

# Graph structure and Monadic second-order logic

*Bruno Courcelle*

Université Bordeaux 1, LaBRI

## *References :*

Chapter 5 in : Handbook of graph grammars vol.1, 1997,

Book in progress,

Articles with J. Makowsky, U. Rotics, P. Weil, S. Oum, A. Blumensath

See : <http://www.labri.fr/perso/courcell/ActSci.html>

Graph structure : Embedding in a surface  
Exclusion of minor, vertex-minor, induced subgraph

### Tree-structuring

**Monadic second-order logic** : expression of properties, queries, optimization functions, number of configurations.

Mainly useful for **tree-structured graphs** (Second-order logic useless)

Two types of questions :

Checking  $G \models \varphi$  for fixed MS formula  $\varphi$ , **given G**

Deciding  $\exists G, G \in C, G \models \varphi$  for fixed **C**, **given MS formula  $\varphi$** .

Tools to be presented

**Algebraic setting** for tree-structuring of graphs

**Recognizability** = finite congruence  $\equiv$  inductive computability

$\equiv$  finite deterministic automaton on terms

**Fefermann-Vaught** : MS definability  $\Rightarrow$  recognizability.

*History* : Confluence of 4 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.

# Summary

## 1. Introduction

### *Extension of Formal Language Theory notions*

2. Recognizability, an algebraic notion.

3. Context-free sets defined by equation systems.

4. The graph algebras VR and HR.

### *Algorithmic applications :*

5. Inductive computations and recognizability; fixed-parameter tractable algorithms.

6. Monadic second-order logic defines inductive properties and functions

### *Formal language theory extended to graphs*

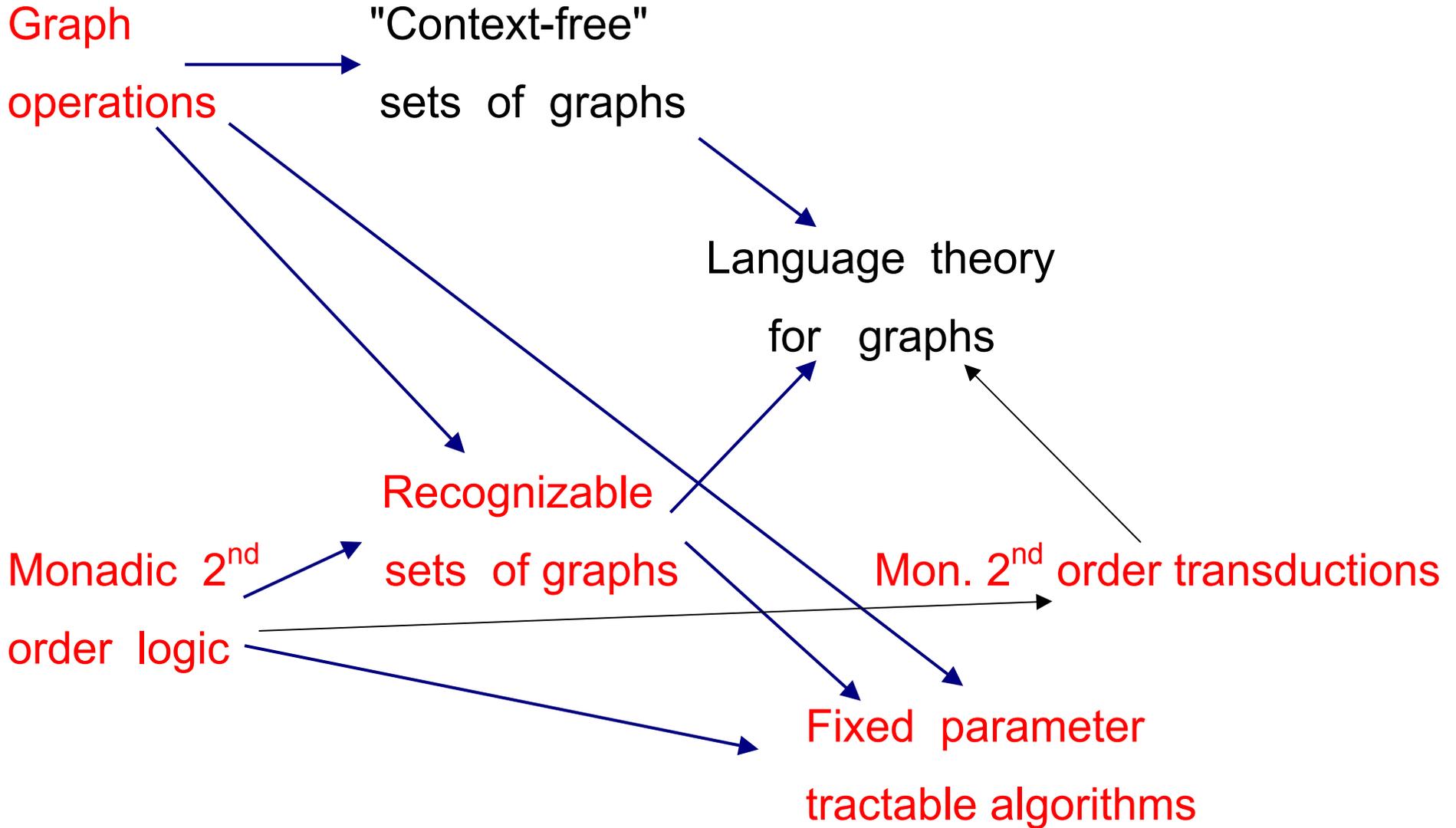
7. Closure and decidability properties ; generation of classes of graphs by monadic second-order transductions.

### *Monadic second-order logic and combinatorics*

8 Seese's conjecture

### *Open questions*

1. Introduction : An overview chart :



## Key concepts of FLT and their extensions

<i>Languages</i>	<i>Graphs</i>
Algebraic structure : monoid $(X^*, *, \varepsilon)$	Algebras based on graph operations : $\oplus, \otimes, //$ <b>quantifier-free definable operations</b> Algebras : HR, VR
Context-free languages : Equational subsets of $(X^*, *, \varepsilon)$	Equational sets of the algebras HR, VR
Regular languages : Finite automata $\equiv$ Finite congruences $\equiv$ Regular expressions $\equiv$	Recognizable sets of the algebras HR, VR defined by congruences
$\equiv$ Monadic Second-order definable sets of words or terms	$\cup$ Monadic Second-order definable sets of graphs
Rational and other types of transductions	<b>Monadic Second-order transductions</b>

## Relationships between algebraic and logical notions for sets of graphs (and hypergraphs)

Algebraic notions	Algebraic characterizations	Logical characterizations	Closure properties
EQ			union, $\cap$ Rec
	equation systems	MS-trans(Trees)	homo
	$Val(REC(Terms))$		MS-trans
REC			Boolean opns
	congruences	MS-def $\subset$ REC	homo <sup>-1</sup>
			MS-trans <sup>-1</sup>

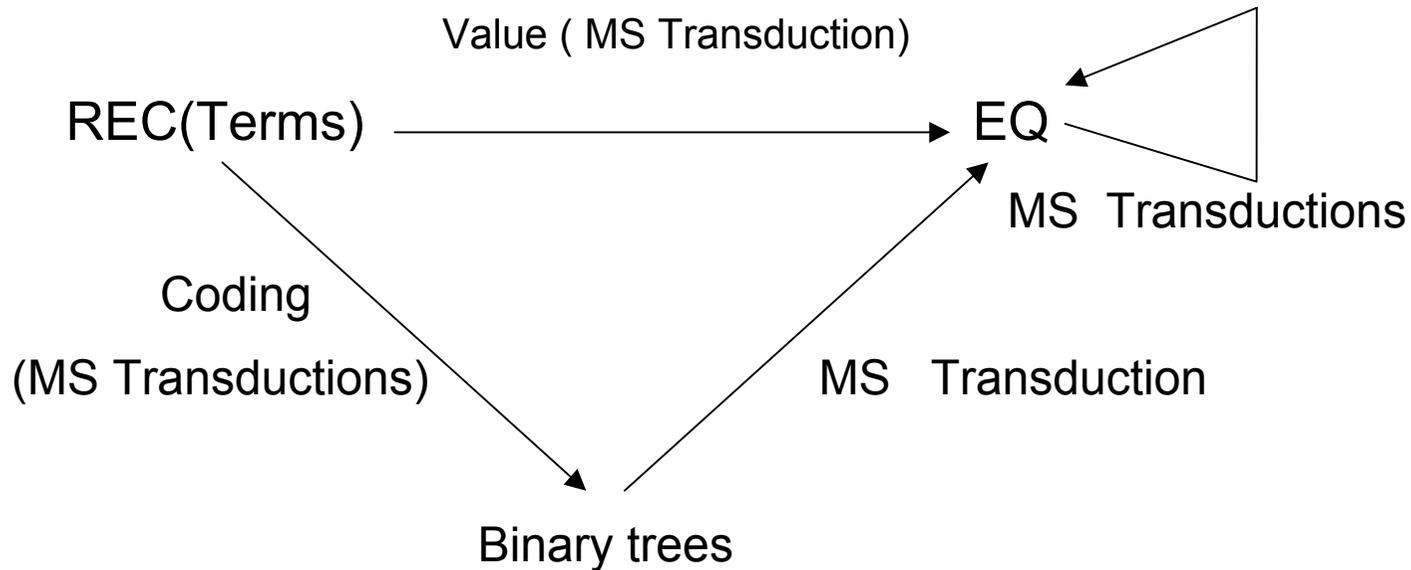
Signatures for graph algebras :

$HR$  : graphs and hypergraphs with “sources”

$VR$  : graphs with vertex labels, “ports”

$VR^+$  :  $VR$  with **quantifier-free operations** (ex. edge complement)

*Another picture :*



Equational sets = MS-Trans(Binary Trees)

Context-free languages = images of the Dyck language  
(which encodes trees) under rational transductions

Since MS transductions are closed under composition, the class of equational sets is closed under MS transductions

## 2. Recognizable sets : algebraic definition

$F$  : a finite set of operations with (fixed) arity.

$\mathbf{M} = \langle M, (f_M)_{f \in F} \rangle$  : an  $F$ -algebra.

*Definition* :  $L \subseteq M$  is *(F-)recognizable* if it is a union of equivalence classes for a finite congruence  $\approx$  on  $\mathbf{M}$  (*finite* means that  $M / \approx$  is finite).

Equivalently,  $L = h^{-1}(D)$  for a homomorphism  $h : \mathbf{M} \rightarrow \mathbf{A}$ , where  $\mathbf{A}$  is a finite  $F$ -algebra,  $D \subseteq A$ . (On terms : *Finite deterministic automata*).

$\text{REC}(\mathbf{M})$  is the set of recognizable subsets of  $M$ , with respect to the algebra  $\mathbf{M}$ .

*Closure properties* :  $\text{REC}(\mathbf{M})$  contains  $M$  and  $\emptyset$ , and is closed under union, intersection and difference.

## *The many-sorted case with infinitely many sorts*

$S$  : the countable set of sorts.

$F$  : an  $S$ -signature : each  $f$  in  $F$  has a type  $s_1 s_2 \dots s_k \rightarrow s$ , with  $s, s_i \in S$

$\mathbf{M} = \langle (M_s)_{s \in S}, (f_{\mathbf{M}})_{f \in F} \rangle$  *F-algebra*,  $M_s \cap M_t = \emptyset$ , if  $s \neq t$

where  $f_{\mathbf{M}} : M_{s_1} \times M_{s_2} \times \dots \times M_{s_k} \rightarrow M_s$

*Definition* :  $L \subseteq M_s$  is *(F-) recognizable* if it is a union of equivalence classes for a congruence  $\approx$  on  $\mathbf{M}$  such that equivalent elements are of the *same sort* and there are *finitely many classes of each sort*.

### 3. Equational (context-free) sets

Equation systems = Context-Free (Graph) Grammars  
in an algebraic setting

In the case of words, the set of context-free rules

$$S \rightarrow a S T ; \quad S \rightarrow b ; \quad T \rightarrow c T T T ; \quad T \rightarrow a$$

is equivalent to the system of two set equations:

$$S = a S T \cup \{ b \}$$

$$T = c T T T \cup \{ a \}$$

where  $S$  is the language generated by  $S$  (idem for  $T$  and  $T$ ).

For graphs (or other objects) we consider systems of equations like:

$$S = f(k(S), T) \cup \{b\}$$

$$T = f(T, f(g(T), m(T))) \cup \{a\}$$

where  $f$  is a binary operation,  $g, k, m$  are unary operations on graphs,  $a, b$  denote basic graphs (up to isomorphism).

An *equational set* is a component of the least (unique) solution of such an equation system. This is well-defined in any algebra.

# Closure properties and algebraic characterizations

## General algebraic properties

Algebraic notions	Algebraic characterizations	Closure properties
EQ	equation systems	union, $\cap$ Rec
REC	$Val(REC(Terms))$ congruences	homomorphisms Boolean operations homomorphisms <sup>-1</sup>

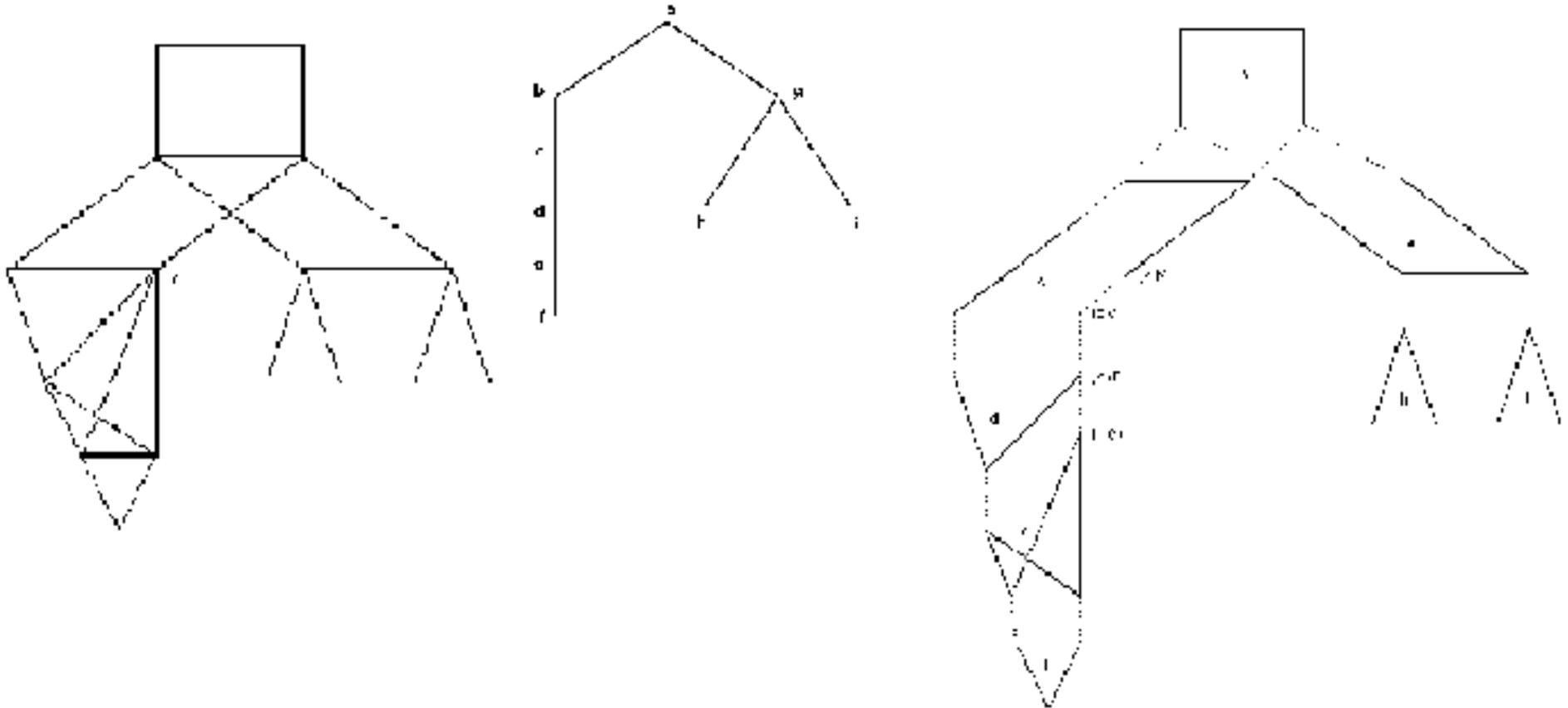
**Theorem** (Mezei and Wright) :

1) In an algebra of terms  $\mathbf{T}(F)$  :  $EQ(\mathbf{T}(F)) = REC(\mathbf{T}(F))$

2) In an F-algebra  $\mathbf{M}$  :  $EQ(\mathbf{M}) = Val_{\mathbf{M}}(REC(\mathbf{T}(F)))$

where  $Val_{\mathbf{M}} : \mathbf{T}(F) \longrightarrow \mathbf{M}$  is the **evaluation mapping**, the unique homomorphism.

## 4. The graph algebras VR and HR



*Tree-width* : Tree-decomposition of width  $k$  :  $k+1 =$  maximum size of a box

**Tree-width** :  $\text{twd}(G) =$  minimum width of a tree-decomposition

Trees have tree-width 1,  $K_n$  has tree-width  $n-1$ , the  $n \times n$  grid has tree-width  $n$

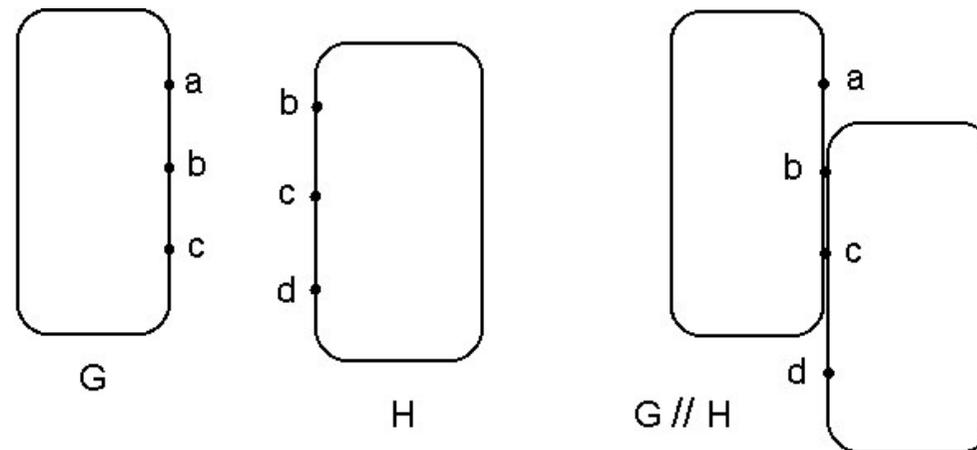
Outerplanar graphs have tree-width at most 2.

*HR operations* : Origin : **H**yperedge **R**eplacement hypergraph grammars ; associated complexity measure : **tree-width**

Graphs have distinguished vertices called **sources**, pointed to by labels from a set of size  $k$  :  $\{a, b, c, \dots, h\}$ .

*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*

$\text{Forget}_a(G)$  is  $G$  without  $a$ -source : the source is no longer distinguished ; it is made "internal".

*Source renaming* :

$\text{Ren}_{a,b}(G)$  exchanges source names  $a$  and  $b$

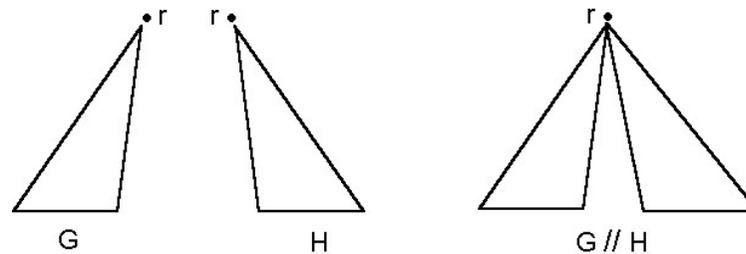
(replaces  $a$  by  $b$  if  $b$  is not the name of a source)

*Nullary operations* denote *basic graphs* : the connected graphs with at most one edge. *For dealing with hypergraphs one takes more nullary symbols for denoting hyperedges.*

More precise algebraic framework : *a many sorted algebra where each finite set of source labels is a sort.* The above operations are overloaded.

*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  $//$ ,  $Ren_{a,b}$  and  $Forget_a$ .

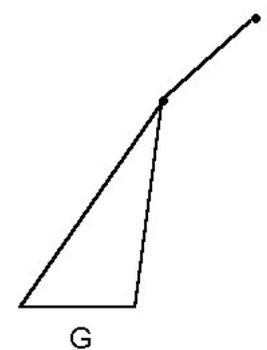
*Example :* Trees are of tree-width 1, constructed with two source labels,  $r$  (root) and  $n$  (new root):  
Fusion of two trees at their roots :



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

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

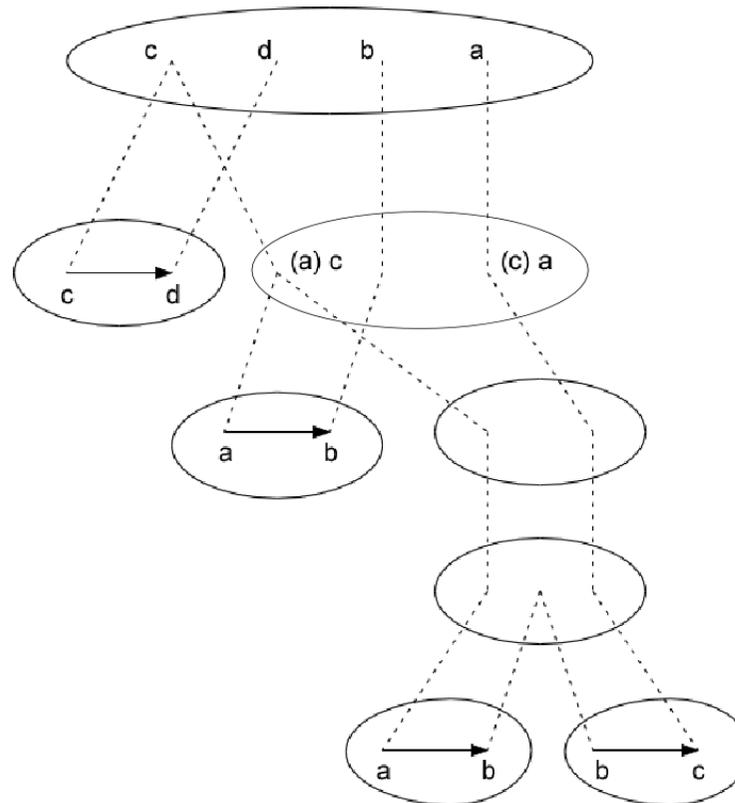
$$Ren_{n,r}(Forget_r(G // E))$$



# From an algebraic expression to a tree-decomposition

Example :  $cd // Ren_{a,c} (ab // Forget_b(ab // bc))$

Constant  $ab$  denotes a directed edge from  $a$  to  $b$ .



The tree-decomposition associated with this term.

# *VR operations*

Origin : **V**ertex **R**eplacement graph grammars

Associated complexity measure : **clique-width**, has no combinatorial characterization but is defined in terms of **few very simple graph operations** (whence easy inductive proofs).

Equivalent notion : **rank-width** (Oum and Seymour) with better structural and algorithmic properties.

Graphs are simple, directed or not.

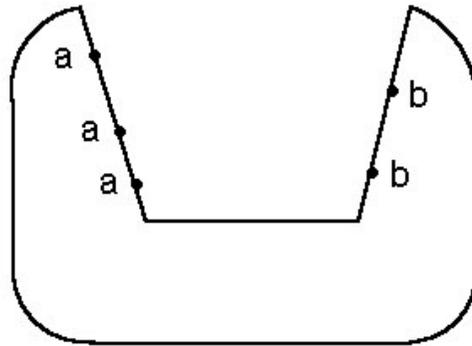
k labels : *a* , *b* , *c* , ..., *h*. Each vertex has one and only one label ;

a label *p* may label several vertices, called the ***p-ports***.

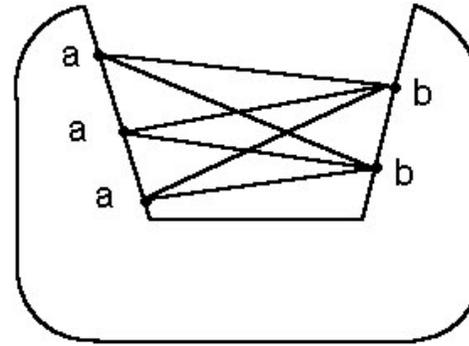
*One binary operation: disjoint union :  $\oplus$*

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

$Add-edg_{a,b}(G)$  is  $G$  augmented with (un)directed edges from every  $a$ -port to every  $b$ -port.



$G$



$Add-edg_{a,b}(G)$

Vertex relabellings :

$Relab_{a,b}(G)$  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 by means of  $k$  labels and the operations  $\oplus$ ,  $Add-edg_{a,b}$  and  $Relab_{a,b}$

Its (exact) clique-width,  $cwd(G)$ , is the smallest such  $k$ .

*Proposition* : (1) If a set of simple graphs has bounded tree-width, it has bounded clique-width, but **not vice-versa**.

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

(3) Deciding “Clique-width  $\leq 3$ ” is a polynomial problem. (Habib *et al.*)

The complexity (polynomial or NP-complete) of “Clique-width = 4” is unknown.

It is NP-complete to decide for given  $k$  and  $G$  if  $\text{cwd}(G) \leq k$ . (Fellows *et al.*)

There exists a cubic approximation algorithm that for given  $k$  and  $G$

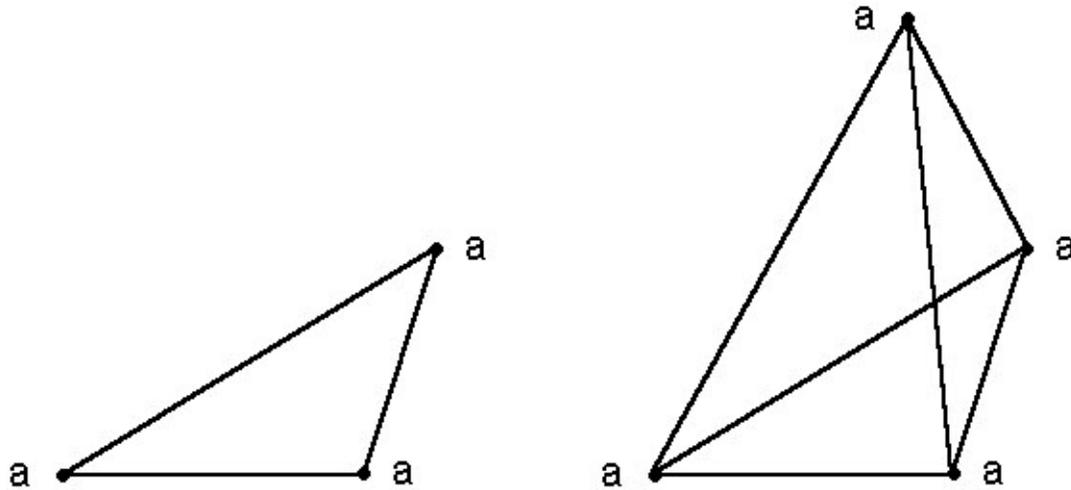
answers (correctly) :

either that  $\text{cwd}(G) > k$ ,

or produces a clique-width algebraic term using  $2^{24k}$  labels. (Oum)

This yields **Fixed Parameter Tractable** algorithms for many hard problems.

*Example* : Cliques have clique-width 2.



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

*Another example* : *Cographs*

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

$$\begin{aligned}
 G \otimes H &= \text{Relab}_{b,a}(\text{Add-edg}_{a,b}(G \oplus \text{Relab}_{a,b}(H))) \\
 &= G \oplus H \text{ with "all edges" between } G \text{ and } H.
 \end{aligned}$$

## 5. Algorithmic applications

### *Fixed parameter tractability results*

Theorem (B.C.): A) For graphs of **clique-width**  $\leq k$ ,

each **monadic second-order** property, (ex. 3-colorability),

each **monadic second-order** optimization function, (ex. distance),

each **monadic second-order** counting function, (ex. # of paths)

is evaluable :

in linear time on graphs given by a term over VR-operations,

in time  $O(n^3)$  otherwise (by S. Oum, 2005).

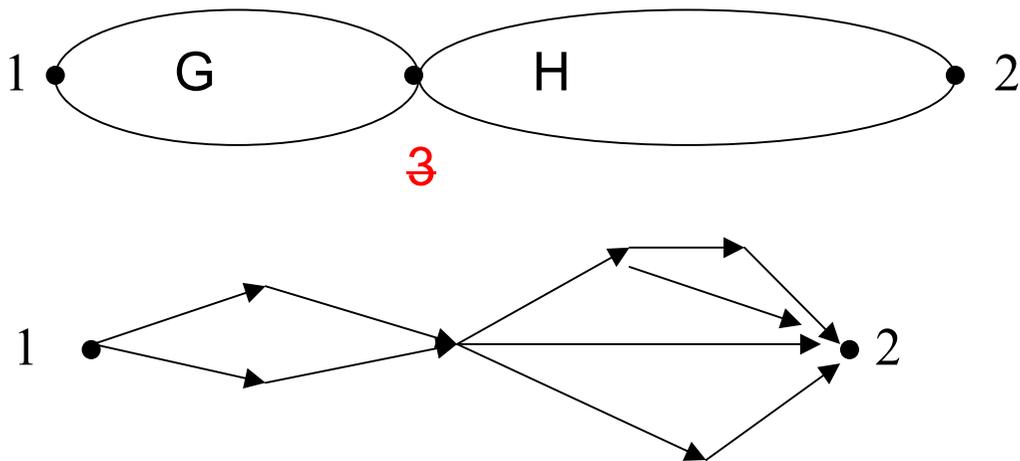
B) All this is possible in linear time on graphs of **tree-width**  $\leq k$ , for each fixed  $k$  (by Bodlaender, 1996).

# Inductive computations

*Example : Series-parallel graphs*, defined as graphs with sources 1 and 2, generated from  $e = 1 \longrightarrow 2$  and the operations  $//$  (**parallel-composition**) and **series-composition** defined from other operations by :

$$G \bullet H = \text{Forget}_3(\text{Ren}_{2,3}(G) // \text{Ren}_{1,3}(H))$$

*Example :*



## *Inductive proofs :*

1)  $G, H$  connected implies :  $G//H$  and  $G \bullet H$  are connected, (**induction**)

$e$  is connected (**basis**) :

$\Rightarrow$  All series-parallel graphs are connected.

3) It is not true that :

$G$  and  $H$  planar implies :  $G//H$  is planar ( $K_5 = H//e$ ).

A stronger property for induction :

$G$  has a planar embedding with the sources in the same “face”

$\Rightarrow$  All series-parallel graphs are planar.

## Inductive computation : Test for 2-colorability

- 1) Not all series-parallel graphs are 2-colorable (see  $K_3$ )
- 2)  $G, H$  2-colorable does not imply that  $G//H$  is 2-colorable (because  $K_3 = P_3//e$ ).

3) One can check 2-colorability with 2 auxiliary properties :

**Same**(G) = G is 2-colorable with sources of the **same color**,

**Diff**(G) = G is 2-colorable with sources of **different colors**

by using rules : **Diff**(e) = True ; **Same**(e) = False

**Same**(G//H)  $\Leftrightarrow$  **Same**(G)  $\wedge$  **Same**(H)

**Diff**(G//H)  $\Leftrightarrow$  **Diff**(G)  $\wedge$  **Diff**(H)

**Same**(G•H)  $\Leftrightarrow$  (**Same**(G)  $\wedge$  **Same**(H))  $\vee$  (**Diff**(G)  $\wedge$  **Diff**(H))

**Diff**(G•H)  $\Leftrightarrow$  (**Same**(G)  $\wedge$  **Diff**(H))  $\vee$  (**Diff**(G)  $\wedge$  **Same**(H))

We can compute for every SP-term **t**, by induction on the structure of **t** the pair of Boolean values (**Same**(Val(**t**)), **Diff**(Val(**t**))).

We get the answer for  $G = \text{Val}(\mathbf{t})$  (the graph that is the *value* of **t**) regarding 2-colorability.

## Important facts :

1) The existence of properties forming an inductive set (w.r.t. operations of  $F$ ) is equivalent to recognizability in the considered  $F$ -algebra.

2) The simultaneous computation of  $m$  inductive properties can be implemented by a "tree" automaton with  $2^m$  states working on terms  $t$ . This computation takes time  $O(|t|)$ .

3) An inductive set of properties can be constructed (at least **theoretically**) from every **monadic-second order formula**.

4) This result extends to the computation of values (integers) defined by **monadic-second order formulas**.

# Recognizability and inductive properties

*Definition* : A finite set  $P$  of properties on an  $F$ -algebra  $\mathbf{M}$  is **F-inductive** if for every  $p \in P$  and  $f \in F$ , there exists a (**known**) Boolean formula  $B$  such that :

$$p(f_{\mathbf{M}}(a,b)) = B[\dots, q(a), \dots, q(b), \dots, q \in P]$$

for all  $a$  and  $b$  in  $\mathbf{M}$ . (where  $q(a), \dots, q(b) \in \{\text{True}, \text{False}\}$ ).

*Proposition* : A subset  $L$  of  $\mathbf{M}$  is **recognizable** iff it is the set of elements that satisfy a property belonging to a finite **inductive set** of properties  $P$ .

*Proof* : Let  $L = h^{-1}(C)$  for a homomorphism  $h : \mathbf{M} \rightarrow \mathbf{A}$ ,  $\mathbf{A}$  a finite  $F$ -algebra and  $C$  a subset of  $A$  (domain of  $\mathbf{A}$ ).

For each  $a$  in  $A$ , let  $\hat{a}$  be the property :  $\hat{a}(m) = \text{True} \Leftrightarrow h(m) = a$ . Let  $p$  be such that  $p(m) = \text{True} \Leftrightarrow h(m) \in C \Leftrightarrow m \in L$ .

Properties  $\{p, \hat{a} / a \in A\}$  form an **F-inductive set**.

If  $P$  is an inductive set of  $k$  properties, one can define an  $F$ -algebra structure on the set  $\mathbf{B}^k$  of  $k$ -tuples of Booleans, such that the mapping

$h : m \mapsto \text{the } k\text{-tuple of Booleans}$  is a homomorphism.

## Inductive properties and automata on terms

If  $P$  is an inductive set of  $k$  properties, one can define a deterministic **automaton** on terms of  $T(F)$  with set of states the  $k$ -tuples of Booleans, that computes in a bottom-up way, for each term  $t$ , the truth values :

$p(\text{Val}(s))$  for all  $p \in P$  and all subterms  $s$  of  $t$ .

Membership of an element  $m$  of  $M$  in a *recognizable set*  $L$  can be tested by such an automaton on *any* term  $t$  in  $T(F)$  defining  $m$ .

## Application to graphs

Immediate but depends on two things :

*Parsing algorithms* building terms from the given graphs :

1) results by Bodlaender for constructing tree-decompositions (in linear time), whence terms representing them,

2) results by Oum for constructing (non-optimal) terms for graphs of clique-width at most  $k$ . (Cubic time algorithm).

*Language* for expressing inductive properties : **Monadic second-order logic**.

## 6. 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 properties** : transitive closure, properties of paths, connectivity,  
planarity (via Kuratowski, uses connectivity), k-colorability.

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

*Non connectivity :*

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

*2-colorability (i.e. G is bipartite) :*

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

## Edge set quantifications

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

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

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

Existence of Hamiltonian circuit is expressible by an  $\text{MS}_2$  formula, but **not by an MS** formula.

*Theorem* :  $\text{MS}_2$  formulas are no more powerful than MS formulas :

for graphs of degree  $< d$ , or of tree-width  $< k$ ,  
or for planar graphs, or graphs without some fixed  $H$  as a minor,  
or graphs of average degree  $< k$  (uniformly  $k$ -sparse).

*Definition* : A set  $L$  of words, of trees, of graphs or relational structures is **Monadic Second-Order (MS) definable** iff

$$L = \{ S \mid S \models \varphi \} \text{ for an MS formula } \varphi$$

**Theorem** : (1) A language (set of words or *finite terms*) is recognizable  $\Leftrightarrow$  it is MS definable

(2) A set of finite graphs is VR- or VR<sup>+</sup>-recognizable  
 $\Leftarrow$  it is MS definable

(3) A set of finite graphs is HR-recognizable  
 $\Leftarrow$  it is MS<sub>2</sub> definable

*Proofs:*

- (1) Doner, Thatcher, Wright, see W. Thomas, Handbook formal languages, vol.3.
- (2) (3) There are two possible proofs .

*Basic facts for (2) :*

Let  $F$  consist of  $\oplus$  and unary **quantifier-free definable operations**  $f$ .

For every MS formula  $\varphi$  of quantifier-height  $k$ , we have

(a) for every  $f$ , one can construct a formula  $f^\#(\varphi)$  such that :

$$f(S) \models \varphi \Leftrightarrow S \models f^\#(\varphi)$$

(b) (Hanf, Fefermann and Vaught, Shelah) one can construct formulas  $\psi_1, \dots, \psi_n, \theta_1, \dots, \theta_n$  such that :

$$S \oplus T \models \varphi \Leftrightarrow \text{for some } i, S \models \psi_i \wedge T \models \theta_i$$

where  $f^\#(\varphi), \psi_1, \dots, \psi_n, \theta_1, \dots, \theta_n$  have **quantifier-height  $\leq k$** .

(c) Up to equivalence, there are finitely many formulas of quantifier-height  $\leq k$  forming a set  $\Phi_k$ .

One builds an automaton with states the **subsets of**  $\Phi_k$  : the MS-theories of quantifier-height  $\leq k$  of the graphs defined by the subterms of the term to be processed.

Fefermann-Vaught for  $\oplus$ .

$$S \oplus T \models \varphi \iff \bigvee_i S \models \psi_i \wedge T \models \theta_i$$

with  $qh(\psi_i), qh(\theta_i) \leq qh(\varphi)$

$$Th_h(S \oplus T) = \Phi_{\oplus, h}(Th_h(S), Th_h(T))$$

We need to handle formulas with **free** variables

$$(S \oplus T, \bar{A}) \models \varphi \iff \bigvee_i (S, \bar{A} \upharpoonright_S) \models \psi_i \wedge (T, \bar{A} \upharpoonright_T) \models \theta_i$$

$$\text{sat}(S \oplus T, \varphi, \bar{x}) = \bigoplus_i \text{sat}(S, \psi_i, \bar{x}) \boxtimes \text{sat}(T, \theta_i, \bar{x})$$

$$\text{where } \mathcal{B} \boxtimes \mathcal{C} = \{ (B_1 \cup C_1, \dots, B_n \cup C_n) \mid \bar{B} \in \mathcal{B}, \bar{C} \in \mathcal{C} \}$$

MS logic with only set variables:

$$X \subseteq Y, X = \emptyset, \text{Sgl}(x), \text{Card}_{p,q}(x)$$

$$R(X, Y, Z) \iff \exists x \in X, y \in Y, z \in Z. R(x, y, z)$$

$$\exists x. \varphi(x, \dots) \iff \exists X. \text{Sgl}(x) \wedge \tilde{\varphi}(X, \dots)$$

Proof by induction on  $\varphi$  :

There exists a **decomposition**

$$[(\psi_1, \theta_1), \dots, (\psi_m, \theta_m)]_{\varphi} \quad \text{gh}(\psi_i), \text{gh}(\theta_i) \leq \text{gh}(\varphi)$$

s.t.

$$\text{sat}(S \oplus T, \varphi) = \bigcup_i \text{sat}(S, \psi_i) \boxtimes \text{sat}(T, \theta_i)$$

Cases :

- $X \subseteq Y$  :  $[(X \subseteq Y, X \subseteq Y)]$
- $\text{Sgl}(X)$  :  $[(\text{Sgl}(X), X = \phi), (X = \phi, \text{Sgl}(X))]$
- $R(X, Y)$  :  $[(R(X, Y), \text{True}), (\text{True}, R(X, Y))]$
- $\varphi \vee \varphi'$  :  $[\dots]_{\varphi} \cdot [\dots]_{\varphi'}$
- $\neg \varphi$  : We use a lemma:

$$\bigwedge_i \neg \alpha_i \boxtimes \neg \beta_i \equiv \bigvee_I \left( \bigwedge_{i \in I} \neg \alpha_i \wedge \bigwedge_{i \notin I} \neg \beta_i \right)$$

gives  $\wedge$

$$(S \oplus T, \bar{A}) \models \exists X. \varphi(X)$$

$$\Leftrightarrow (S \oplus T, \bar{A}B) \models \varphi \quad \text{for some } B$$

$$\Leftrightarrow \forall V_i (S, \bar{A}_{\uparrow S} B \cap D_S) \models \psi_i \wedge (T, \bar{A}_{\uparrow T} B \cap D_T) \models \theta_i$$

$$\Leftrightarrow \forall V_i (S, \bar{A}_{\uparrow S}) \models \exists X. \psi_i(X) \wedge (T, \bar{A}_{\uparrow T}) \models \exists X. \theta_i(X)$$

Hence we get  $[\dots]_{\exists X. \varphi(X)}$  from  $[\dots]_{\varphi}$   
by adding  $\exists X.$  to formulas.

The condition on quantifier-height is satisfied

For  $\forall X. \varphi(X)$  we use  $\neg \exists X. \neg \varphi(X)$ .

Exercises : •  $\text{Card}_{p,q}(X)$

• Replace  $U_i$  by  $(+)_i$

## 7. Monadic second-order transductions

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

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

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

where  $T$  is :

a) defined by MS formulas

b) inside the structure:  $S \oplus S \oplus \dots \oplus S$

(fixed number of disjoint "marked" copies of  $S$ )

c) in terms of "parameters", subsets  $X_1, \dots, X_p$  of the domain of  $S$ .

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

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

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

$(G, X, Y) \mapsto$  the minor of  $G$  resulting from contraction of edges in  $X$  and deletion of edges and vertices in  $Y$ .

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

For  $u = aac$ , we have  $S$

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

$S \oplus S$

$$\begin{array}{ccc} \cdot \rightarrow \cdot \rightarrow \cdot & & \cdot \rightarrow \cdot \rightarrow \cdot \\ a \quad a \quad c & & a \quad a \quad c \\ p_1 \quad p_1 \quad p_1 & & p_2 \quad p_2 \quad p_2 \end{array}$$

$\delta(S)$

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

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

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

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

The fundamental property of MS transductions :

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

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

Every MS formula  $\psi$  has an effectively computable *backwards translation*  $\tau^\#(\psi)$ , an MS formula, such that :

$$S \models \tau^\#(\psi) \quad \text{iff} \quad \tau(S) \models \psi$$

The verification of  $\psi$  in the **object structure**  $\tau(S)$  **reduces** to the **verification** of  $\tau^\#(\psi)$  in the given structure  $S$ .

*Intuition* :  $S$  contain all 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 satisfiability problem**, so has its image under an MS transduction.

## Another look at $MS_2$ versus MS formulas

*Theorem* : The mapping  $G \rightarrow \text{Inc}(G)$  is an MS-transduction on each of the following classes of simple graphs :

- degree  $< d$ ,
- tree-width  $< k$ ,
- planar graphs,
- graphs without some fixed  $H$  as a minor,
- graphs of average degree  $< k$  (uniformly  $k$ -sparse) .

## Other results

1) A set of graphs is VR-equational iff it is the image of **(all)** binary trees under an MS transduction. VR-equational sets are stable under MS-transductions.

A set of graphs has bounded **clique-width** iff it is the image of **a** set of binary trees under an **MS** transduction.

2) A set of graphs is HR-equational iff it is the image of **(all)** binary trees under an **MS<sub>2</sub> transduction**.

HR-equational sets are stable under **MS<sub>2</sub>**-transductions.

A set of graphs has bounded **tree-width** iff it is the image of **a** set of binary trees under an **MS<sub>2</sub> transduction**.

3) A set of hypergraphs is QF-equational iff it is the image of **(all)** binary trees under an MS-transduction.

QF-equational sets are stable under MS-transductions.

4) (A.Blumensath, B.C., 2004) : QF-recognizable sets are preserved under **inverse MS transductions**.

## Relationships between algebraic and logical notions

Algebraic notions	Algebraic characterizations	Logical characterizations	Closure properties
EQ			union, $\cap$ Rec
	equation systems	MS-trans(Trees)	homo
	$Val(REC(Terms))$		MS-trans
REC			Boolean opns
	congruences	MS-def $\subset$ REC	homo <sup>-1</sup>
			MS-trans <sup>-1</sup>

### Signatures for graphs and hypergraphs :

*HR* : graphs and hypergraphs with “sources”

*VR* : graphs with vertex labels (“ports”)

*VR*<sup>+</sup> : *VR* with quantifier-free operations (ex. edge complement)

*QF* : hypergraphs, i.e., relational structures (disjoint union  $\oplus$  and quantifier-free definable unary operations)

## 8. Links between MS logic and combinatorics: Seese's Theorem and Conjecture

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

*Conjecture* (Seese 1991): If a set of graphs has a decidable  $MS$  satisfiability problem, it is the image of a set of trees under an  $MS$  transduction, equivalently, has bounded **clique-width**.

*Theorem* (B.C., S. Oum 2004): If a set of graphs has a decidable  $C_2MS$  satisfiability problem, it has bounded clique-width.

$MS$  = (Basic)  $MS$  logic **without** edge quantifications,  $MS_2$  =  $MS$  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 for every formula in  $L$  whether it 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  $MS_2$  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<sub>2</sub>MS satisfiability** problem, so has Vertex-Minors( $C$ ), because  $C \longrightarrow$  Vertex-Minors( $C$ ) is a C<sub>2</sub>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. Result with 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

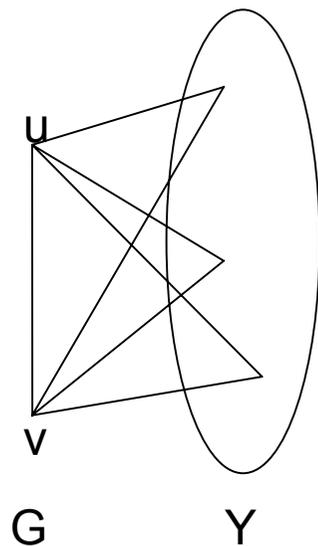
(at all vertices)

Vertex-minor relation :

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

*Proposition (Courcelle and Oum 2007)* : The mapping that associates with  $G$  its locally equivalent graphs is a **C<sub>2</sub>MS** transduction.

Why is the even cardinality set predicate necessary ?



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

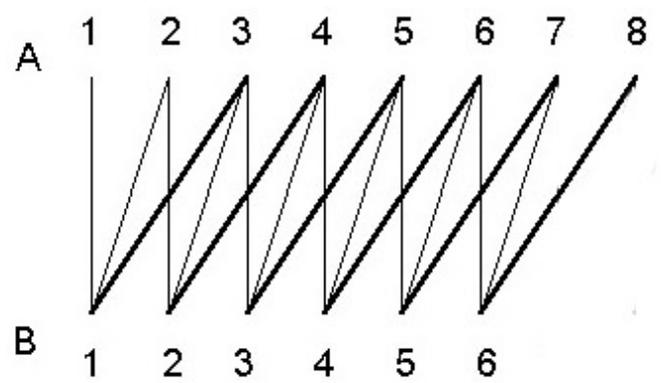
$u$  is linked to  $v$  in  $G * X$

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

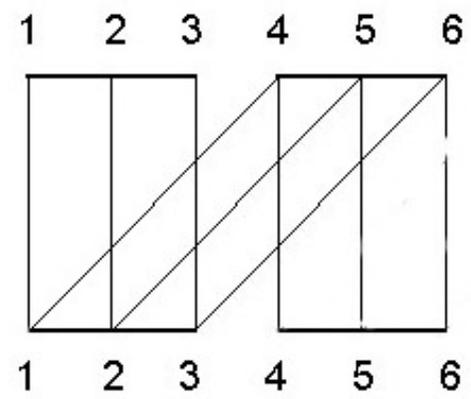
**Definition of**  $S_k$  : bipartite :  $A = \{1, \dots, (k+1)(k-1)\}$  ,  $B = \{1, \dots, k(k-1)\}$  for  $j \in A$ ,  $i \in B$  :

$$\text{edg}(i,j) \Leftrightarrow i \leq j \leq i+k-1$$

From  $S_k$  to  $\text{Grid}_{k \times k}$  by an MS transduction



$S_3$



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

1) One can define the orderings of A and B :

$$x, y \text{ are consecutive} \Leftrightarrow \text{Card}(n_G(x) \Delta n_G(y)) = 2$$

2) One can identify 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)

3) One can create 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...) and delete others (from  $4 \in B$  to  $6 \in A$  etc...), and vertices like 7,8 in A, to get a grid containing  $\text{Grid}_{k \times k}$

*Corollary* : If a set of directed acyclic graphs having Hamiltonian directed paths has a decidable **MS** satisfiability problem, then :

it has bounded clique-width,

it is the image of a set of trees under an MS transduction.

*Proof* : Since on these graphs a linear order is MS definable, **MS** and **C<sub>2</sub>MS** are equivalent.

The previously known techniques for similar results (in particular for line graphs or interval graphs, B.C. 2004) do not work in this case.

## 9. A few open questions

*Question 1* (A. Blumensath, P. Weil, B.C.): Which operations, quantifier-free definable or not, yield extensions of VR, HR, QF that are equivalent ?

*Question 2* : Under which operations, quantifier-free definable or not, are REC(VR) and REC(HR) closed ?

The case of REC(HR) is considered in B.C.: *(HR-)Recognizable sets of graphs, equivalent definitions and closure properties*, 1994. It is not hard to see that REC(VR) is closed under  $\oplus$  (disjoint union) but not under the operations Add-edg<sub>a,b</sub>.

*Answer : unary operations with inverse defined by an MS-transduction.*

*Question 3* : Is it true that the decidability of the MS (and not of the C<sub>2</sub>MS) satisfiability problem for a set of graphs implies bounded clique-width, as conjectured by D. Seese?

What about sets of hypergraphs or relational structures?