $\text{MS}_2$ ) formulas written with  $\text{inc}$  can use quantifications on sets of edges.

The existence of a perfect matching or a Hamiltonian circuit is expressible by an  $\text{MS}_2$  formula, but **not by an MS** formula.

Facts :  $\text{MS}_2$  formula are more expressive than **MS** formulas

$\text{MS}_2$  logic fits well with bounded **tree-width**

**MS** logic fits well with bounded **clique-width / rank-width**



## Why is MS logic interesting ?

- 1) Every  $MS_2$  (**MS**) property has an  $O(n)$  ( $O(n^3)$ ) FPT algorithm for parameter **tree-width** (**clique-width / rank-width**)  
( $n$  = number of vertices)
- 2) If a class of graphs is minor-closed and if we know an  $MS_2$  definition NOT using excluded minors, **and** a bound on the tree-widths of its obstructions, then the set of obstructions is computable (*but not practically !*).

- Proof sketch* :
- $G(X,Y)$  = the minor of  $G$  obtained by contracting the edges of  $X$  and deleting vertices and edges of  $Y$  ;
  - $\mu(X,Y,u,v)$  :  $MS_2$  formula expressing that  $u$  and  $v$  are adjacent in  $G(X,Y)$  ;
  - If a minor-closed class  $\mathcal{C}$  is defined by an  $MS_2$  formula  $\psi$  , then  $\text{Obst}(\mathcal{C})$  is defined by the  $MS_2$  formula  $\theta$  expressing :
    - $\psi$  is false and for every  $X,Y$  with  $X \cup Y$  not empty, the minor  $G(X,Y)$  of  $G$  satisfies formula  $\psi$  (which can be written with the help of  $\mu(X,Y,u,v)$  ) ;
  - Since  $\text{Obst}(\mathcal{C})$  is finite with known bound on tree-width, and defined by an  $MS_2$  formula , it can be enumerated (see next slide)

*Application : Apex graphs over a minor-closed class with known obstructions (Adler, Grohe, Kreutzer 2008).*

**Proposition** : Let  $\psi$  be an  $MS_2$  formula such that, for each  $k$ , the set  $L(k)$  of graphs of **tree-width at most  $k$**  that **satisfy  $\psi$  is finite**.

1) For each  $k$ , the set  $L(k)$  can be enumerated.

2) The maximal size of  $G$  in  $L(k)$  is bounded by  $f(k)$ , where  $f$  is an “elementary” function computable from  $\psi$

(  $f(k) = \exp^h(k)$  , iterated exponential, where  $h$  depends only on  $\psi$  ).

**Proof sketch** : - For each  $k$ , the graphs of tree-width at most  $k$  are defined by algebraic terms written with a finite set  $F_k$  of graph operations (Last slides of this presentation)

- The terms over  $F_k$  that define graphs satisfying  $\psi$  are those recognized by a finite automaton with  **$f(k)$  states**, constructible from  $\psi$  (the key to FPT algorithms for MS properties for parameter tree-width).  $h = O(\text{quantifier-height}(\psi))$ .
- The remaining assertions follow from results of automata theory.

## 2. Monadic second-order transductions

Transformations of graphs specified by MS (or by  $MS_2$ ) formulas.

**Two types** : MS-transductions for graphs =

(vertices, adjacency relation)

$MS_2$ -transductions for graphs represented by their

**incidence graphs** = (vertices *and* edges, incidence relation)

**Results** :

MS-transductions preserve bounded **clique-width**

$MS_2$ -transductions preserve bounded **tree-width**

They preserve the decidability of MS- (or  $MS_2$ -) theories.

## Examples of MS and MS<sub>2</sub> transductions :

$G \longmapsto H$ , its edge complement  $\text{edg}_H(x,y) \Leftrightarrow x \neq y \text{ and } \neg (\text{edg}_G(x,y))$

$G \longmapsto H = G^{(2)}$  defined by :

$$\text{edg}_H(x,y) \Leftrightarrow \text{edg}_G(x,y) \vee \exists z (\text{edg}_G(x,z) \wedge \text{edg}_G(z,y))$$

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

$(G, X, Y) \longmapsto G(X, Y)$  the minor of  $G$  resulting from contracting the edges of  $X$  and deleting the edges and vertices of  $Y$  (MS<sub>2</sub>-transduction).

$G$  planar 3-connected graph  $\longmapsto$  its unique plane embedding  
(formalized as a circular ordering of edges around each vertex).

## Informal definition

MS transductions are multivalued mappings  $\tau$  from relational structures to relational structures

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

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

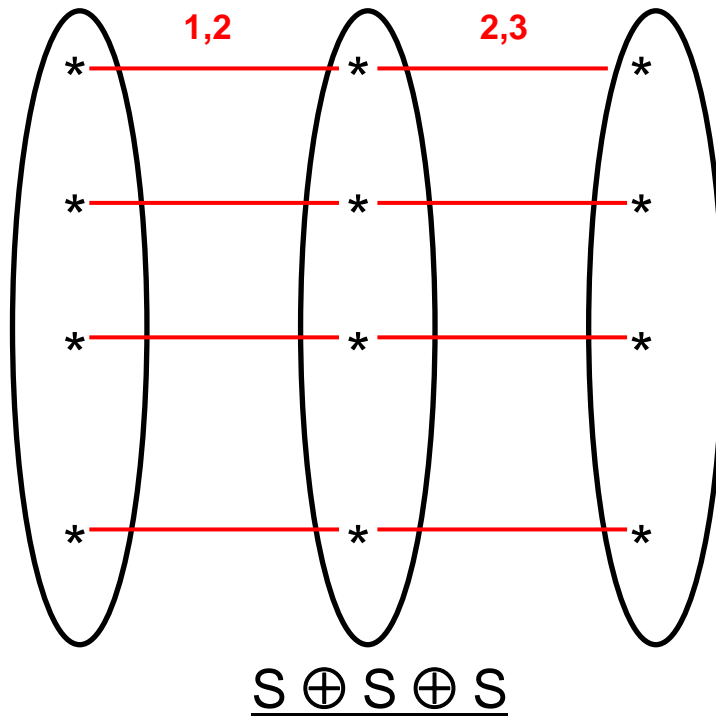
*(Examples of parameters :  $X, Y$  in the definition of  $G(X, Y)$ )*

$\tau$  is multivalued because of parameters

The general definition makes it possible to define  $T$  “larger” than  $S$ .

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.

The fundamental property of MS transductions (“reduction tool”) :

$$S \longmapsto \tau(S)$$

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

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 what is necessary to determine  $\tau(S)$  ; the MS properties of  $\tau(S)$  are expressible by MS formulas in  $S$ ).

**Consequence** : If  $L$  has a **decidable MS theory**, so has its image under an MS transduction.



MS - transductions and  $MS_2$  - transductions are **incomparable**

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

For building graphs with  $MS_2$  - transductions, we have more possibilities on the input graph, but we want to **specify each edge** from some vertex or some edge of the input graph.

**Transitive closure** is an MS-transduction that is **not**  $MS_2$

**Edge subdivision** is an  $MS_2$  - transduction that is **not** MS.

Proofs of negative facts are based on the observation that  
if  $S$  is transformed into  $T$  by an MS-transduction, then :

$$|\text{domain}(T)| = O(|\text{domain}(S)|)$$

### 3. Robustness results : Preservation of “widths” and generation

For every class of graphs  $C$  :

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

*Follows from :*

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

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

*Follows from :*

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

Characterizations in terms of trees and MS logic.

No relation with excluded minors, vertex-minors or induced subgraphs.

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.
- 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).

## How can one prove these results ?      What is clique-width ?

*Idea for* : if  $C$  has bounded tree-width then  $C \subseteq \tau(\text{Trees})$  for some  $MS_2$  – transduction  $\tau$  (*the converse is much more technical*).

1) A graph can be described by the tree  $T$  of some tree-decomposition, with node labels encoding edges and equalities of vertices in the different boxes.

2) For tree-width  $\leq k$ , a bounded number  $N(k)$  of labels is sufficient.

3)  $N(k)$  parameters (sets of nodes of the tree  $T$ ) of an MS transduction can “guess” the possible labellings.

4) MS formulas can “decode” : that is, they can define the graph from  $T$  and its labels.

Clique-width a “more powerful” graph complexity measure than tree-width.

Defined in terms of **simple graph operations** that construct graphs.

Equivalent notion: **rank-width** (Oum and Seymour) with better structural and algorithmic properties (characterization by excluded vertex-minors, exact cubic decomposition algorithm).

Graphs are loop-free, simple, directed or not.

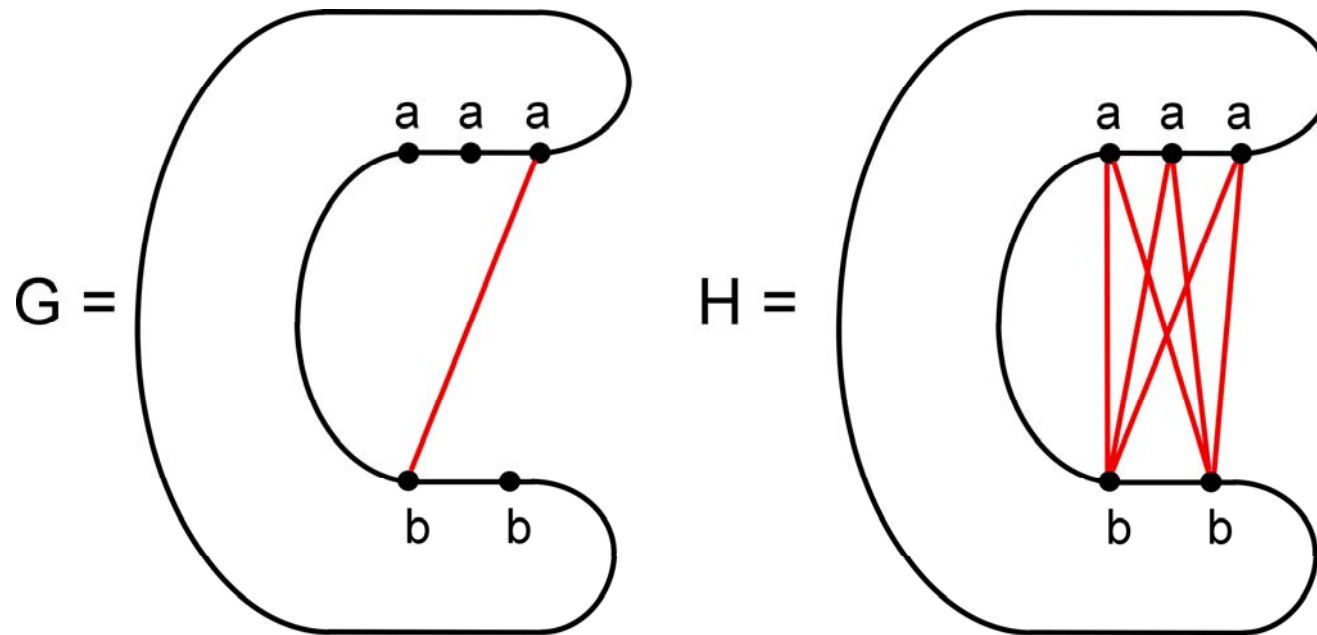
Labels :  $1, \dots, k$ . Each vertex has one label ; labels define a partition of the vertex set.

*One binary operation disjoint union :  $\oplus$*

*Unary operations*: Edge addition denoted by  $Add-edg_{a,b}$

Indexed by labels  $a, b$

$Add-edg_{a,b}(G)$  is  $G$  augmented with undirected edges from every  $a$ -labelled vertex to every  $b$ -labelled vertex.



$H = Add-edg_{a,b}(G)$  ; only 5 new edges added

The number of added edges depends on the argument graph.

The similar operation  $\xrightarrow{\text{Add-edge}_{a,b}}$  adds directed edges from  $a$  to  $b$ .

Vertex relabellings :

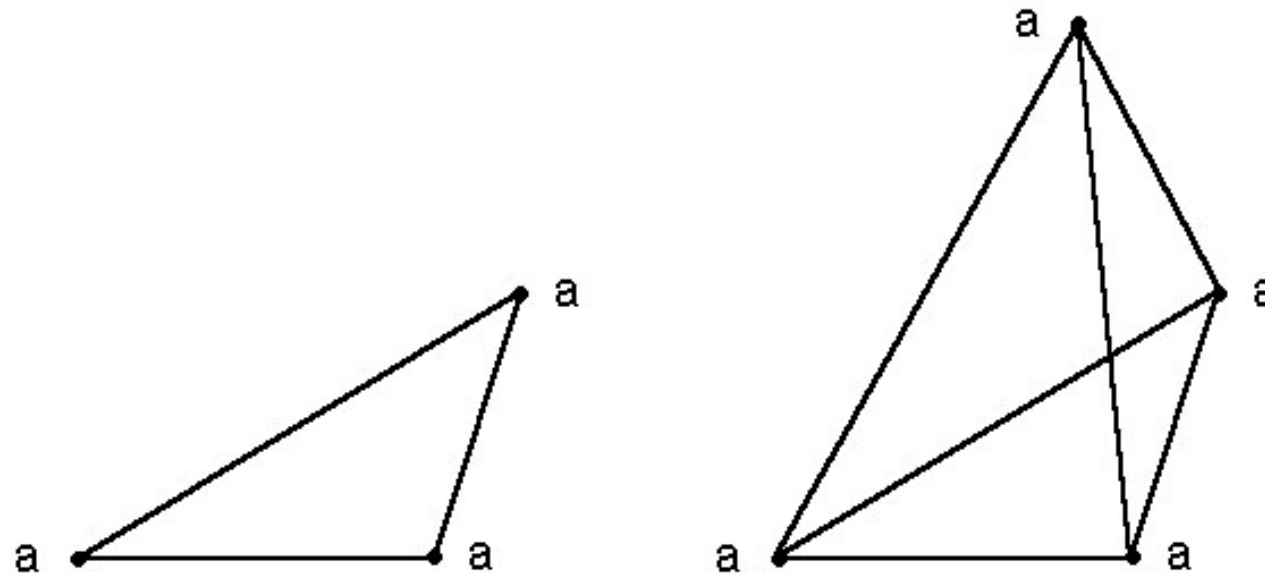
$\text{Relab}_{a \rightarrow b}(G)$  is  $G$  with every vertex labelled by  $a$  relabelled into  $b$

Basic graphs are those with a single vertex.

*Definition:* A graph  $G$  has **clique-width**  $\leq k \Leftrightarrow$  it can be constructed from basic graphs with the operations  $\oplus$ ,  $\text{Add-edge}_{a,b}$  (or the directed variants) and  $\text{Relab}_{a \rightarrow b}$  with labels  $1, \dots, k$ .

Its clique-width  $\text{cwd}(G)$  is the smallest such  $k$ .

Cliques have clique-width 2.



$K_n$  is defined by  $t_n$  where  $t_{n+1} = \text{Relab } b \rightarrow a(\text{Add-edge } a, b(t_n \oplus \mathbf{b}))$

Cographs are generated by  $\oplus$  and by  $\otimes$  defined by :

$$G \otimes H = \text{Relab } b \rightarrow a(\text{Add-edge } a, b(G \oplus \text{Relab } a \rightarrow b(H)))$$

=  $G \oplus H$  with “all edges” between  $G$  and  $H$ .



*Proposition* : (1) Bounded tree-width implies bounded clique-width, but **not conversely**.

(2) Unlike tree-width, clique-width is sensible to edge directions: Cliques have clique-width 2, tournaments have unbounded clique-width.

Classes of unbounded tree-width and **bounded clique-width**:

Cliques (2), Complete bipartite graphs (2), Distance hereditary graphs (3),

Graphs without  $P_5$  and  $1 \otimes P_4$  (5), or  $1 \oplus P_4$  and  $1 \otimes P_4$  (16) as induced subgraphs. (many similar results for exclusion of induced subgraphs with 4 and 5 vertices).

Classes of unbounded clique-width :

Planar graphs of degree 3, Tournaments, Interval graphs,

Graphs without induced  $P_5$ . ( $P_n$  = path with n vertices).

## Logical Characterization of bounded clique-width :

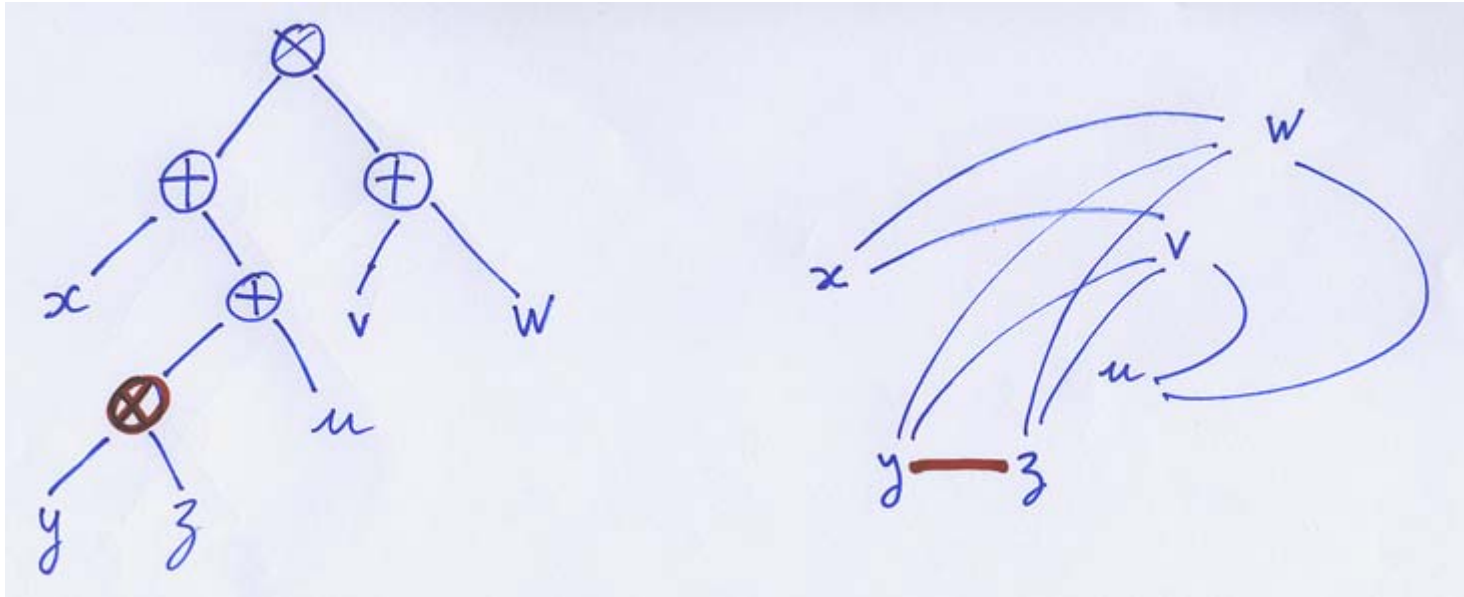
1) A *k-bounded* clique-width term is a rooted binary tree with nodes labelled by the finitely many operations symbols using labels  $1, \dots, k$ .

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  $\tau_k$ .

3) As for tree-width the converse is quite technical (both proofs use the same logical tools).

Proof idea of 2) for  $k=2$ , i.e., for **cographs**, defined by terms written with  $\oplus$  (disjoint union),  $\otimes$  (complete join) constants denoting vertices.



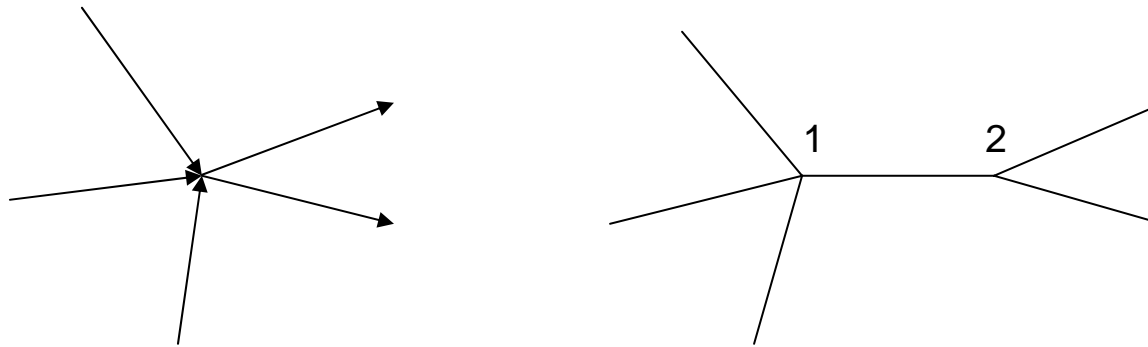
Vertices =  $\{x,y,z,u,v,w\}$  = occurrences of constants in the term

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.

# Application : Encoding a directed graph $G$ by a vertex-labelled undirected graph $B(G)$

Each vertex of  $G$  is split into 2 vertices labelled by 1 and 2 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, actually first-order ones, hence they preserve bounded clique-width.)

Algorithms for recognizing rank-width of undirected graphs can be used to build approximation algorithms for clique-width of directed graphs (Oum, Hlineny, Seymour) because rank-width and clique-width are related by fixed functions.

## 4. 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 (also edges) of  $G$ ).

Say  $H$  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 :

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

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

We consider **MS<sub>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</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 can we say about  $<_c$  (the covering relation of  $\leq$ ) ?*

With help of Robertson and Seymour (GM1, GM5) :

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

These classes **encode** respectively (by  $MS_2$  transductions) :  
finite sets,  
sets of graphs of bounded path-width,  
sets of graphs of bounded tree-width,  
all sets of graphs .

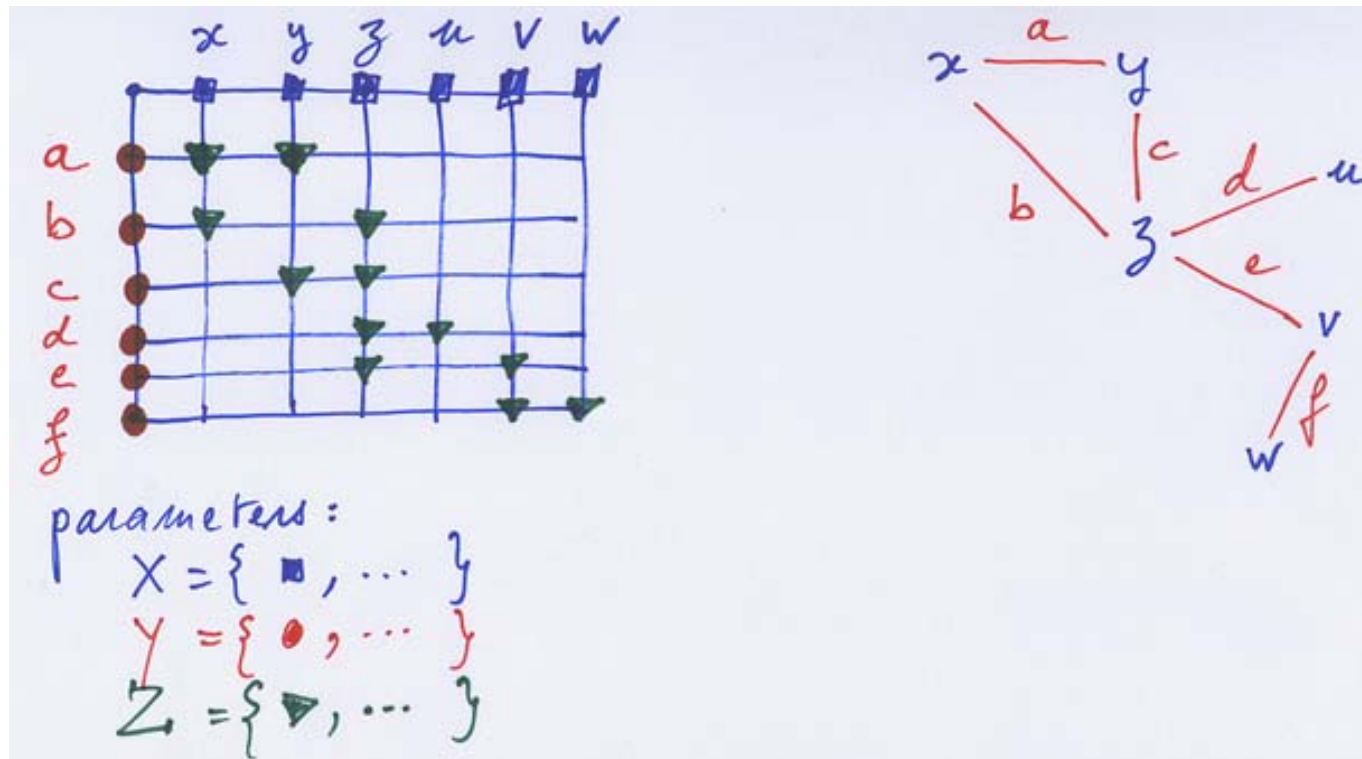
Proof :  $\text{Trees} <_c \text{Grids}$ .

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 and B. C., Logic Colloq. 2008, submitted)

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

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  $k$ ).

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

Clearly :

$$\text{twd}(G) \leq \dots \leq \text{twd}_{n+1}(G) \leq \text{twd}_n(G) \leq \dots \leq \text{twd}_1(G) = |V(G)| - 1$$

**Combinatorial properties of this variant of tree-width.**

- 1)  $\text{pwd}(G) \leq n \cdot \text{twd}_n(G)$
- 2) If G is a minor of H then  $\text{twd}_n(G) \leq \text{twd}_n(H)$
- 3)  $\text{twd}_n(P_m)$  is  $\approx m^{1/n}$  ( $P_m$  = path with m vertices).
- 4) **Excluded Path Theorem** (cf. Excluded Tree and Grid Thms, GM1, GM5)  
A set of graphs C excludes some path as a minor  
if and only if, for some n,  $\text{twd}_n$  is bounded on C

*Questions :*

- 1) What is the minimal integer  $n = N(m)$  such that  $\text{twd}_n$  is bounded on the class of graphs without  $P_m$  as minor ?
- 2) What is the corresponding least upper bound of  $\text{twd}_{N(m)}$  on this class ?
- 3) How can one compute  $n$ -depth tree-width ?

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

**Proposition** : For each  $n$  and  $k$ , there exists an  $MS_2$ -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 1** : For tree-width, there is a difficult (unpublished), weaker result : For each  $k$  there is an  $MS_2$ -transduction that maps every graph of tree-width  $\leq k$  *to one of its* tree-decompositions of width at most  $k$ . (Impossible to get *all* strict decompositions).

**Remark 2** : The obstruction set of graphs for  $n$ -depth tree-width  $\leq k$  is computable from  $k$  (because we have a monadic second-order characterization and a bound on the tree-widths of the obstructions).

In the hierarchy :

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

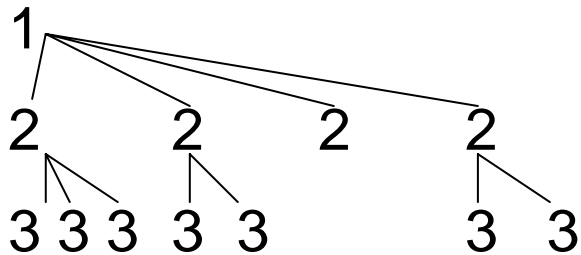
each level  $T_n$  encodes the sets of graphs of bounded  $n$ -depth tree-width.

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 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  $C \leq T_{n-1}$ .

## 5. Graph classes with decidable MS theories (or 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 2004): 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, GM5)
- 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  $MS_2$  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.

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 **decidable  $C_2MS$  satisfiability** problem, so has  $\text{Vertex-Minors}(C)$ , because  $C \longrightarrow \text{Vertex-Minors}(C)$  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  $v$

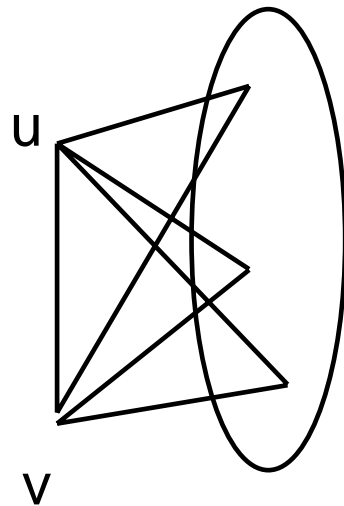
*Local equivalence* ( $\approx_{loc}$ ) = transitive closure of local complementation  
(at all vertices)

*Vertex-minor* relation :

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

*Proposition (Courcelle 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  $X \subseteq Y$  :

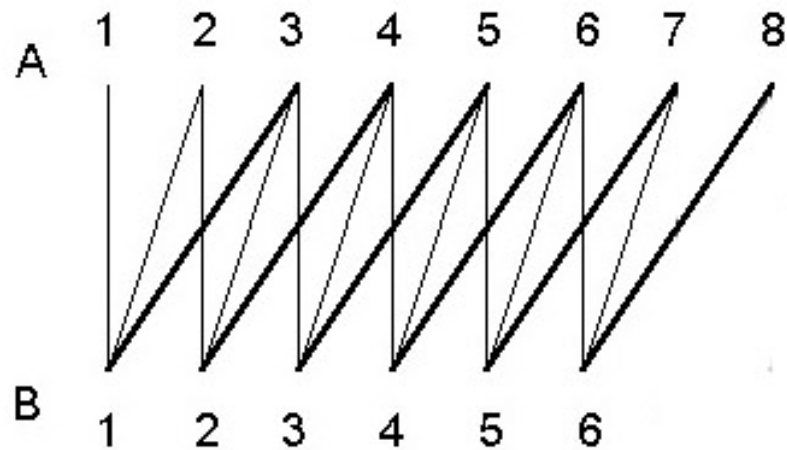
$u$  is linked to  $v$  in  $G * X$

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

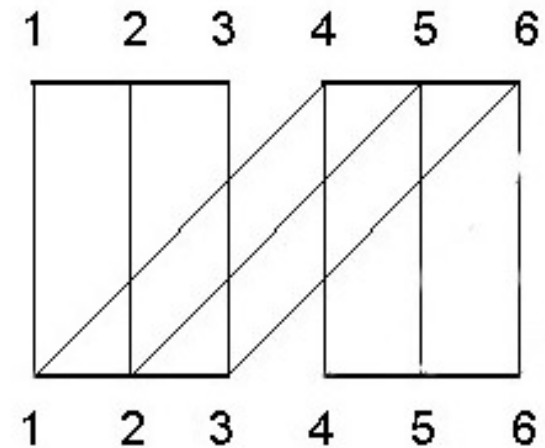
$G$        $Y$

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

## Open questions

**The most difficult one** : Is it true that if a set of relational structures has a decidable CMS-theory, then it is the image of a set of trees under an MS-transduction ? (CMS = MS logic with modulo cardinality set predicates, generalizing Even(X) )

Remark : Some results like the Blatter-Specker Theorem do not extend from graphs to relational structures. (See works by Fischer and Makowsky).

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

Properties of n-depth tree-width.

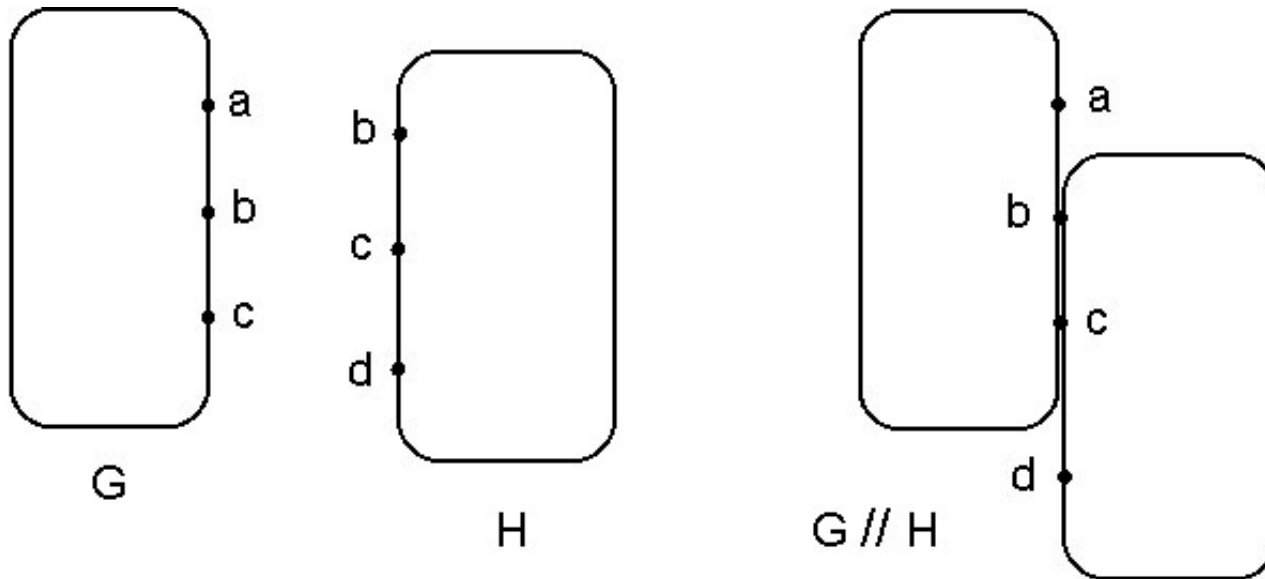
# Bonus : Graph operations characterizing tree-width

Graphs have distinguished vertices called *sources*, (or *terminals* or *boundary vertices*) pointed to by *source labels* from a finite set :  $\{a, b, c, \dots, h\}$ .

*Binary operation(s)* : *Parallel composition*

$G // H$  is the disjoint union of  $G$  and  $H$  and sources with same label are *fused*.

(If  $G$  and  $H$  are not disjoint, one first makes a copy of  $H$  disjoint from  $G$ ).



*Unary operations* : *Forget a source label*

$Forget_a(G)$  is  $G$  without  $a$ -source: the source is no longer distinguished ;  
(it is made "internal").

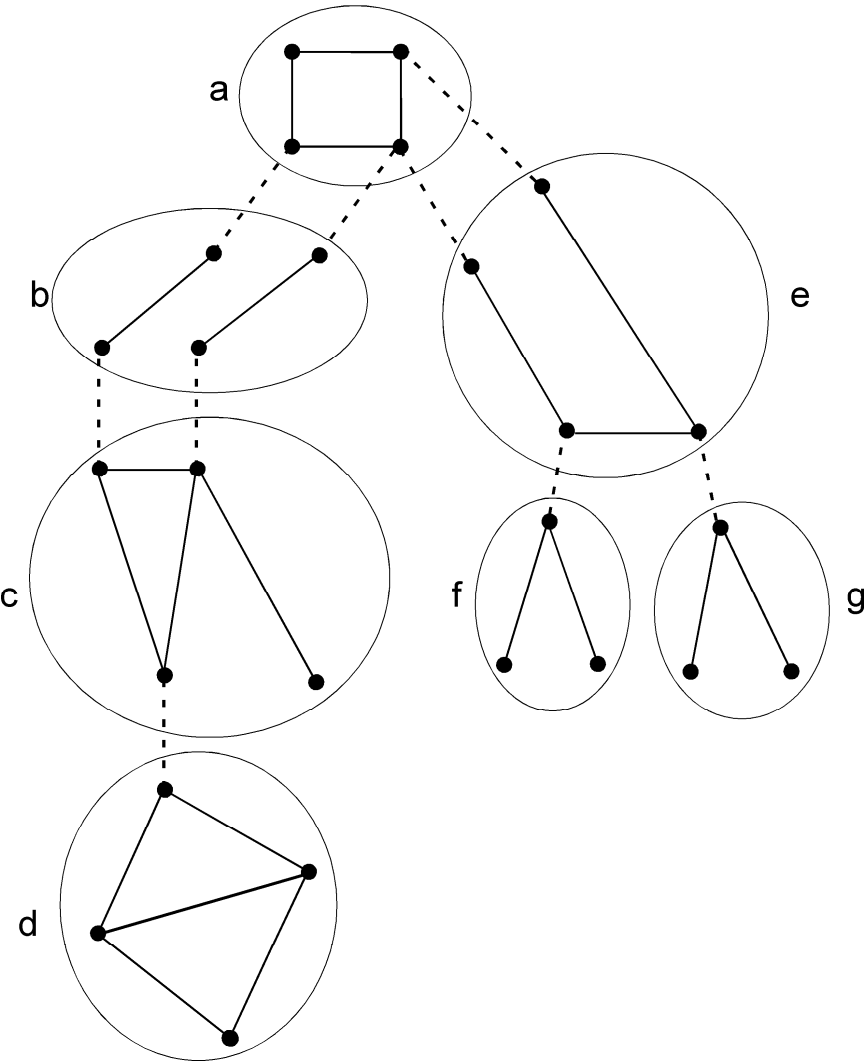
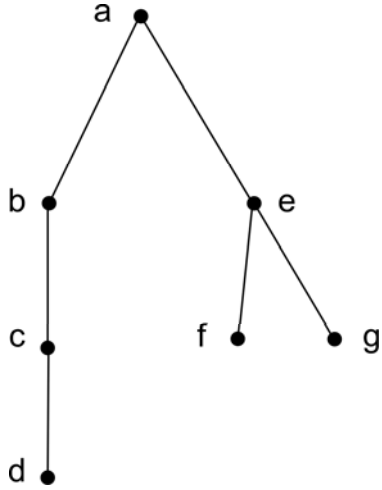
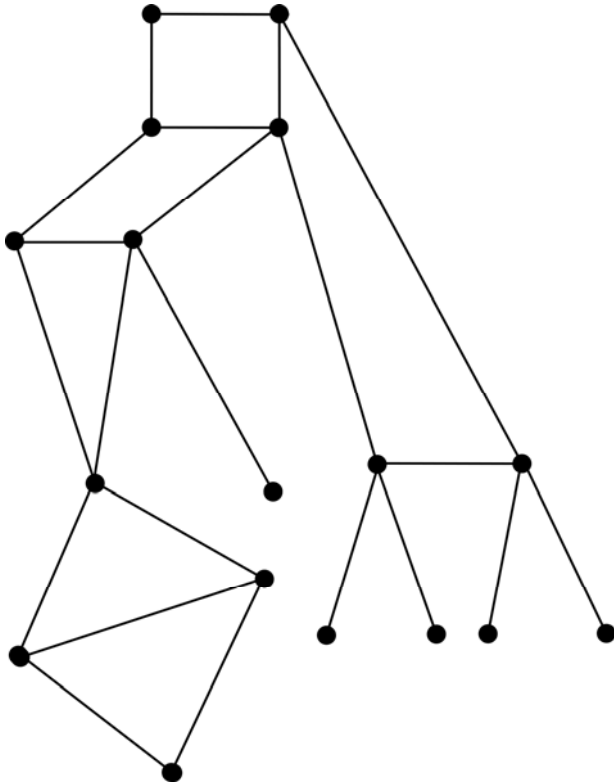
*Source renaming* :

$Rena_{a \leftrightarrow b}(G)$  exchanges source labels  $a$  and  $b$   
(replaces  $a$  by  $b$  if  $b$  is not the label of a source)

*Nullary operations* denote *basic graphs* : the connected graphs with at most one edge.

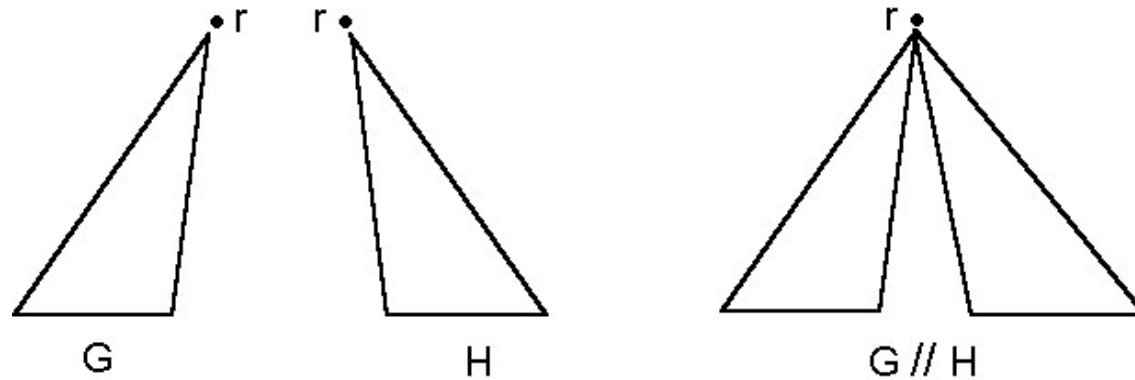


# Tree-decompositions



*Proposition:* A graph has **tree-width**  $\leq k$  if and only if it can be constructed from basic graphs with  $\leq k+1$  labels by using the operations  $//$ ,  $Ren_{a \leftrightarrow b}$  and  $Forget_a$ .

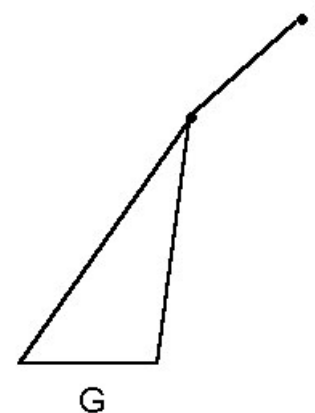
*Example :* Trees are of tree-width 1, constructed with two source labels,  $r$  (root) and  $n$  (new root): Fusion of two trees at their roots :



Extension of a tree by parallel composition with a new edge, forgetting the old root, making the "new root" as current root :

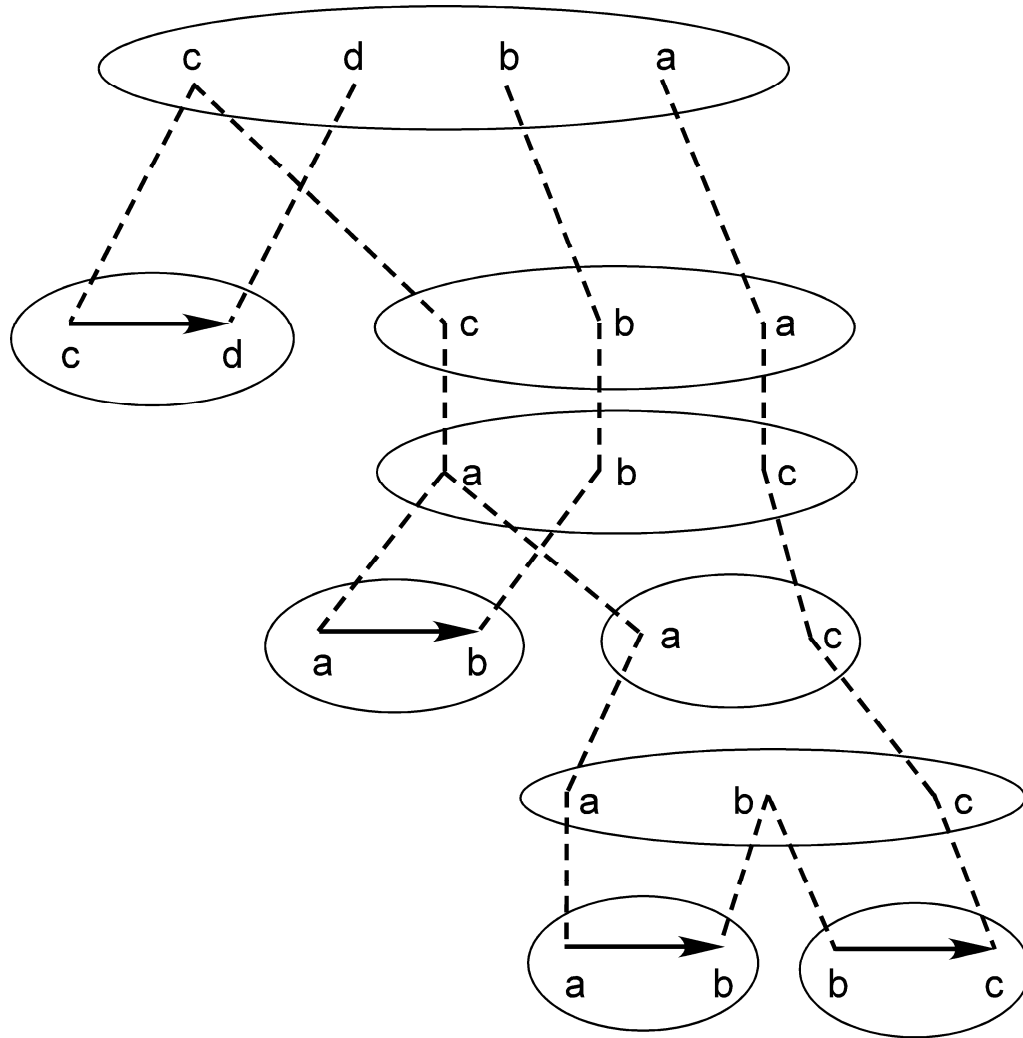
$$e = r \bullet \text{---} \bullet n$$

$$Ren_{n \leftrightarrow r} (Forget_r (G // e))$$



# From an algebraic expression to a tree-decomposition

Example :  $cd // \text{Ren}_{a \leftrightarrow c} (ab // \text{Forget}_b (ab // bc))$  (Constant  $ab$  denotes an edge from  $a$  to  $b$ )



The tree-decomposition associated with this term.