



# Monadic second-order logic and recognizability

*Bruno Courcelle*

Université Bordeaux 1, LaBRI, and  
Institut Universitaire de France



*References* : Book in progress,

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

<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.

# 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.

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

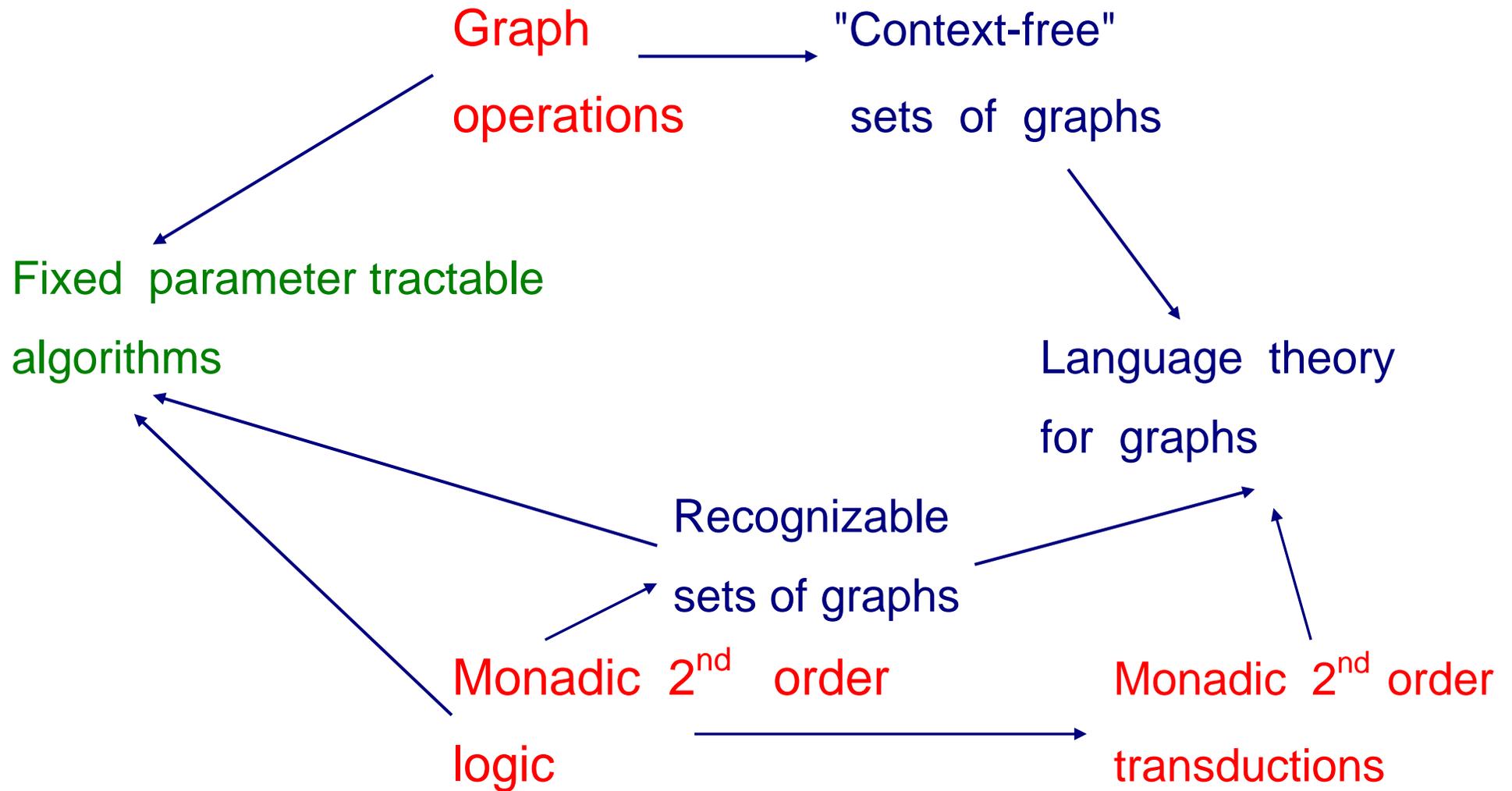
6. Monadic second-order transductions.

7. Preservation of recognizability by inverse monadic second-order transductions.

8. Preservation of recognizability by quantifier-free definable operations.

### *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>

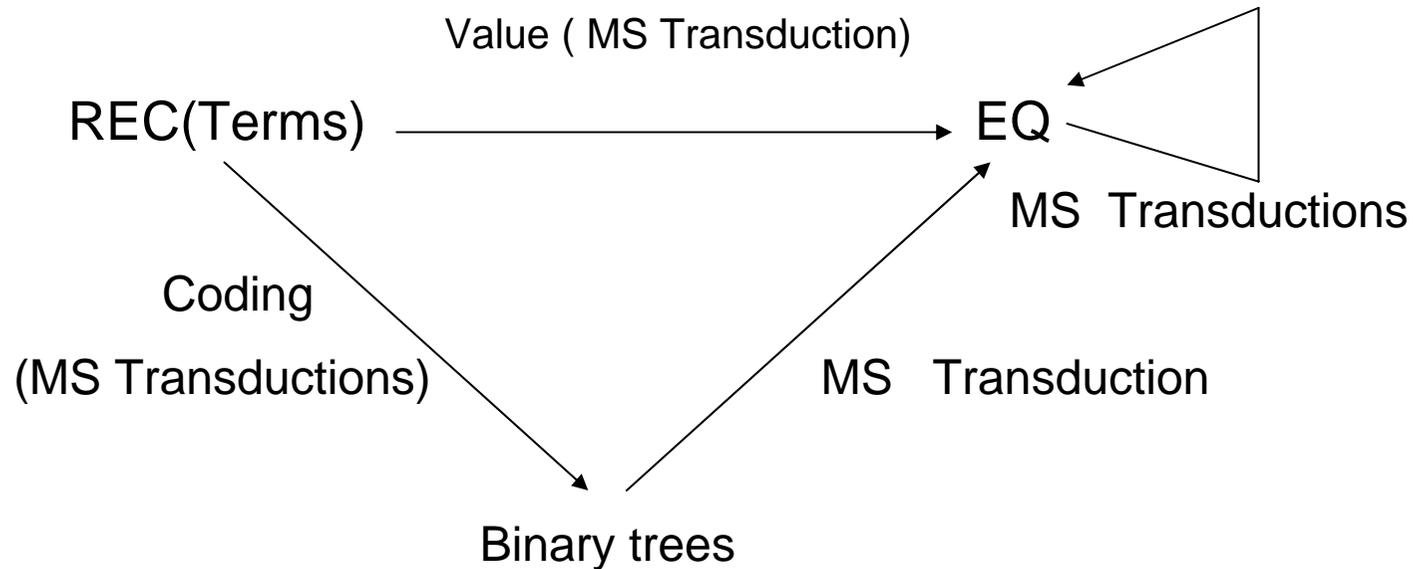
Signatures for graph algebras :

*HR*: graphs and hypergraphs with “sources”

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

*VR*<sup>+</sup>: *VR* with (QF) **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:

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

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

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

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

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

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

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*.

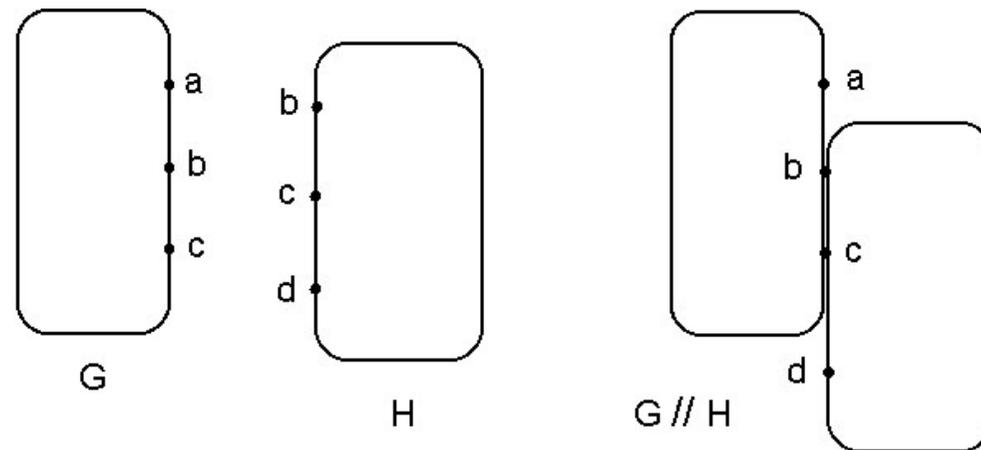
## 4. The graph algebras VR and HR

*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* :

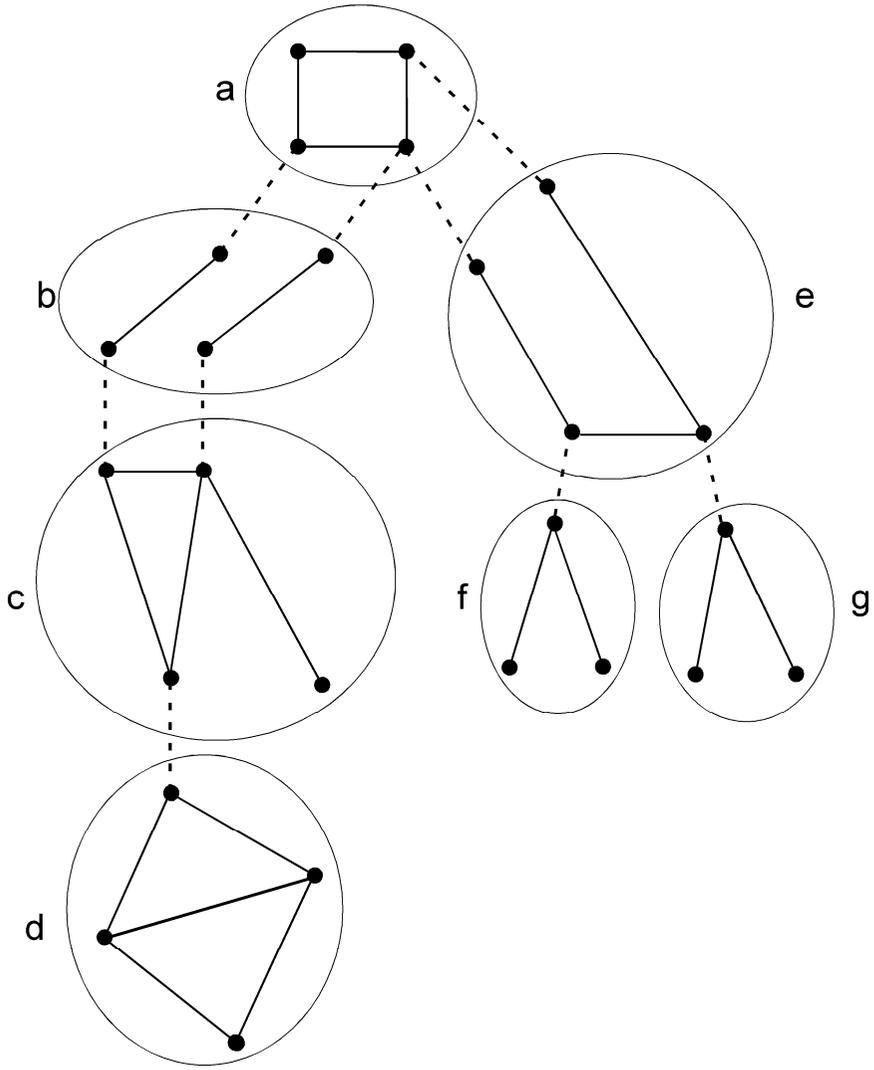
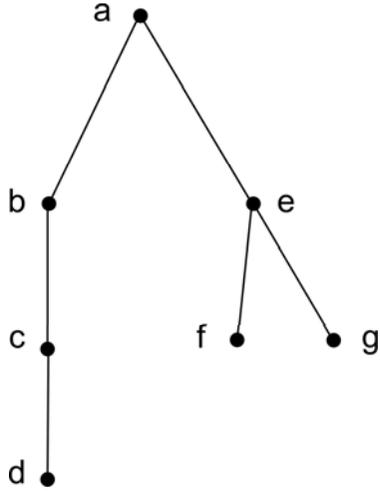
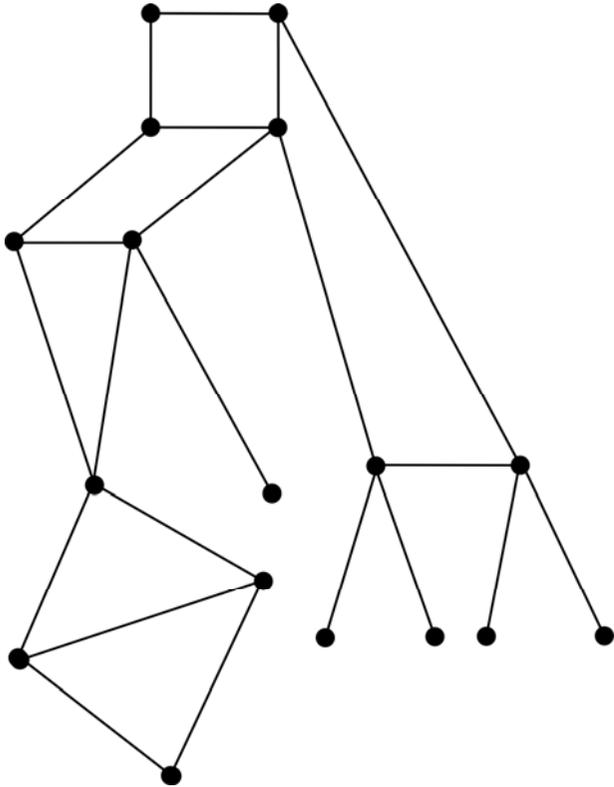
$\text{Ren}_{a \rightarrow b}(G)$  exchanges source names  $a$  and  $b$   
(replaces  $a$  by  $b$  if  $b$  is not the name of a source)

*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.*

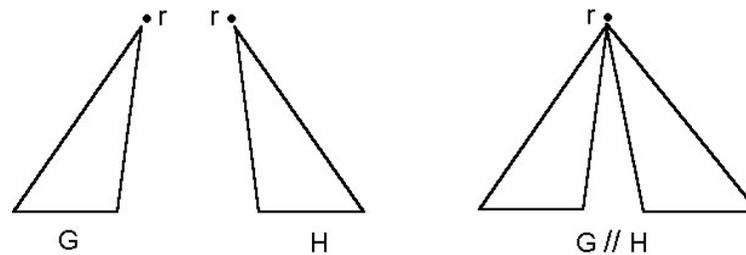
More precise algebraic framework : **a many sorted algebra where each finite set of source labels is a sort.** The above operations are overloaded.

# 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  $Forget_a$ .

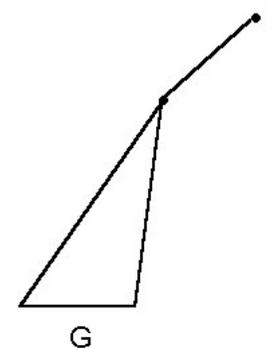
*Example :* Trees are of tree-width 1, constructed with two source labels,  $r$  (root) and  $n$  (new root): Fusion of two trees at their roots :



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

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

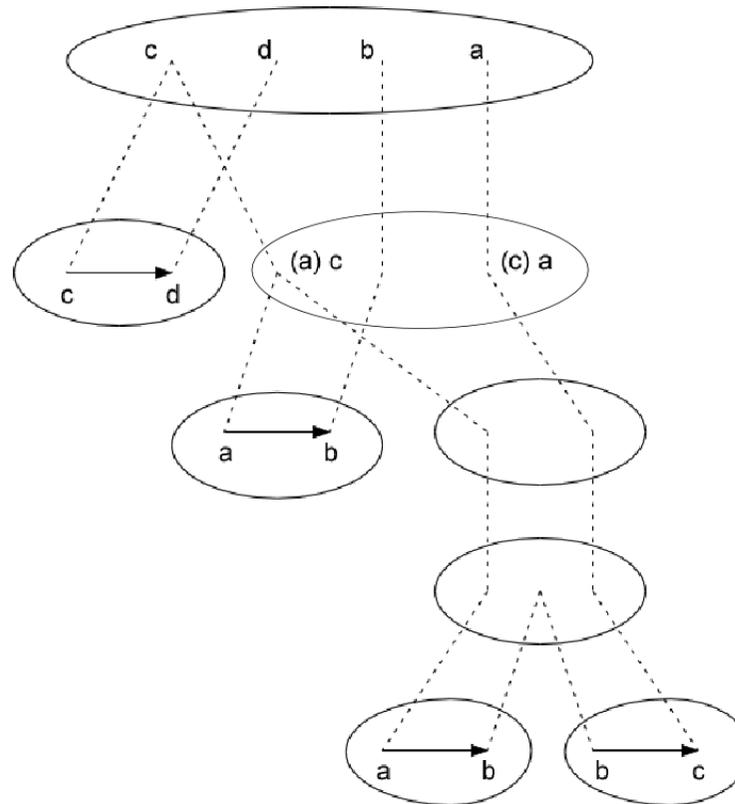
$$Ren_{n \leftrightarrow r} (Forget_r (G // E))$$



# From an algebraic expression to a tree-decomposition

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

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.

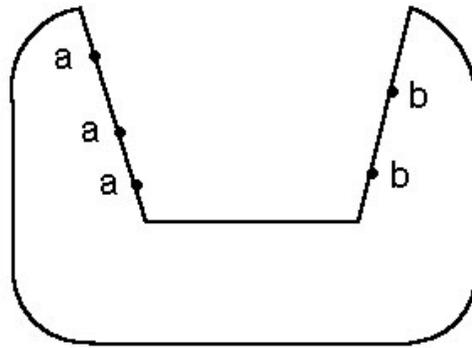
$k$  labels :  $a, b, c, \dots, h$ . Each vertex has one and only one label ;

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

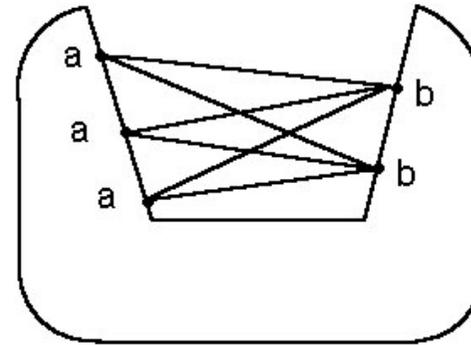
*One binary operation: disjoint union :  $\oplus$*

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.



$G$



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

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.

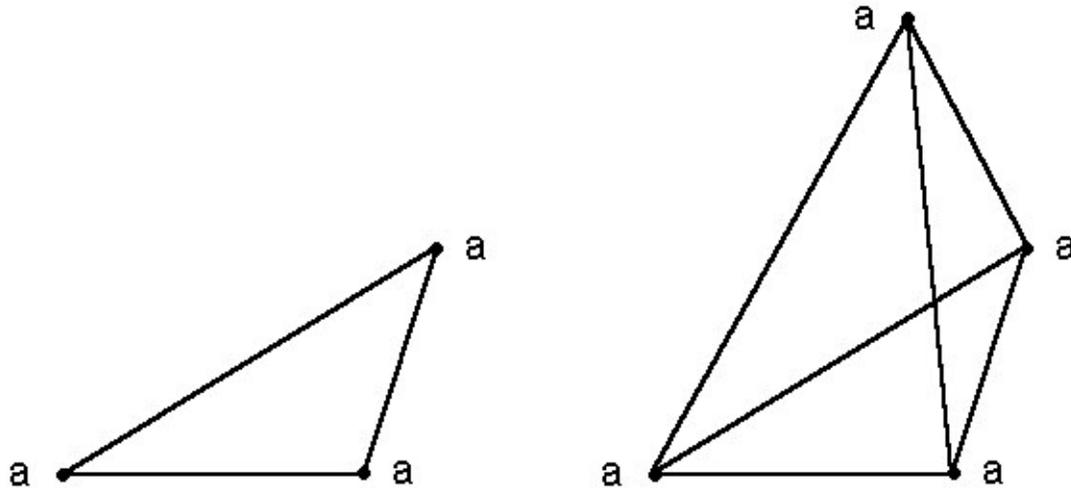
*Definition:* A graph  $G$  has **clique-width**  $\leq k \Leftrightarrow$  it can be constructed from basic graphs by means of  $k$  labels and the operations  $\oplus$ ,  $Add-edge_{a,b}$  and  $Relab_{a \rightarrow b}$

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.

*Example* : Cliques have clique-width 2.



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

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

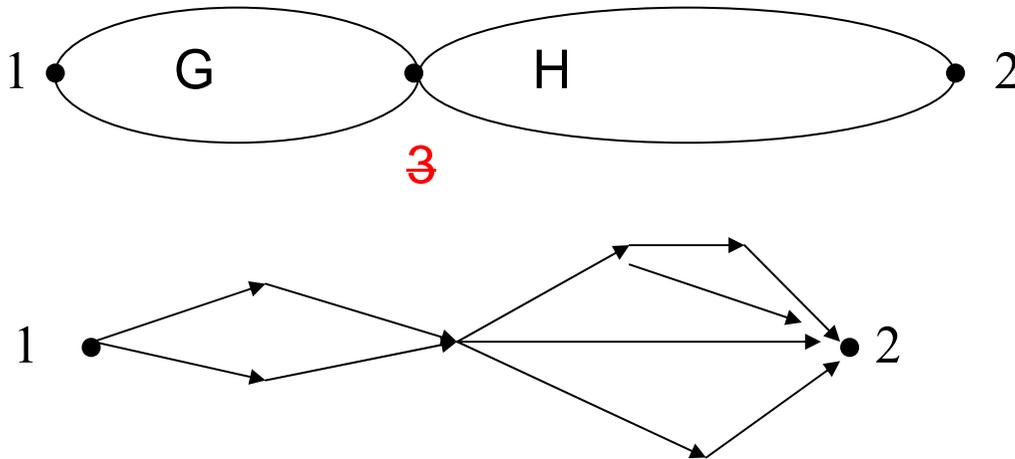
$$\begin{aligned}
 G \otimes H &= \text{Relab } \mathbf{b} \rightarrow \mathbf{a} (\text{Add-edg}_{\mathbf{a},\mathbf{b}} (G \oplus \text{Relab } \mathbf{a} \rightarrow \mathbf{b} (H))) \\
 &= G \oplus H \text{ with "all edges" between } G \text{ and } H.
 \end{aligned}$$

# Inductive proofs and 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 :

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

*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.

2) 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.

# 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

- 1) The existence of properties forming a *finite* 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 effectively 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 (number of satisfying tuples, distance)
- 5) 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  $m$  (in *some* for an *equational*, i.e. context-free set).

## 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 :*

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

*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.

*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, one of them based on the **Fefermann-Vaught paradigm**, saying that the validity of an MS formula in the disjoint union of two structures can be deduced from those of formulas of no larger quantification height in each of the two structures.

## 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 :

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

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

The fundamental property of MS transductions :

$$\begin{array}{ccc} S & \longrightarrow & \tau(S) \\ & & \longleftarrow \\ \tau^\#(\psi) & & \psi \end{array}$$

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.

## 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 relational structures (hypergraphs) is QF-equational iff it is the image of **(all)** binary trees under an MS-transduction. (**QF =  $\oplus$  and quantifier-free operations**)  
**QF-equational sets** are stable under MS-transductions.

4) Proof of results by Seese, Courcelle and Oum saying that graphs classes with decidable MS<sub>2</sub> and C<sub>2</sub>MS theories have **bounded tree-width and clique-width**.

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

More precisely, QF-recognizability of sets of relational structures is preserved.

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

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

1) Copy<sub>k</sub> :

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

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

Hence Copy<sub>k</sub> is “almost” a homomorphism, REC is preserved under inverse homo's.

## 2) Forgetting unary relations :

We let  $\text{fgt}_{\Pi} : \text{STR}(\Sigma \cup \Pi) \rightarrow \text{STR}(\Sigma)$  forget  $\Pi$ , a finite set of unary relations.

Its inverse consists in **guessing** the relations in  $\Pi$ , this can be done by means of **parameters** of an MS transduction.

Lemma : If  $L \in \text{REC}$ , then  $\text{fgt}_{\Pi}(L) \in \text{REC}$ .

*Proof* : From a congruence  $\equiv$  for  $L$ , we define the equivalence :

$$S \cong T \quad \text{iff} \quad \{ [U]_{\equiv} / \text{fgt}_{\Pi}(U) = S \} = \{ [U]_{\equiv} / \text{fgt}_{\Pi}(U) = T \}$$

It is finite, saturates  $\text{fgt}_{\Pi}(L)$ . It is a congruence : we use the fact that

$$\begin{aligned} S \oplus T = \text{fgt}_{\Pi}(W) \quad \text{iff} \quad & \text{there exist } S', T' \text{ such that } W = S' \oplus T', \\ & S = \text{fgt}_{\Pi}(S'), \text{ and } T = \text{fgt}_{\Pi}(T'), \end{aligned}$$

and a similar observation for unary operations.

### 3) Parameterless noncopying MS transductions.

We let  $L \in \text{REC}(\Gamma)$  and  $\tau$  be a parameterless noncopying MS transduction :

$\text{STR}(\Sigma) \rightarrow \text{STR}(\Gamma)$  of quantifier-height  $k$

We prove that  $\tau^{-1}(L) \in \text{REC}(\Sigma)$

From a congruence  $\equiv$  for  $L$ , we define, on each set  $\text{STR}(\Delta)$ , the equivalence :

$S \cong T$  iff  $\text{tp}_k(S) = \text{tp}_k(T)$  and  
for every parameterless noncopying MS transduction  
 $\mu : \text{STR}(\Delta) \rightarrow \text{STR}(\Gamma)$  of quantifier-height at most  $k$ , we have  $\mu(S) \equiv \mu(T)$ .

( $\text{tp}_k(S)$  is the MS theory of  $S$  of quantifier height at most  $k$ .)

It is finite, saturates  $\tau^{-1}(L)$  (because  $\tau$  is one of the considered transductions  $\mu$ ).

Claim : It is a congruence.

a) We consider  $f$  unary, quantifier-free. Let  $S \cong T$  :

$$\text{tp}_k(f(S)) = f^@(\text{tp}_k(S)) = f^@(\text{tp}_k(T)) = \text{tp}_k(f(T))$$

For every  $\mu$ ,  $\mu \circ f$  is a parameterless noncopying MS transduction of quantifier-height  $k$ :

$$\mu(f(S)) = \mu \circ f(S) \equiv \mu \circ f(T) = \mu(f(T)). \text{ Qed}$$

b) Remains  $\oplus$  : Let  $S \cong T$ ,  $S' \cong T'$ , we want  $S \oplus S' \cong T \oplus T'$ .

We have  $\text{tp}_k(S \oplus S') = \text{tp}_k(T \oplus T')$  (by the Fefermann-Vaught paradigm)

For every  $\mu$  :  $\mu(S \oplus S') = f[\text{tp}_k(S \oplus S')](M_{k,m}(S \oplus S'))$  ( $m = \text{MaxArity}(\Gamma)$ )

$$= f[\text{tp}_k(S \oplus S')](\oplus^\$[\text{tp}_k(S), \text{tp}_k(S')](M_{k,m}(S), M_{k,m}(S'))))$$

and similarly for  $\mu(T \oplus T')$

Since  $M_{k,m}$  is an MS transduction of quantifier-height  $\leq k$  :

$$M_{k,m}(S) \equiv M_{k,m}(T), M_{k,m}(S') \equiv M_{k,m}(T')$$

Also :  $\oplus^\$[\text{tp}_k(T), \text{tp}_k(T')] = \oplus^\$[\text{tp}_k(S), \text{tp}_k(S')]$

$$\text{And } f[\text{tp}_k(S \oplus S')] = f[\text{tp}_k(T \oplus T')]$$

Since  $\equiv$  is a QF-congruence, and  $f[p]$ ,  $\oplus^\$$  are in QF :

$$\mu(S \oplus S') \equiv \mu(T \oplus T'), \text{ hence } S \oplus S' \cong T \oplus T'$$

## Notation and facts :

Let  $S$  in  $\text{STR}(\Sigma)$ ,  $a_1, \dots, a_n$  in  $D_S$  (the *domain* of  $S$ ).

$$\begin{aligned} \text{tp}_k(S, a_1, \dots, a_n) &= \\ &\{ \varphi / S \mid \models \varphi(a_1, \dots, a_n), \varphi \in \text{MS}(\Sigma, x_1, \dots, x_n) \text{ of quantifier-height } \leq k \} \\ &\in P(\text{MS}_k(\Sigma, x_1, \dots, x_n)) \\ &= \text{(k,n)-Types} \end{aligned}$$

$\text{tp}_k(S, a_1, \dots, a_n)$  is a **local information** relative to  $a_1, \dots, a_n$  in  $S$ .

$\text{tp}_k(S)$  is a **global information** relative to  $S$ .

**Annotation** :  $S \mid \rightarrow M_{k,m}(S) = (D_S, (T_p)_{p \in \text{(k,n)-Types}, n \leq m})$ .

$$T_p(a_1, \dots, a_n) : \Leftrightarrow \text{tp}_k(S, a_1, \dots, a_n) = p.$$

$$M_{k,m}(S) = S + \text{local information}$$

**Facts** :  $M_{k,m}$  is an MS transduction of quantifier-height  $k$ .

$M_{0,m}$  is a quantifier-free transduction.

The inverse of  $M_{k,m}$  is a quantifier-free transduction ;  
it is functional because  $M_{k,m}$  is injective.

## Homomorphic properties of $M_{k,m}$

Proposition :

- 1)  $M_{k,m}(S \oplus T) = \oplus^{\$} [\text{tp}_k(S), \text{tp}_k(T)] (M_{k,m}(S), M_{k,m}(T))$
- 2) For  $g$  quantifier-free :  $M_{k,m}(g(S)) = g^{\$} (M_{k,m}(S))$

where  $g^{\$}$ ,  $\oplus^{\$} [p, p']$  are compositions of QF operations.

*Proof sketch :* 1) By the Fefermann-Vaught paradigm :

$$S \oplus T \models \varphi(a_1, \dots, a_n, b_1, \dots, b_q)$$

iff for some  $i$  :

$$S \models \psi_i(a_1, \dots, a_n) \text{ and } T \models \theta_i(b_1, \dots, b_q)$$

where  $\psi_i$  and  $\theta_i$  are MS formulas of quantifier-height  $\leq$  that of  $\varphi$ , and that only depend on  $\varphi$ . Hence :

$$M_{k,m}(S \oplus T) = f(M_{k,m}(S) \oplus M_{k,m}(T))$$

where  $f$  redefines the types in  $S$  and in  $T$  (in terms of  $\text{tp}_k(T)$  and  $\text{tp}_k(S)$ ) and creates appropriate relations for the types concerning both  $S$  and  $T$  (using  $\text{add}_{U,V,W}$ ).

- 2) By the fact that :  $\text{tp}_k(g(S), a_1, \dots, a_n) = g^{\textcircled{a}}(\text{tp}_k(S, a_1, \dots, a_n))$  for some mapping  $g^{\textcircled{a}} : (k,n)\text{-Types} \rightarrow (k,n)\text{-Types}$  (by using backwards translation relative to  $g$ ).

We let  $g^{\$}$  replace  $g$  by  $p$  whenever  $g^{\textcircled{a}}(q) = p$

## Factorization :

Through  $M_{k,m}$  annotations, MS transductions of quantifier-height  $\leq k$  reduce to quantifier-free ones :

If  $g : \text{STR}(\Sigma) \rightarrow \text{STR}(\Gamma)$  is an MS transduction of quantifier-height  $\leq k$ , and  $m = \text{MaximumArity}(\Gamma)$  :

$$g(S) = f [\text{tp}_k(S)] (M_{k,m}(S))$$

where  $f [p] : \text{STR}(\Sigma_{k,m}) \rightarrow \text{STR}(\Gamma)$  is quantifier-free.

Easy construction, using renamings of the type relations and deletions of elements.

Initial question : Which graph operations preserve recognizability ?

*Facts :*

- 1) The class of **recognizable sets** of an algebra  $M$  is **not preserved** under the operations of  $M$ . (The class of **equational sets** is).
- 2) The concatenation of two languages is. A proof not using automata (but finite congruences) can be done based on the fact that :  
 $uv = xy$  iff for some  $w$  :  $uw = x$  and  $v = wy$  or  $u = xw$  and  $wv = y$
- 3) By this type of argument, I could prove (Math. Struct. In Computer Science, 1994) that the class of HR-recognizable sets of graphs is closed under the HR operations.

*Question :* What about the VR algebra ?

The operation **add<sub>a,b</sub>** does **not** preserve recognizability.

a — b

a — b

.....

a — b

—————>

the graphs  $K_{n,n}$

Why ? this operation is not **invertible** unless we use edge set quantifications.

MS invertible operation :

$\text{relab}_a \rightarrow b$  : it fuses two sets of vertices ; to invert it suffices to guess the former set of vertices labelled by a.

This can be done by an MS transduction with **one set parameter**.

*Theorem* : If the inverse of a unary operation  $f$  is an MS transduction, then  $f$  preserves QF-recognizability.

Immediate consequence of the result with Achim (AB-BC-2004).

*For the VR-algebra.*

One can limit  $\text{add}_{a,b}$  to graphs that have no edges between  $a$ -vertices and  $b$ -vertices : for them this operation is MS invertible.

*Open question* : What about relational structures ? No such trick is known (to me).

## 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)

## 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* : 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 (IMHO) :

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