



# Graph structure and monadic second-order logic: Language theoretical aspects

*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* : 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.

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)**

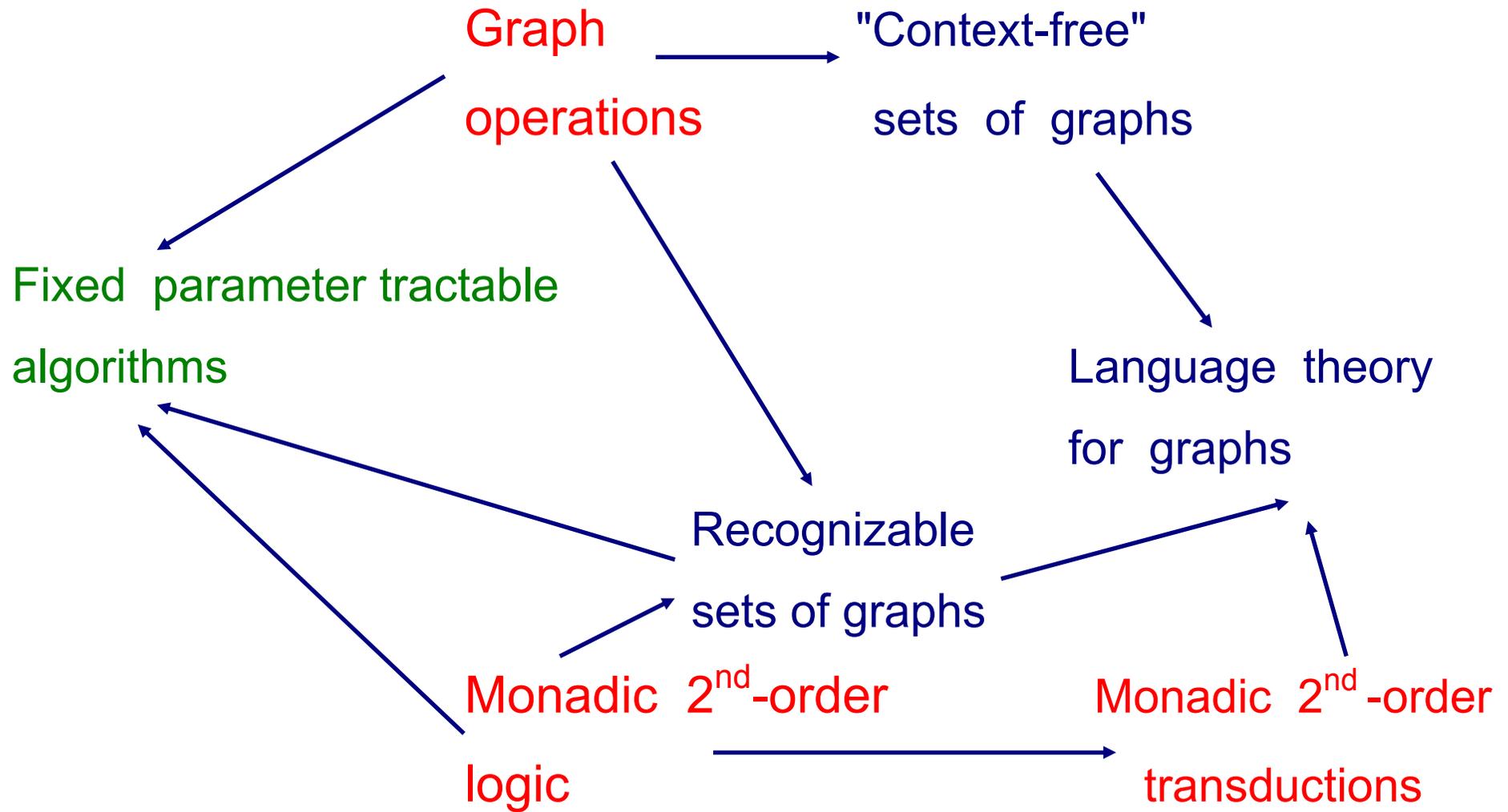
for expressing graph properties (i.e., defining graph classes)

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

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

- 1) 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  $\leq k$ , **excluded 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 transductions** that are independent of the initial combinatorial or algebraic definitions : robustness of definitions and stability under graph transformations specified in MS logic.

# An overview chart



# Key concepts of Language Theory 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 : <b>HR, VR</b>
Context-free languages : Equational subsets of $(X^*, *, \varepsilon)$	Equational sets of the algebras <b>HR, VR</b>
Regular languages : Finite automata $\equiv$ Finite congruences $\equiv$ Regular expressions $\equiv$	Recognizable sets of the algebras <b>HR, VR</b>  defined by finite 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>

# Summary

Introduction (finished)

## *Extension to graphs of Language Theoretical notions*

1. Context-free sets defined by equation systems.
2. The graph algebras **VR** and **HR**.
3. Recognizability as an algebraic notion.
4. Monadic second-order logic defines inductive properties and functions
5. Monadic second-order transductions.
6. Robustness results : preservation of classes under direct and inverse monadic second-order transductions.
7. Comparing *encoding powers* of graph classes via MS transductions

## *Links with logic and graph theory*

8. Graph classes on which MS logic is decidable

## *Open questions*

# 1. 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:

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

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

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*.

Many properties are valid at the general algebraic level.

## 2. The graph algebras HR and VR

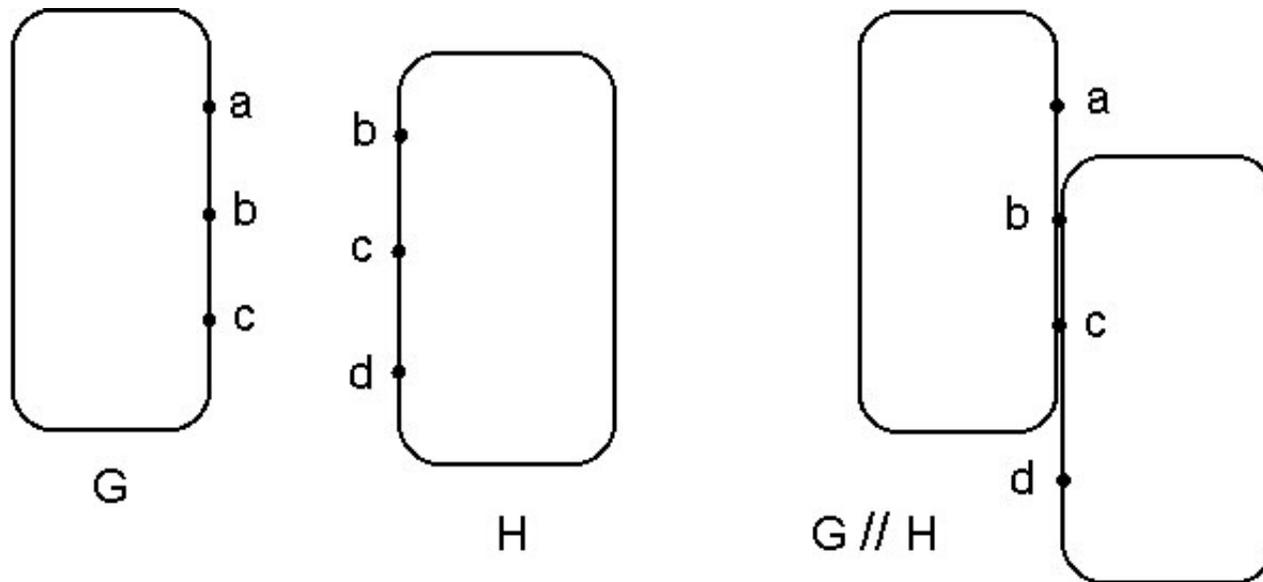
*HR operations* : Origin : Hyperedge Replacement hypergraph grammars ; associated complexity measure : **tree-width**

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.

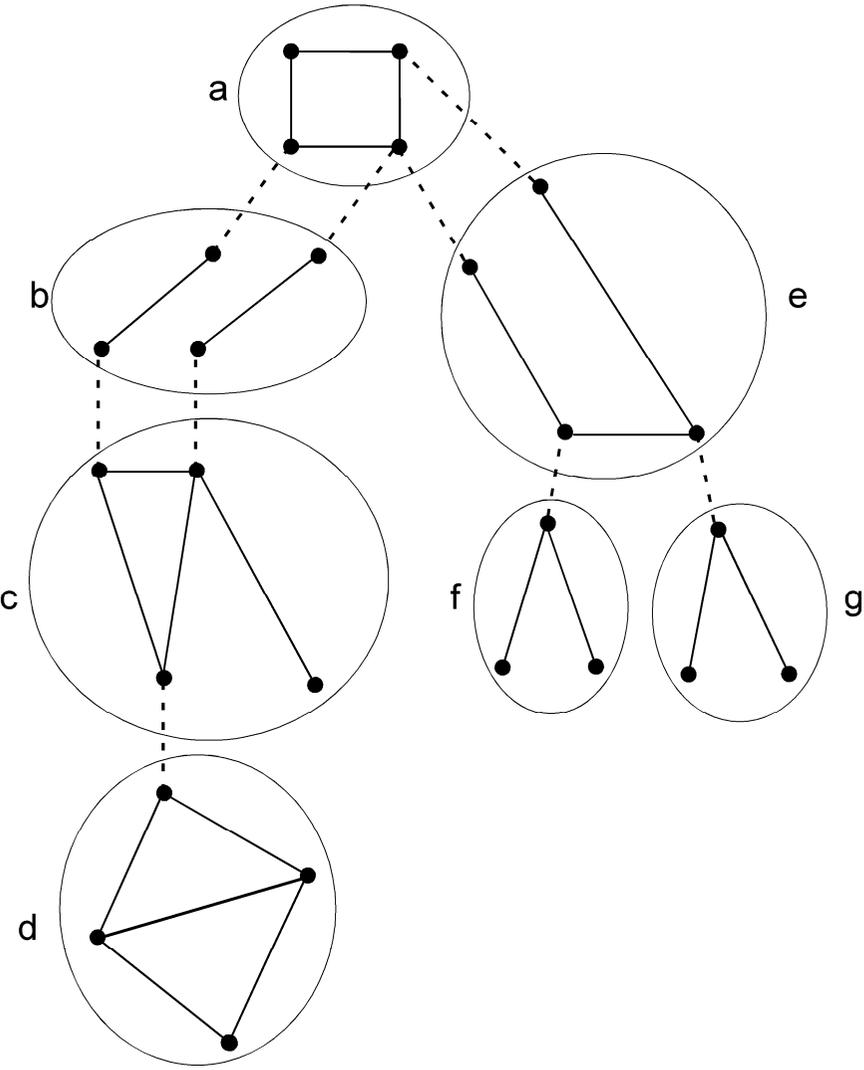
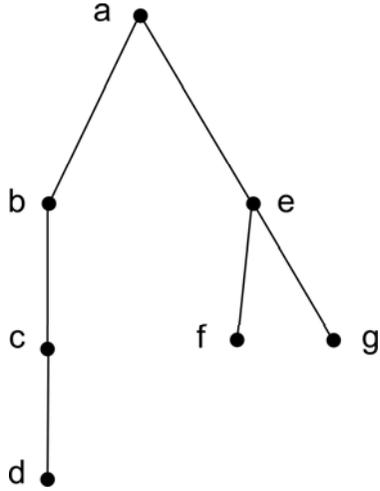
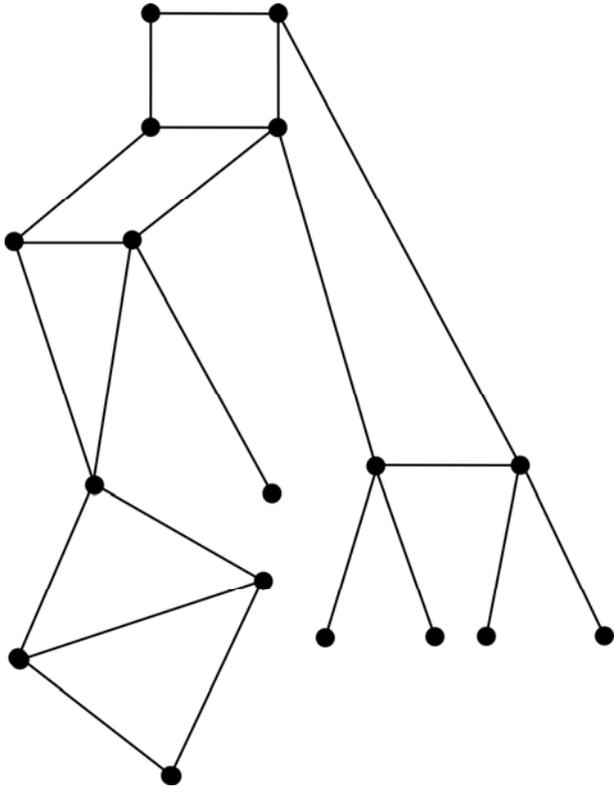
*For dealing with hypergraphs one takes more nullary symbols for denoting hyperedges.*

Each graph  $G$  has *type*  $\tau(G)$  = the set of labels of its sources.

The type function has a homomorphic behaviour :

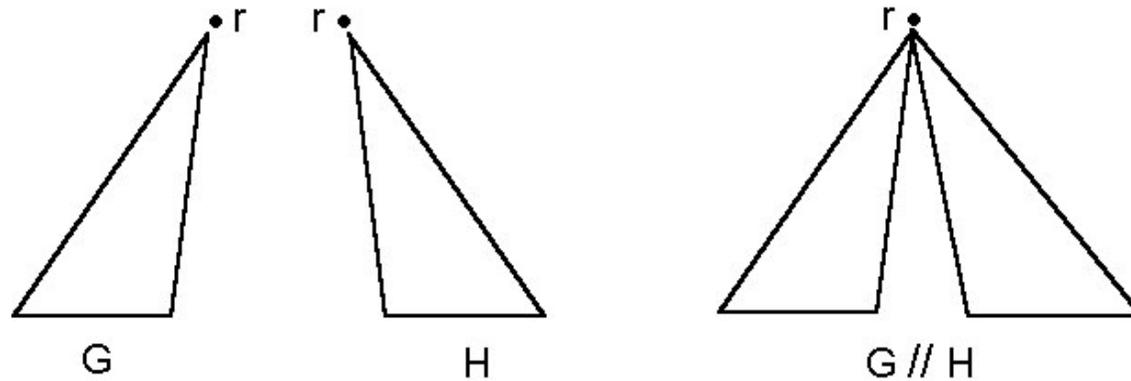
$$\tau(G//H) = \tau(G) \cup \tau(H) ; \tau(Forget_a(G)) = \tau(G) - \{a\} ; \tau(Rena_{a \leftrightarrow b}(G)) = \tau(G)[a/b, b/a].$$

# 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  $Forget_a$ .

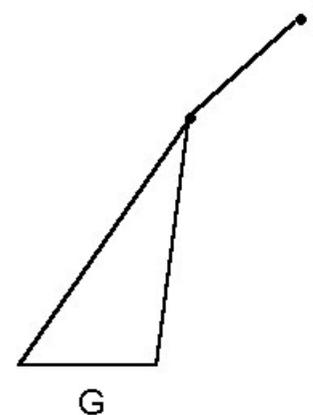
*Example :* Trees are of tree-width 1, constructed with two source labels,  $r$  (root) and  $n$  (new root): Fusion of two trees at their roots :



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

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

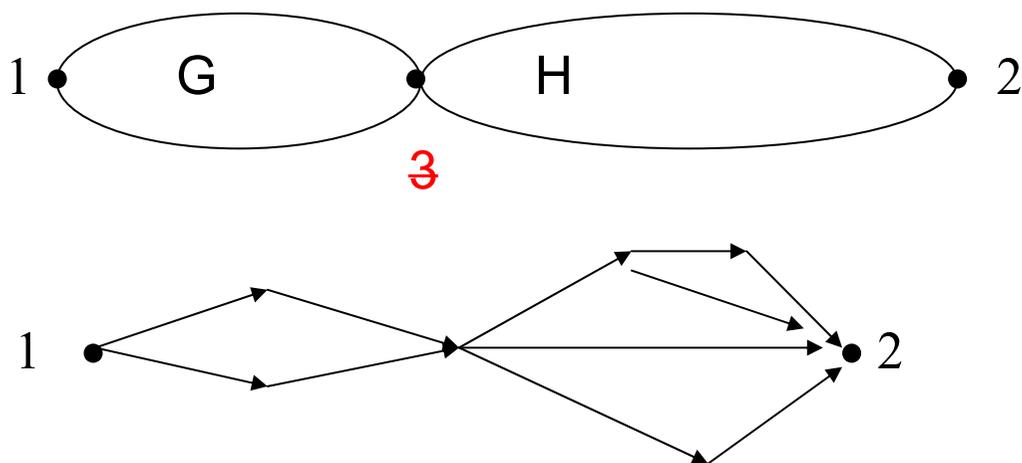
$$Ren_{n \leftrightarrow r} (Forget_r (G // e))$$



Example : *Series-parallel graphs*, defined as directed 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 :

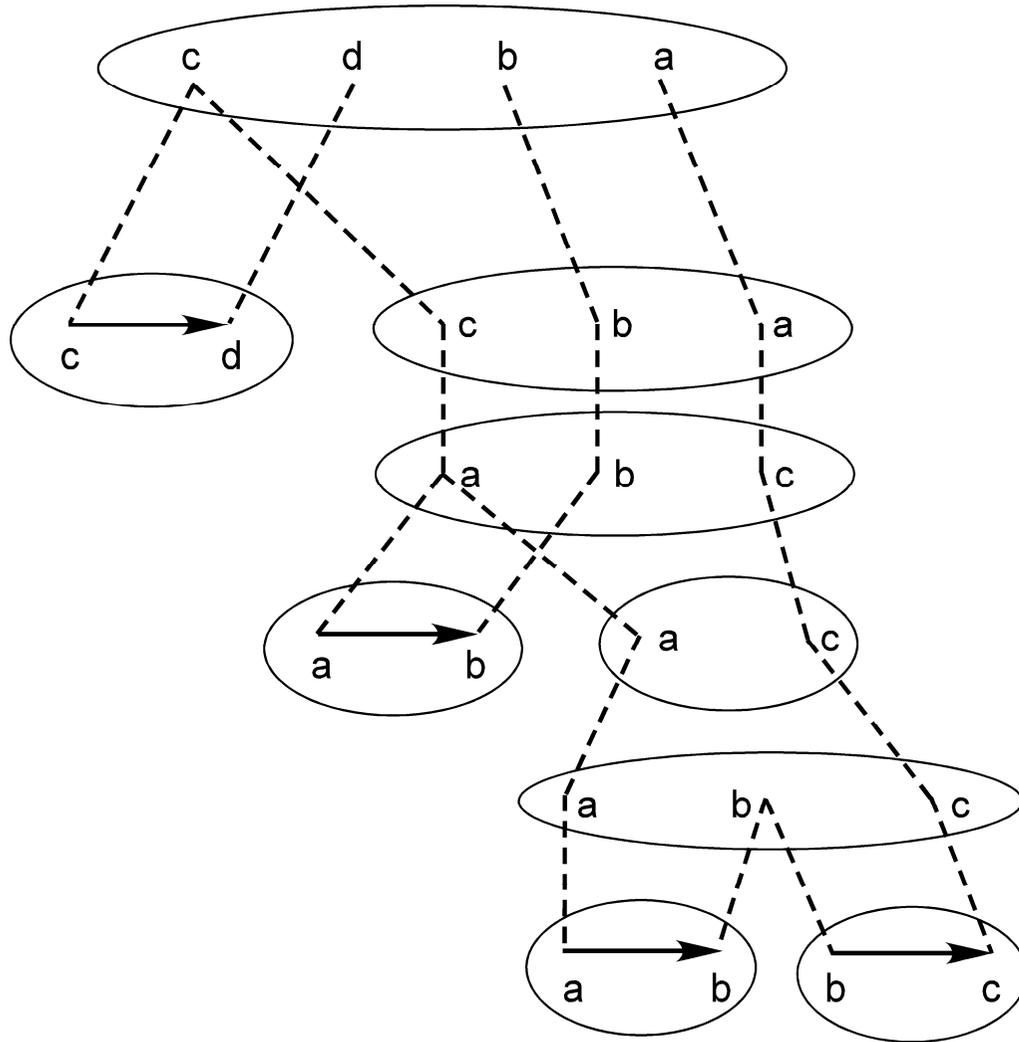
$$G \bullet H = \text{Forget}_3(\text{Ren}_{2 \leftrightarrow 3}(G) // \text{Ren}_{1 \leftrightarrow 3}(H))$$

Example :



# From an algebraic expression to a tree-decomposition

Example :  $cd // Ren_{a \leftrightarrow c} (ab // Forget_b(ab // bc))$  (Constant  $ab$  denotes an 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** (giving easy inductive proofs).

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

Graphs are simple, directed or not.

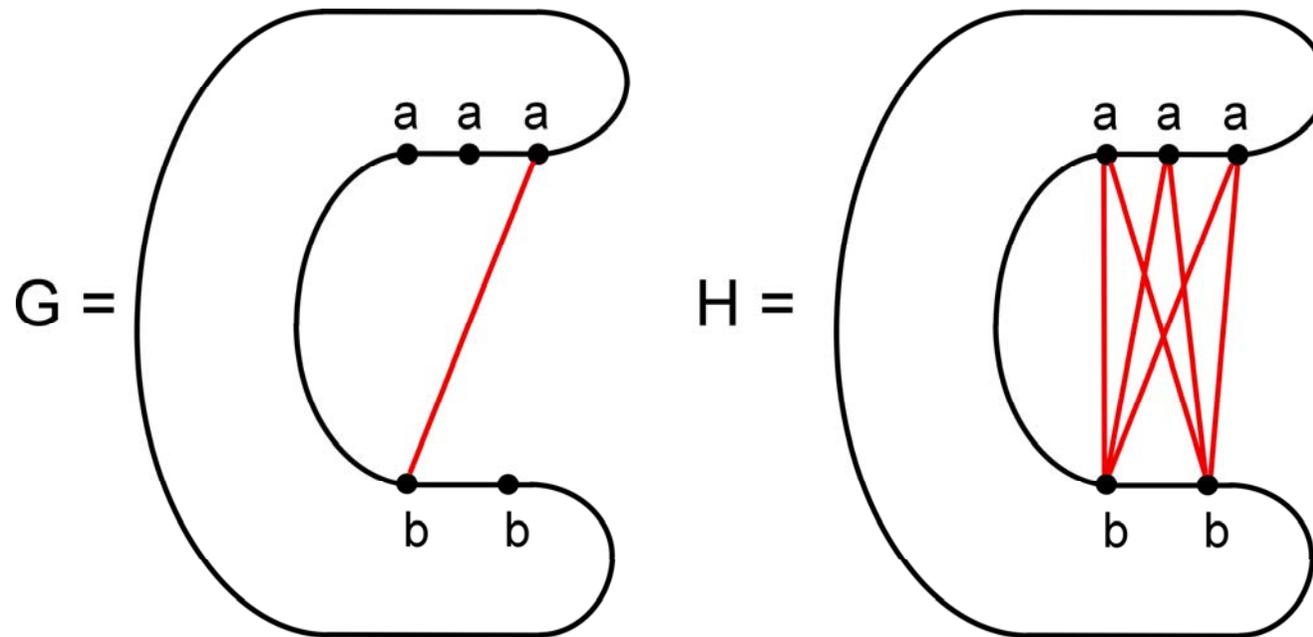
$k$  labels :  $a, b, c, \dots, h$ . Each vertex has one and only one label ;

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.



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

The number of added edges depends on the argument graph.

Vertex relabellings :

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

Basic graphs are those with a single vertex.

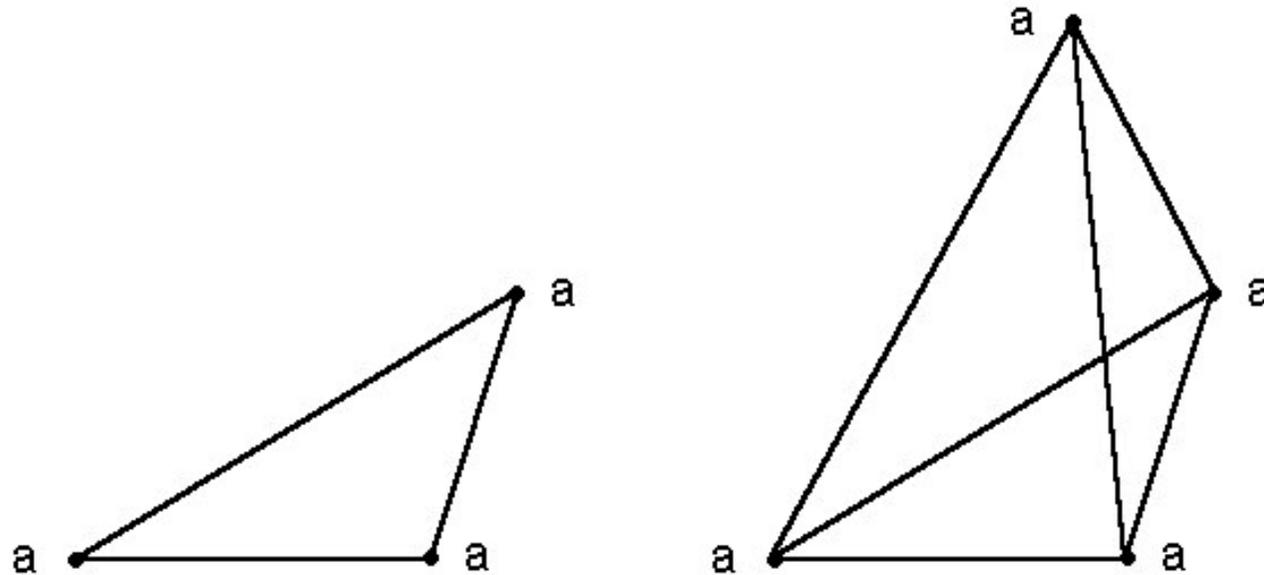
*Definition:* A graph  $G$  has **clique-width**  $\leq k \Leftrightarrow$  it can be constructed from basic graphs with the operations  $\oplus$ ,  $Add-edg_{a,b}$  and  $Relab_{a \rightarrow b}$  with  $k$  labels.

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

The **type** (for the VR algebra) of graph  $G$  is  $\tau(G)$  = the set of port labels having an occurrence. Type has a homomorphic behaviour :

$$\tau(G \oplus H) = \tau(G) \cup \tau(H) ; \tau(Add-edg_{a,b}(G)) = \tau(G) ; \tau(Relab_{a \rightarrow b}(G)) = \tau(G)[b/a].$$

*Example* : Cliques have clique-width 2.



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

*Another example* : Cographs are generated by  $\oplus$  and  $\otimes$  defined by :

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

$$= G \oplus H \text{ with "all edges" between } G \text{ and } H.$$

*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).

## Two algebras of graphs **HR** and **VR**

Two notions of context-free sets : the equational sets of algebras **HR** and **VR**, (and two notions of recognizable sets, based on congruences).

Why not using a third algebra ?

One could, but  $\text{Equat}(\mathbf{HR})$  and  $\text{Equat}(\mathbf{VR})$  have **robustness results** :

Independent logical characterizations,  
stability under certain logically defined transductions,  
generation from trees.

Which properties follow from the algebraic setting ?

**Answers** : Closure under union,  $//$  ,  $\oplus$  and the unary operations.

Emptiness and finiteness are decidable (**finite sets are computable**)

Parikh's Theorem (**semi-linearity**)

Derivation trees, denotation of the generated graphs by terms,

Upper bounds to tree-width and clique-width.

Which properties do not hold as we could wish ?

**Answers** : The set of all (finite) graphs is neither HR- nor VR-equational.

Not even is the set of all square grids (planar graphs of degree 4)

Parsing is sometimes NP-complete.

Comparison of the two classes :

$$\text{Equat}(\mathbf{HR}) \subseteq \text{Equat}(\mathbf{VR})$$

= sets in  $\text{Equat}(\mathbf{VR})$  , all graphs of which are without some fixed  $K_{n,n}$  as subgraph.

### 3. Recognizable sets : an algebraic definition

$F$  : a finite set of operations with (fixed) arity, called **a signature**

$\mathbf{M} = \langle M, (f_M)_{f \in F} \rangle$  : 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,  $h$  is the run of a **finite deterministic automaton**.

**REC(M)** = the recognizable subsets of  $M$  (with respect to the algebra  $\mathbf{M}$ )

For the algebras **HR** and **VR** that have infinite signatures :

we require that the congruence  $\approx$  is type preserving :

$G \approx H$  implies  $\tau(G) = \tau(H)$

it has finitely many classes **of each type**,

L is the union of finitely many classes.

We could also use many-sorted algebras, with  $\tau(G)$  as sort of G,  
because the type function has a homomorphic behaviour.

Two notions of recognizable sets on algebras **HR** and **VR**.

Which properties of recognizable sets **follow** from the algebraic setting ?

*Answers* : Closure under union, intersection and difference, inverse homomorphisms, inverse unary derived operations.

The **intersection of an equational set and a recognizable one is equational** (with effective constructions)

Which properties of recognizable sets **do not follow** algebraically ?

*Answers* : Closure under the operations of the algebras :  $//$ ,  $\oplus$ , the unary operations.

(**False for add-edg** but true for some harmless restriction of the use of this operation).

Which properties do not hold as we could wish or expect ?

*Answers* : Emptiness is **not** decidable (because of infinite signatures).

REC and EQUAT are **incomparable**

**Every set of square grids** is HR- and VR-recognizable.

Hence there are **uncountably** many recognizable sets  
and **no characterization by finite automata  
or logical formulas.**

## Inductive proofs and computations

Based on equations like the one that defines *Series-Parallel graphs* :

$$S = S // S \cup S \bullet S \cup e$$

“All series-parallel graphs are connected”

“All series-parallel graphs are planar”

“Number of directed paths from Entry to Exit in a given series-parallel graph”

Sometimes, **auxiliary properties and / or functions** are necessary.

**Recognizability** means “**finitely** auxiliary properties suffice”

From a monadic second-order expression of a property, one can (theoretically) **generate** a finite set of auxiliary monadic second-order properties, and need not have to **guess and check**.

*Inductive computation* : Test of 2-colorability for series-parallel graphs

Not all series-parallel graphs are 2-colorable (see  $K_3$ )

$G, H$  2-colorable does not imply that  $G//H$  is 2-colorable (because  $K_3 = P_3//e$ ).

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

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

For every SP-term  $t$ , we can compute, by running a finite **deterministic bottom-automaton** on  $t$ , the pair of Boolean values (**Same**( $\text{Val}(t)$ ), **Diff**( $\text{Val}(t)$ )).

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

## Recognizability and inductive sets of properties

*Definition* : A 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 :

$$p(f_{\mathbf{M}}(a,b)) = B[\dots, q(a), \dots, q'(b), \dots] \text{ for all } a \text{ and } b \text{ in } \mathbf{M}$$

(here  $q, q' \in P$  ,  $q(a), \dots, q(b) \in \{True, False\}$  ) .

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

Inductive sets formalize the notion of “auxiliary properties” useful to get an inductive proof.

## Inductive sets of properties and automata on terms

The simultaneous computation of  $m$  inductive properties can be implemented by a finite deterministic bottom-up automaton with  $2^m$  states running on terms  $t$ .

This computation takes time  $O(|t|)$ : the key to fixed-parameter tractable algorithms

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

Membership of an element  $m$  of  $\mathbf{M}$  in a recognizable set  $L$  can be tested by such an automaton on any term  $t$  in  $T(F)$  defining  $m$   
(in some term if  $L$  is equational, i.e. “context-free”).

## 4. Monadic Second-Order (MS) Logic

*A logical language which specifies inductive properties and functions*

= First-order logic on power-set structures

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

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

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

*Non connectivity :*

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

*3-colorability :*

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

*Definition* : A set  $L$  of words, or trees, or graphs or relational structures is **Monadic Second-Order (MS) definable** if it is the set of finite models of an MS sentence  $\varphi$  (a formula without free variables).

$$L = \{ S \mid S \text{ finite, } S \models \varphi \} \text{ for a fixed MS formula } \varphi$$

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 **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.

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

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

(3) A set of finite graphs is HR-recognizable  $\Leftarrow$  it is  $MS_2$ -definable

*Proofs:* (1) Doner, Thatcher, Wright, (1968 -1970).

(2, 3) Can be proved simultaneously in 2 ways : Either by using the **Feferman-Vaught paradigm**, saying that the validity of an MS formula in the disjoint union of two structures can be deduced from those of **finitely many auxiliary** formulas of no larger quantification height in each of the two structures. This is **inductivity / recognizability**.

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

# Consequences of the Recognizability Theorem

## Algorithms

**Fixed-parameter *linear*** algorithms for checking **MS<sub>2</sub>** properties on graphs of tree-width  $\leq k$ ; finding a tree-decomposition is possible in linear time (Bodlaender, 1996)

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

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

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

## Language Theory extended to graphs

One can **filter out** from HR- or VR-equational sets the graphs which do not satisfy given MS<sub>2</sub>- or MS-properties and obtain HR- or VR-equational sets.

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

## Decidability of logical theories

The **MS<sub>2</sub>-theory** of the set of graphs of tree-width  $\leq k$  is **decidable**.

(Is some given formula true in all graphs ?)

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

## 5. Monadic second-order transductions

Transformations of graphs or, more generally of relational structures specified by MS (or  $MS_2$ ) formulas.

**Two types** : MS-transductions, for graphs = (vertices, adjacency relation)

$MS_{2,2}$ -transductions, for graphs represented by **incidence graphs** = (vertices and edges, incidence relation)

**Results :**

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

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

Preserve the decidability of **MS**- (or  $MS_2$ -) theories.

## Definition

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

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

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

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

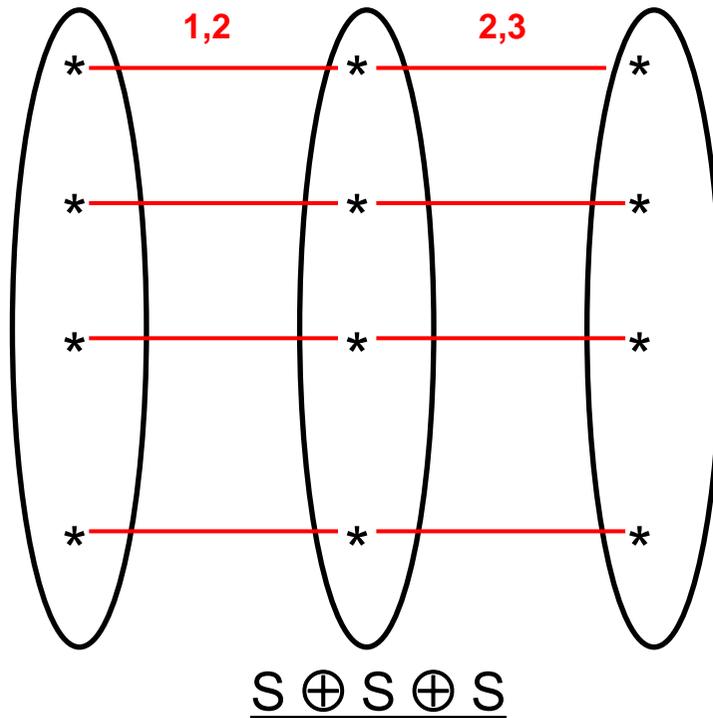
*Examples* :  $(G, \{u\}) \longmapsto$  the connected component containing  $u$ .

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

*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.

General case : T is defined in this way inside

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



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

Example of an MS transduction (without parameters) : The *square* mapping  $\delta$  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  $p_1, p_2$ .

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 :

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

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 information to describe  $\tau(S)$  ; the MS properties of  $\tau(S)$  are expressible by MS formulas in  $S$  ).

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

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

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

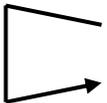
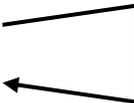
**Transitive closure** is an MS-transduction that is **not**  $MS_{2,2}$

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

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

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

## 6. Robustness results : Preservation and generation

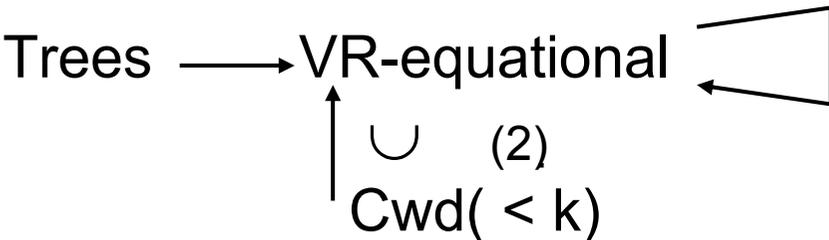
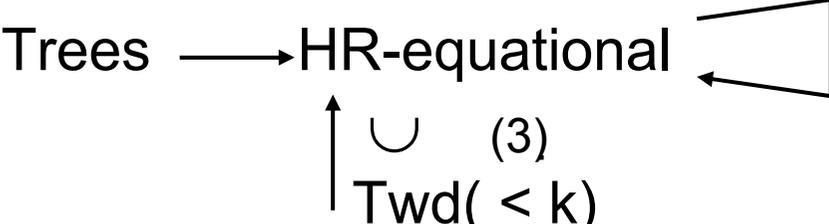
<b>Words : Rational transductions (= inverse rational transductions)</b>	
REC 	Dyck lang. $\rightarrow$ Context-free (trees) 
<b>Inverse MS transductions</b>	<b>Direct MS transductions</b>
 MS-def. $\subset$ VR-recog.  (1)	Trees $\rightarrow$ VR-equational  $\cup$ (2) $\uparrow$ Cwd( $\leq k$ )

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</sub> transductions	Direct MS <sub>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 disjoint union and transformations expressed by formulas without quantifiers.

Edge-complement, relabellings, edge-creation operations (cf. the definition of clique-width) are typical examples of quantifier-free definable operations on graphs.

Proof sketch : Every MS transduction is the composition of MS transductions of 3 types :

- Copy<sub>k</sub>
- Parameterless and noncopying transduction
- Guessing unary relations

We can prove that recognizability is preserved by the inverses of transductions of each type.

Copy<sub>k</sub> :

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

Disjoint union with binary relations  $Y_{i,j}$  for  $1 \leq i < j \leq k$  defined as  
 $\{(x,y) \mid x \text{ is the } i\text{-copy, } y \text{ is the } j\text{-copy of some } u \text{ in } D_S\}$

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, REC is preserved under inverse homomorphisms.

## 7. 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 elements 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 will 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  $C < E < D$

Consequence of results by Robertson and Seymour :

$\{\bullet\} < \text{Paths} <_c \text{Trees} <_c \text{Square grids}$

that can **encode** respectively (by MS<sub>2,2</sub>-transductions) :  
finite sets, sets of bounded path-width,  
sets of bounded tree-width, all sets of graphs.

- Questions* :
1. What is below Paths ?
  2. What about MS transductions ?
  3. What about countable graphs ?

Answer to question 1 (A. Blumensath and B. C., 2008, submitted)

$\{\bullet\} <_c T_1 <_c \dots T_n <_c T_{n+1} <_c \dots < \text{Paths} <_c \text{Trees} <_c \text{Square grids}$

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

$T_n$  encodes the classes of graphs having tree-decompositions of height at most  $n$  and width at most  $k$  (for each  $k$ ). Close links with **tree-depth** (Nesetril, Ossona de Mendez).

## 8. 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)
- B) If a set of graphs **contains all  $k \times k$ -grids**, its  $MS_2$  satisfiability problem is **undecidable**
- C) If  $C$  has **decidable  $MS_2$  satisfiability** problem, so has  $\text{Minors}(C)$ ,  
because  $C \longrightarrow \text{Minors}(C)$  is an  $MS_{2,2}$  transduction.

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

*Proof of Courcelle-Oum's Theorem :*

D) Equivalence between the cases of all (directed and undirected) graphs and bipartite undirected graphs.

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  $v$

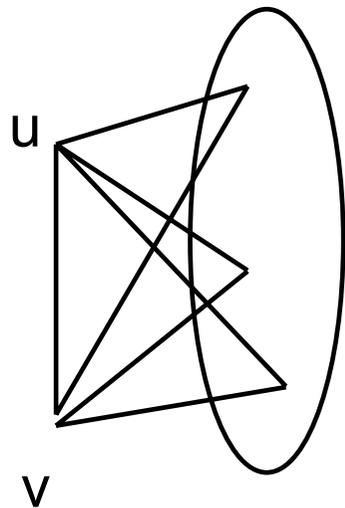
*Local equivalence* ( $\approx_{loc}$ ) = transitive closure of local complementation  
(at all vertices)

*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<sub>2</sub>MS** transduction.

Why is the even cardinality set predicate necessary ?



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

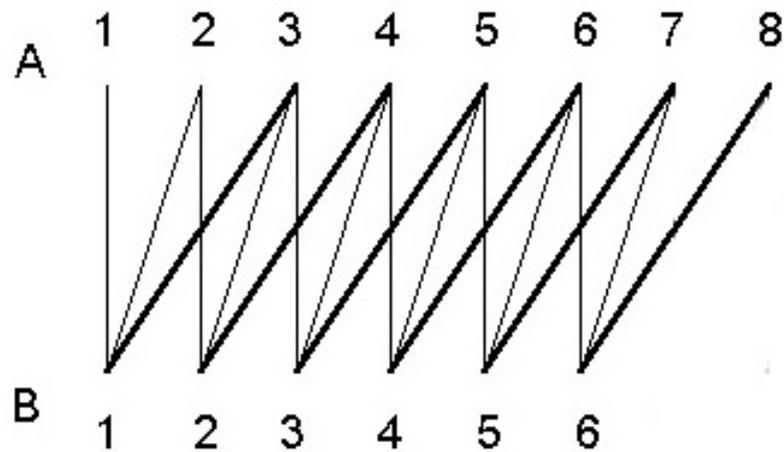
$u$  is linked to  $v$  in  $G * X$

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

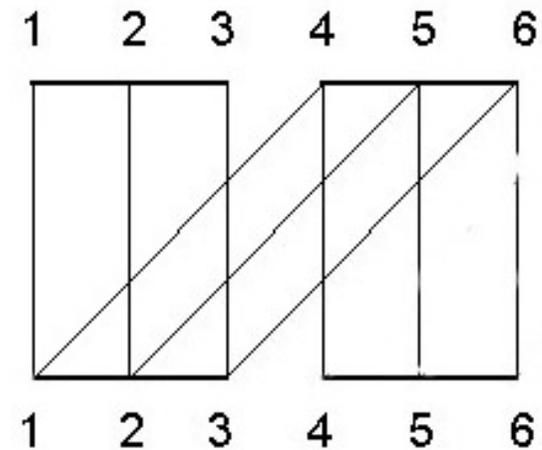
$G$        $Y$

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

## 9. A few open questions

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

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

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

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

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

More important (personal opinion) :

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

Other questions (and answers) are in the proceedings.

Thanks for your attention.