

Graph structuring and monadic second-order logic

Bruno Courcelle

Université Bordeaux 1, LaBRI

History: The 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

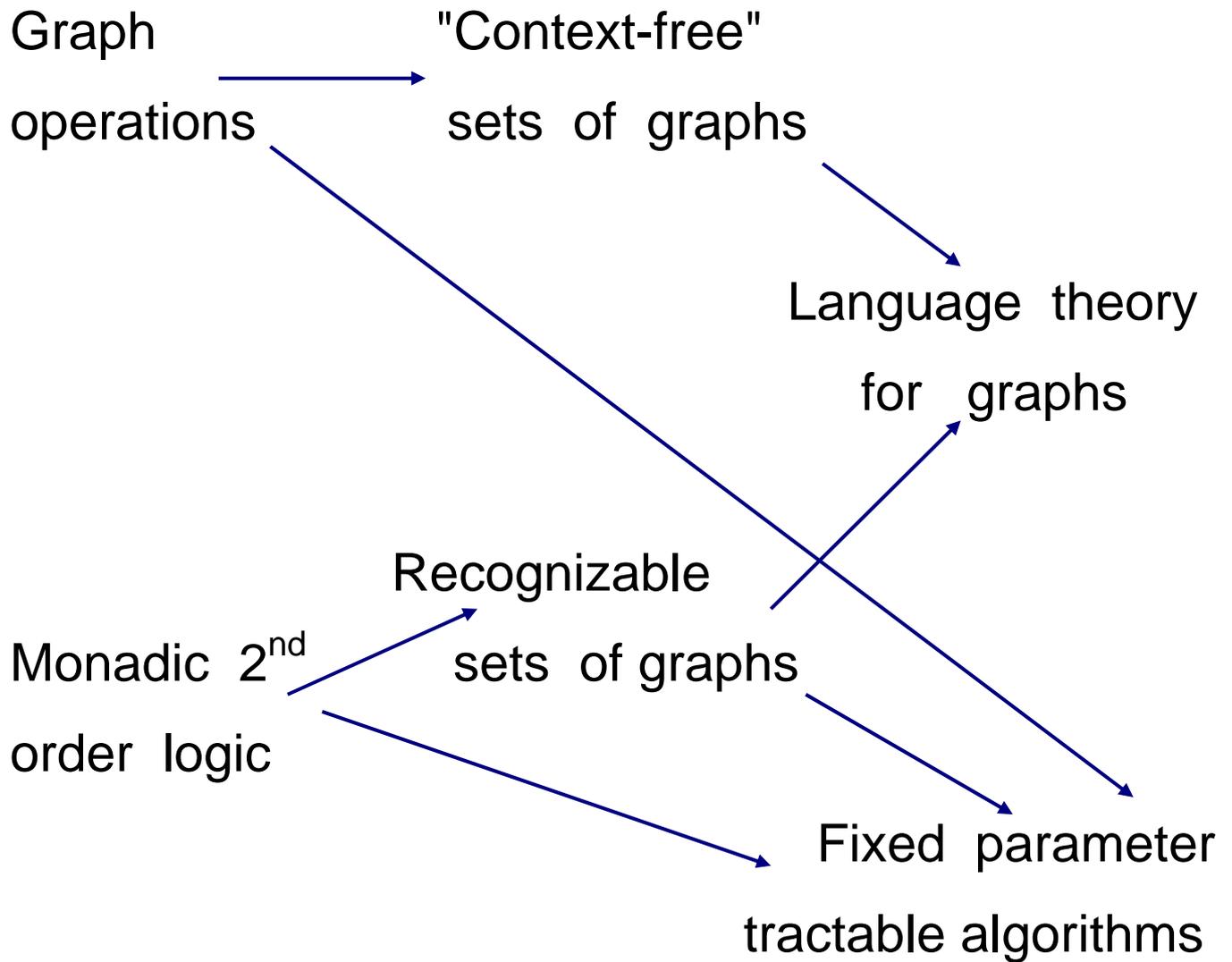
- 7 Closure and decidability properties ; generation of classes of graphs by monadic second-order transductions.

Monadic second-order logic and combinatorics

- 8 Seese's conjecture

Open questions

An overview chart :



1. Introduction

Key concepts of FLT and their extensions

| <i>Languages</i> | <i>Graphs</i> |
|--|---|
| Algebraic structure : monoid $(X^*, *, \varepsilon)$ | Algebras based on graph operations : $\oplus, \otimes, //$ quantifier-free definable operations 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 | Monadic Second-order transductions |

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 ⁻¹ |
| | | | MS-trans ⁻¹ |

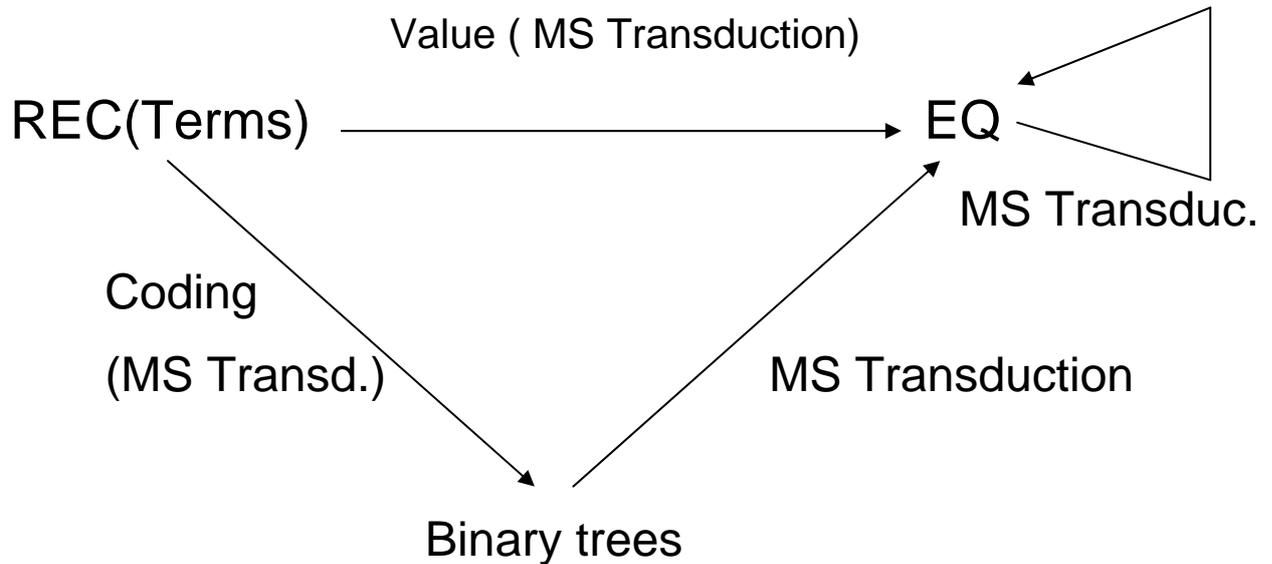
Signatures for graph algebras :

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)

Compare :

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, $D \subseteq A$.

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

Closure properties : 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 (means that each f in F has a type

$$s_1 s_2 \dots s_k \rightarrow s, \text{ with } s, s_i \in S)$$

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

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

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 we consider similarly systems of equations like:

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

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

where f is a binary operation, g, k, m are unary operations on graphs, a, b are basic graphs.

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 |
| | $Val(REC(Terms))$ | homomorphisms |
| REC | congruences | Boolean operations |
| | | homomorphisms ⁻¹ |

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

HR operations

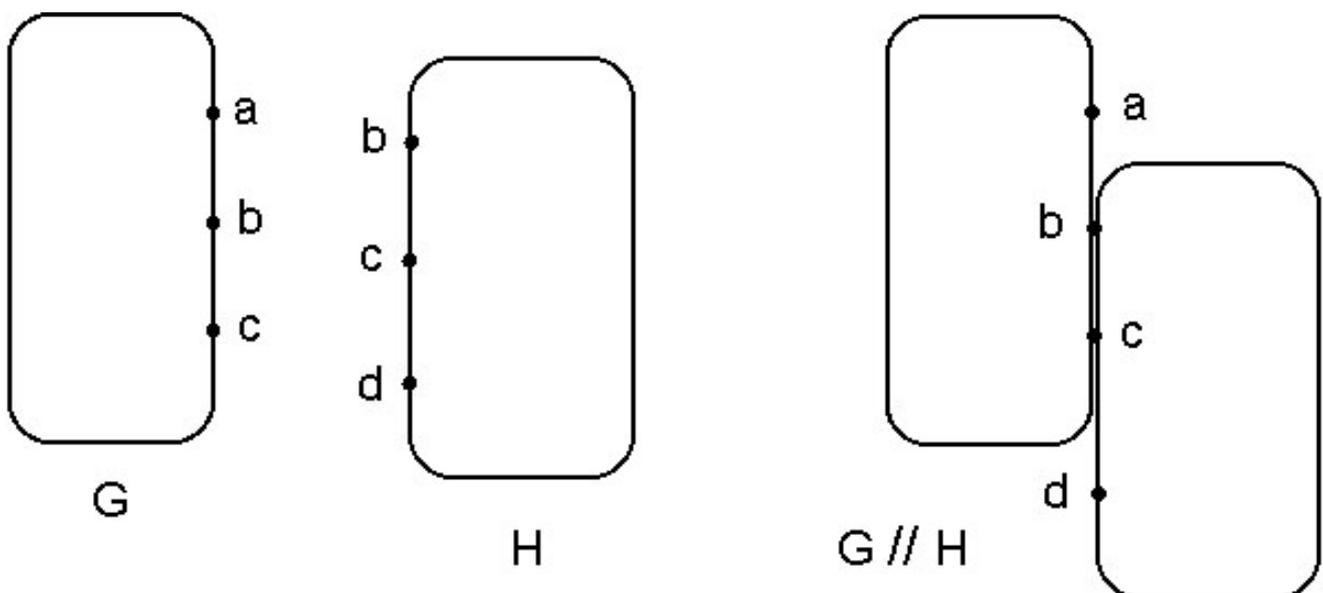
(Origin : **H**yperedge **R**eplacement hypergraph grammars ;
associated complexity measure : **tree-width**)

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

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

Source renaming :

$Rena_{a,b}(G)$ exchanges a and b
(replaces a by b if b is not a source)

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

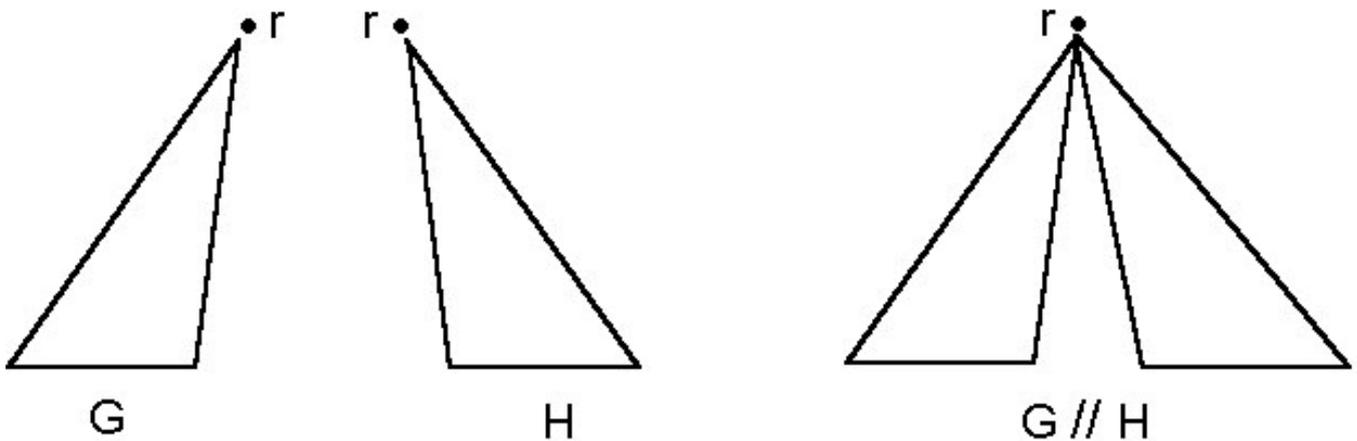
For dealing with hypergraphs one takes more nullary symbols for denoting hyperedges.

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):

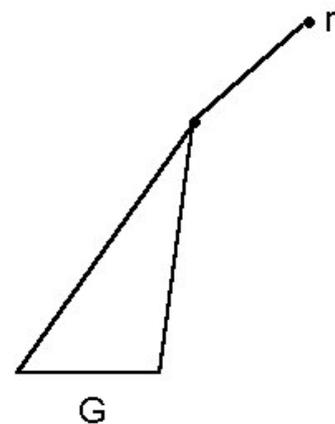
Fusion of two trees at their roots :



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

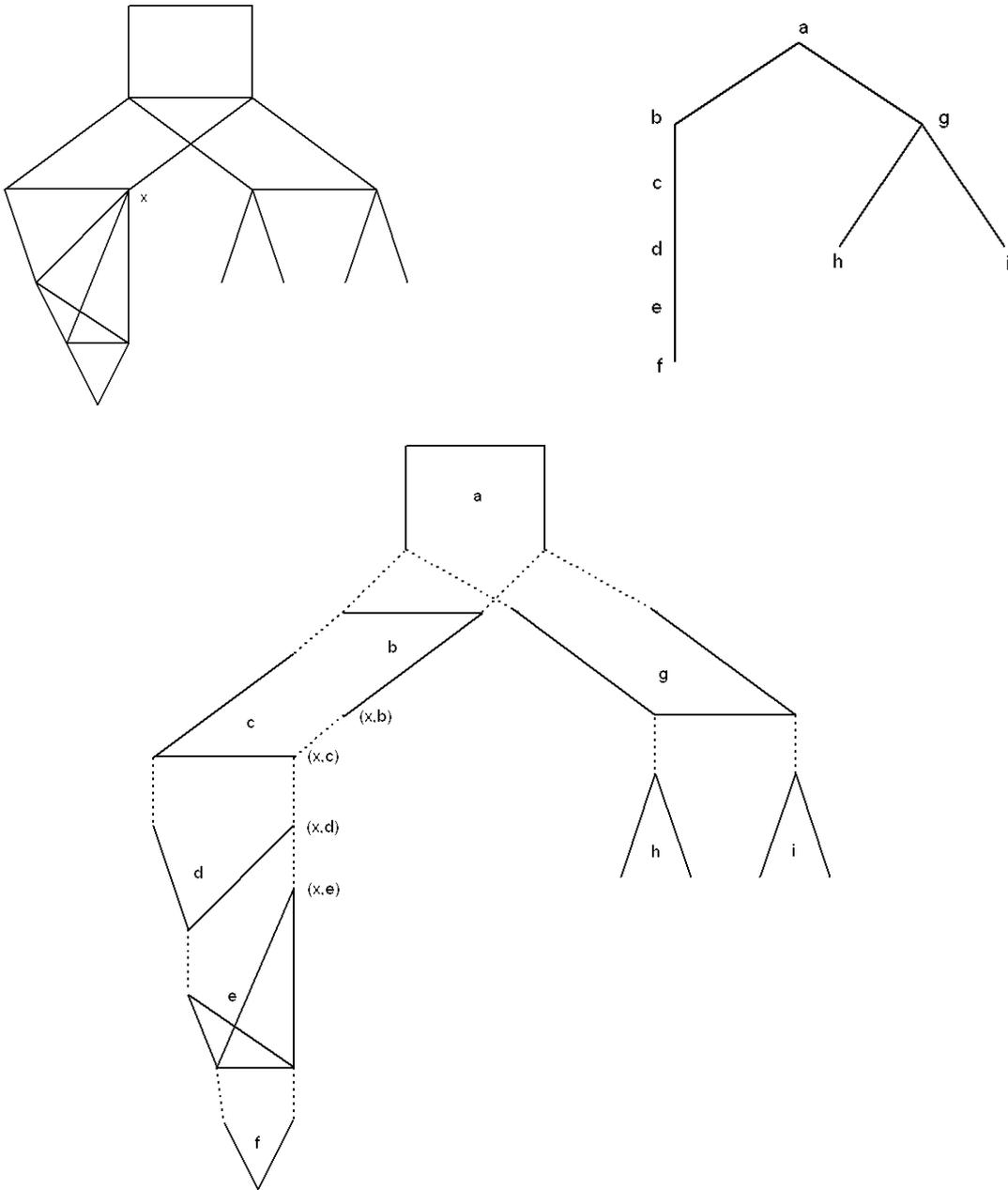
$$E = r \bullet \text{---} \bullet n$$

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



Tree-width

Tree-decomposition of width k : $k+1 = \text{max. size of a box}$



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

Trees have tree-width 1, K_n has tree-width $n-1$

The $n \times n$ grid has tree-width = n

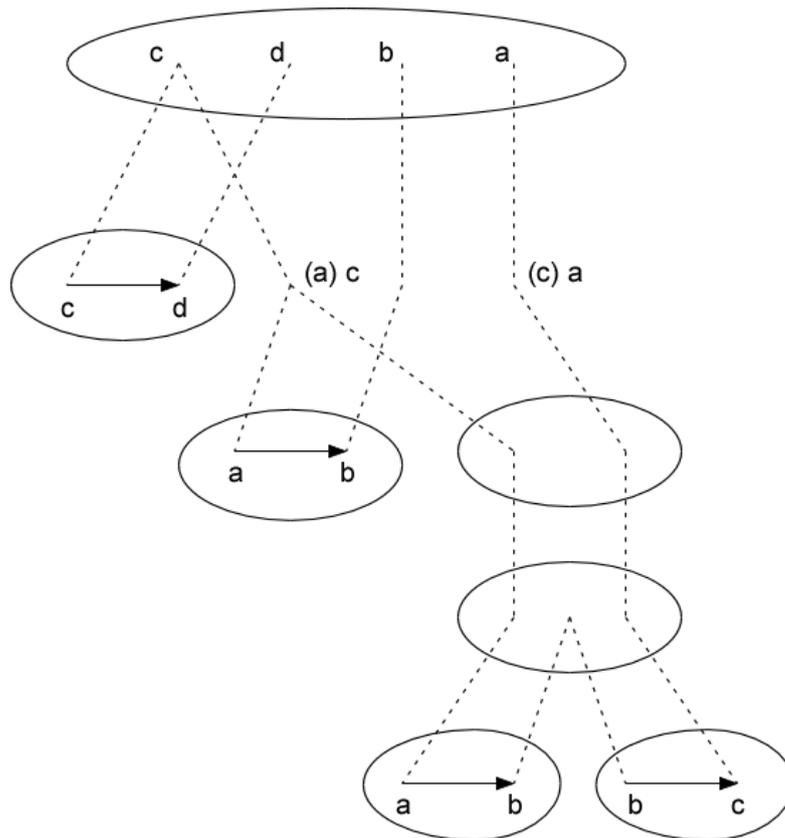
Planar graphs and cliques have unbounded tree-width.

Outerplanar graphs have tree-width at most 2.

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**)

Graphs are simple, directed or not.

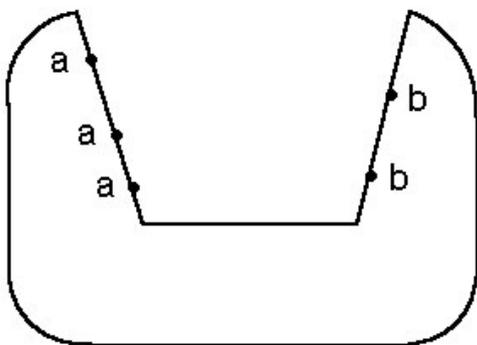
k labels : a, b, c, \dots, h .

Each vertex has one and only one label ; a label p may label several vertices, called the p -ports.

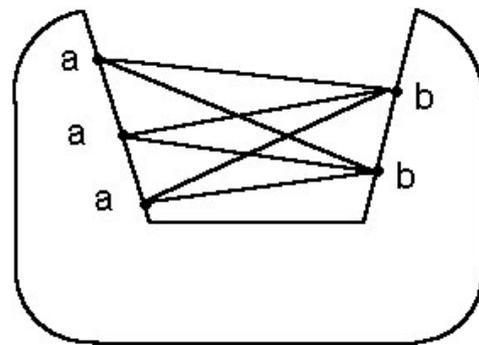
Binary operation: disjoint union \oplus

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

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



G



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

$Relab_{a,b}(G)$ 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 , $Add-edge_{a,b}$
and $Relab_{a,b}$

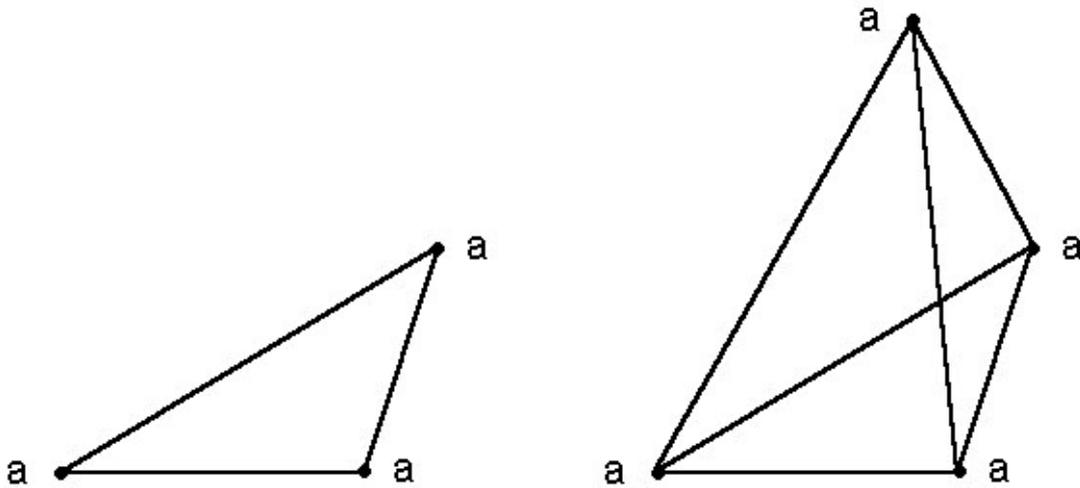
Its (exact) clique-width, **$cwd(G)$** , is the
smallest such k .

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) “Clique-width ≤ 3 ” is polynomial. The complexity (polynomial or NP-complete) of “Clique-width = 4” is unknown.

Example : Cliques have clique-width 2.



K_n is defined by t_n

$$t_{n+1} = \text{Relab}_{b,a}(\text{Add-edge}_{a,b}(t_n \oplus \mathbf{b}))$$

Another example : *Cographs*

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

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

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

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,

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 .

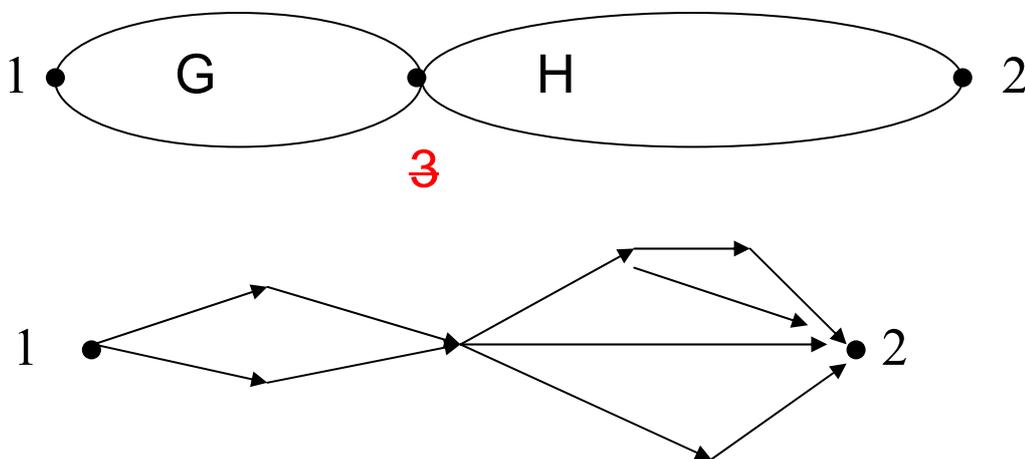
(by Bodlaender, 1996)

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 $G \bullet H$ connected, e is connected :

\Rightarrow All series-parallel graphs are connected.

2) It is not true that G, H planar implies $G//H$ 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

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

G, H 2-colorable does not imply that $G//H$ is 2-colorable.

$$(K_3 = P_3 // e).$$

One can check the property 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 : $\text{Diff}(e) = \text{True}$ $\text{Same}(e) = \text{False}$

$$\text{Same}(G//H) \Leftrightarrow \text{Same}(G) \wedge \text{Same}(H)$$

$$\text{Diff}(G//H) \Leftrightarrow \text{Diff}(G) \wedge \text{Diff}(H)$$

$$\text{Same}(G \bullet H) \Leftrightarrow (\text{Same}(G) \wedge \text{Same}(H)) \vee (\text{Diff}(G) \wedge \text{Diff}(H))$$

$$\text{Diff}(G \bullet H) \Leftrightarrow (\text{Same}(G) \wedge \text{Diff}(H)) \vee (\text{Diff}(G) \wedge \text{Same}(H))$$

We can compute for every SP-term t , by induction on the structure of t the pair of Boolean values ($\text{Val}(t)$ = graph value of t).

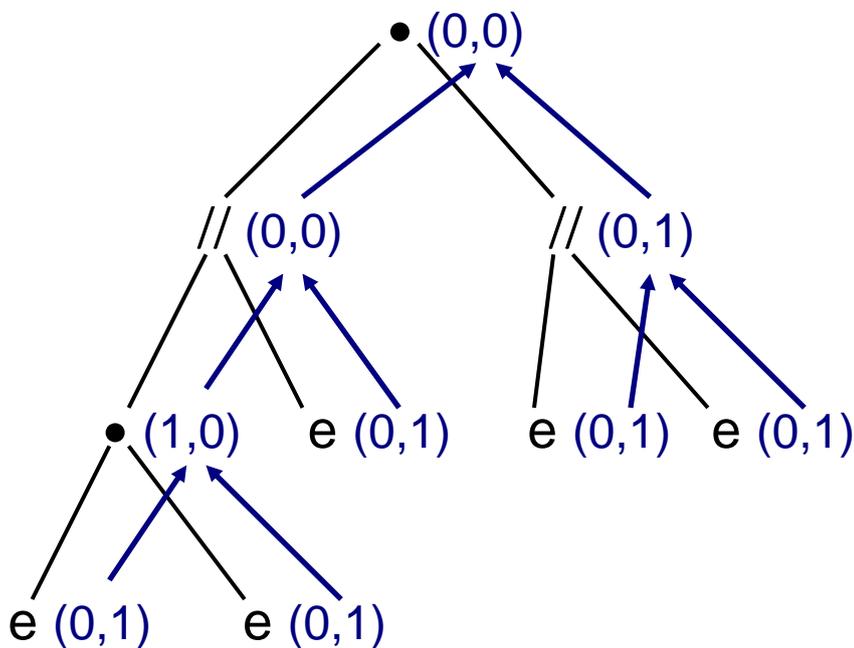
$$(\text{Same}(\text{Val}(t)), \text{Diff}(\text{Val}(t))).$$

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

Implementation by an automaton

States are pairs of Boolean values, 0 = False, 1 = True, representing $(\text{Same}(\text{Val}(t)), \text{Diff}(\text{Val}(t)))$, for t = the subterm processed.

Example :



The graph $\text{Val}(((e \bullet e) // e) \bullet (e // e))$ is not 2-colorable.

Important facts to be developed :

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 2^m -state automaton working on terms t . It takes **time** $O(|t|)$.

3) An inductive set of properties can be found (**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 (we know) a 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} . ($q(a), \dots, q(b) \in \{\text{True}, \text{False}\}$)

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 : $\hat{a}(m) \Leftrightarrow h(m) = a$.

Let p be such that $p(m) = \text{True} \Leftrightarrow h(m) \in C \Leftrightarrow m \in L$.

Properties $p, \hat{a}, a \in A$, are F -inductive.

If P is an inductive set of m properties, one can define an F -algebra structure on the set A of m -tuples of Booleans, such that the mapping $h : m \mapsto$ the m -tuple of Booleans is a homomorphism.

Inductive properties and automata on terms

If P is an inductive set of m properties, one can define a deterministic **automaton** on terms of $T(F)$ with set of states the m -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 : Bodlaender (for twd), Oum (for cwd).

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 $G = \langle V_G, \text{edg}_G(\cdot, \cdot) \rangle$, undirected

Non connectivity :

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

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

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

Non MS properties :

Equicardinality, bijections, automorphisms.

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

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

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

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

This is true for graph algebras HR, VR, VR^+ .

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

Recognizable \rightarrow automaton with states $1, \dots, n \rightarrow$ MS formula :

$$\exists X_1, \dots, X_n \text{ ("these sets define an accepting run")}$$

Conversely, induction on the structure of formula, using closure of recognizable sets under union, intersection, difference, **alphabetic homomorphisms**.

Basic facts for (2): Let F consist of \oplus and unary **quantifier-free definable operations** f (examples : relabellings, edge-complement).

For every MS formula φ of quantifier-height k , we have

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

such that : $f(S) \models \varphi \Leftrightarrow S \models f^\#(\varphi)$

Better, for φ with free variables X, Y, Z :

$$\text{Sat}(f(S), \varphi, X, Y, Z) = \text{Sat}(S, f^\#(\varphi), X, Y, Z)$$

where $\text{Sat}(S, \varphi, X, Y, Z)$ is the set of triples of sets that satisfy φ in S .

(b) (**Fefermann-Vaught**) : One can construct formulas ψ_1, \dots, ψ_n , $\theta_1, \dots, \theta_n$ such that :

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

Better :

$$\text{Sat}(S \oplus T, \varphi, X, Y, Z) = \bigcup_i \text{Sat}(S, \psi_i, X, Y, Z) \otimes \text{Sat}(T, \theta_i, X, Y, Z)$$

Here $f^\#(\varphi)$, ψ_1, \dots, ψ_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 Φ_k . One builds an automaton with states the subsets of Φ_k (the MS-theories of quantifier-height $\leq k$ of the graphs defined by the subterms of the term to be processed).

Consequences

1) For **checking the validity** of properties defined by closed formulas : point (c) gives the finite set of auxiliary formulas ; (a) and (b) show they form an inductive set.

2) For **computing values** :

$$\text{opt}(\varphi)(G) = \text{Min} \{ |X| / G \models \varphi(X) \}$$

$$\text{Sat}(S \oplus T, \varphi, X) = \bigcup_i \text{Sat}(S, \psi_i, X) \otimes \text{Sat}(T, \theta_i, X)$$

yields :

$$\text{opt}(\varphi)(S \oplus T) = \text{Min}_i \{ \text{opt}(\psi_i)(S) + \text{opt}(\theta_i)(T) \}$$

3) Similar calculations **for counting answers to queries** :

$$\#(\varphi)(G) = \text{Card} \{ (X, Y) / G \models \varphi(X, Y) \}.$$

4) Applications : linear delay enumeration of answers to MS expressible queries.

7. Monadic Second-Order Transductions

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

τ is multivalued by the different choices of parameters.

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

$(G, X, Y) \longmapsto$ 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 δ on words: $u \mapsto uu$

We let $u = \text{aac}$

S

| | | | | |
|---|---|---|---|---|
| • | → | • | → | • |
| a | | a | | c |

$S \oplus S$

| | | | | | | | | | |
|-------|---|-------|---|-------|-------|---|-------|---|-------|
| • | → | • | → | • | • | → | • | → | • |
| a | | a | | c | a | | a | | c |
| p_1 | | p_1 | | p_1 | p_2 | | p_2 | | p_2 |

$\delta(S)$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| • | → | • | → | • | → | • | → | • | → | • |
| a | | a | | c | | a | | a | | c |

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

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

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

The fundamental property of MS transductions :

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

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

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

The verification of ψ 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 \Leftrightarrow 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** \Leftrightarrow it is the image of **a** set of binary trees under an **MS** transduction.

2) A set of graphs is HR-equational \Leftrightarrow it is the image of **(all)** binary trees under an **MS₂ transduction**. (**MS₂** = MS with edge set quantifications).

HR-equational sets are stable under **MS₂**-transductions.

A set of graphs has bounded **tree-width** \Leftrightarrow it is the image of **a** set of binary trees under an **MS₂ transduction**.

3) A set of hypergraphs is QF-equational \Leftrightarrow 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**.

QF = the signature for relational structures consisting of \oplus (disjoint union) and quantifier-free definable transformations. It includes VR^+ .

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 ⁻¹ |
| | | | MS-trans ⁻¹ |

Signatures for graphs and hypergraphs :

HR : graphs and hypergraphs with “sources”

VR : graphs with vertex labels (“ports”)

VR⁺ : *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 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 $Minors(C)$, because $C \longrightarrow Minors(C)$ 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 $Minors(C)$.

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 $Vertex-Minors(C)$, because $C \longrightarrow Vertex-Minors(C)$ is a **C_2MS** transduction.
- E) An **MS** transduction transforms S_k into the **$k \times k$ -grid**.

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

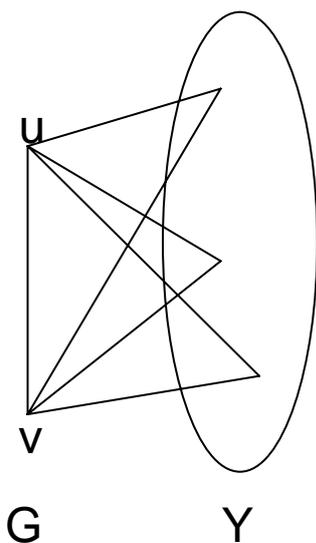
Local equivalence (\approx_{loc}) = transitive closure of local complementation (at all vertices)

Vertex-minor relation :

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

Proposition : The mapping associating with G its locally equivalent graphs is a **C₂MS** transduction.

Why is the even cardinality set predicate necessary ?



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

u is linked to v in $G * X$

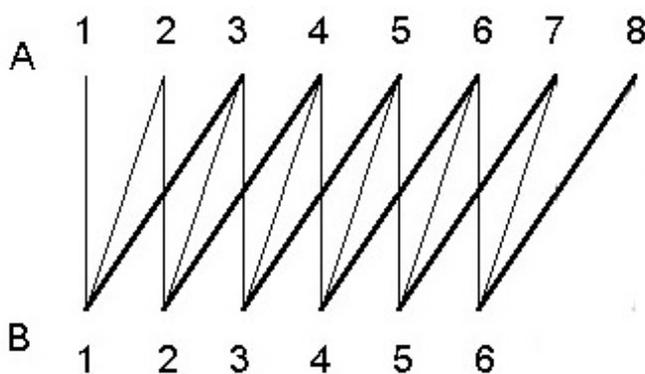
$\Leftrightarrow \text{Card}(X)$ is even

Definition of S_k :

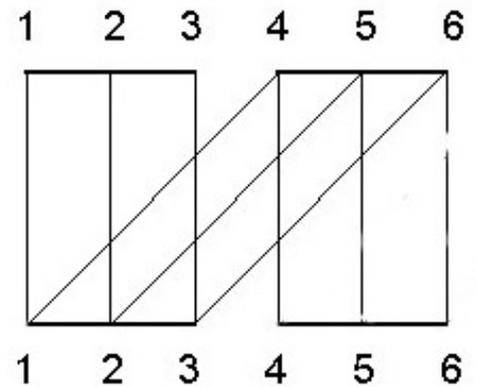
bipartite : $A = \{1, \dots, (k+1)(k-1)\}$, $B = \{1, \dots, k(k-1)\}$

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 **C₂MS** 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. Some open questions

Question 1 : Under which operations, quantifier-free definable or not, are $\text{REC}(\text{VR})$ and $\text{REC}(\text{HR})$ closed ?

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

Question 2 : Is it true that the decidability of the MS (and not the C_2MS) satisfiability problem for a set of graphs implies bounded clique-width, as conjectured by D. Seese ?