

Graph equivalences and decompositions
definable in Monadic Second-Order logic

Circle Graphs

Bruno COURCELLE

LaBRI, Bordeaux 1 University

Graph Decompositions : 4 motivations

1. Context-free Graph Grammars
2. Algorithmic applications (FPT, preliminary steps)
3. Decidability of logical theories (Monadic 2nd order logic)
4. Graph Theory , structural descriptions :

Tree-decompositions and the Graph Minor Theorem

Extensions to matroids (in view of a Matroid Minor Theorem)

Modular and "split" decompositions

Monadic Second-Order Logic

MSOL = First-order logic over subsets of the domain

Graph $G = \langle V, \text{edg}(\dots) \rangle = \langle \text{Vertices, adjacency relation} \rangle$

Logical expression of graph properties :

Non connectivity : $\exists X (\exists u. u \in X \wedge \exists v. v \notin X$
 $\wedge [\forall u, v. \text{edg}(u, v) \Rightarrow (u \in X \Leftrightarrow v \in X)])$

Typical Properties : *k-colorability (fixed k)*
to be a tree
planarity (by Kuratowsky's characterization)

but cannot express : equal cardinality of two sets ; bijections.

Why Monadic Second-Order Logic is interesting ?

Context-free Graph Grammars :

MS logic → Recognizable sets of graphs
→ Transductions (no automata)

Algorithmic applications :

MS logic → Fixed Parameter Tractable problems
for tree-width and clique-width

→ Query evaluation techniques
implementation of graphs ;
linear delay evaluation

Decidability of logical theories :

- * MS_2 decidability of graphs implies bounded tree-width,
- * MS decidability of matroids implies bounded branch-width

Graph Theory : Decompositions and Monadic 2nd order logic

a) Non-canonical graph decompositions :

Tree-decompositions of width $\leq k$, fixed k , are MS constructible : this gives Recognizability = CMS-definability for graphs of tree-width $\leq k$.

b) Canonical decompositions are easier to construct in M. 2nd o. logic:

3-Connected components

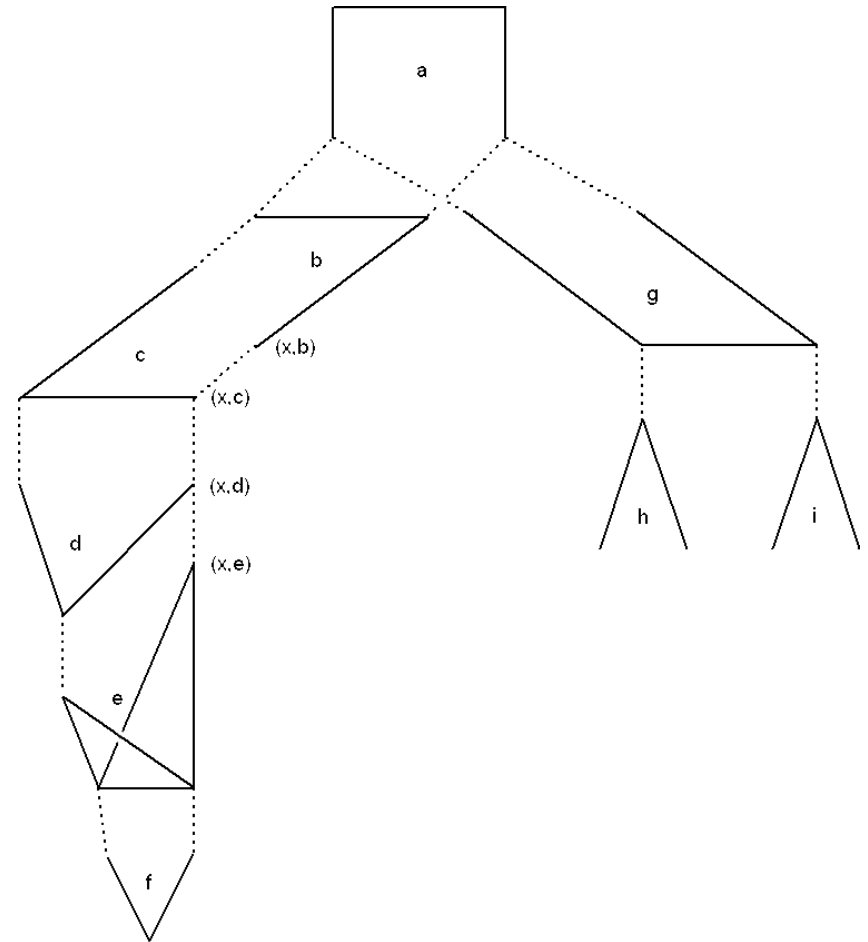
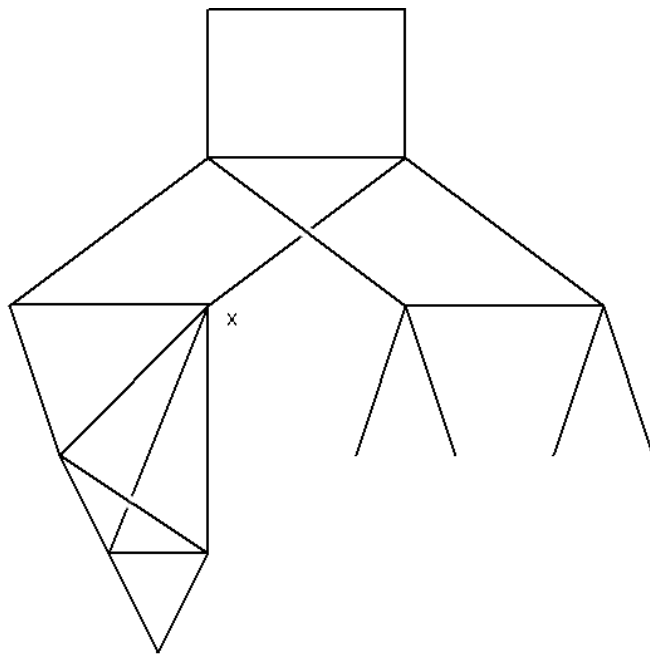
Modular decomposition

Split decomposition

General question : Which graph theoretical constructions are expressible in MS logic ?

- Decompositions
- Intersection models (**circle graphs**)
- Geometrical models (surface embeddings)
- Extensions of these questions to matroids

Tree decompositions



A graph and a tree-decomposition of width 3

Why interesting ?

1) Structure theorems for the theory of excluded minors
(Robertson and Seymour)

2) Inductive proofs and computations

Example : 3-colorability is NP-complete but testable in linear time on graphs of tree-width $\leq k$, for each fixed k .

Thm : Every property expressible in Monadic 2nd order Logic is linear on graphs of tree-width $\leq k$.

3) Can be constructed in linear time (for each fixed k)

Tool box

Graph grammars : they produce graphs with tree-decompositions of bounded tree-width (defined by the grammar).

Algorithms for tree-decompositions and other types of decompositions

Classes of graphs characterised by excluded configurations (minors, induced subgraphs, vertex-minors)

Basic graph properties expressible in Monadic 2nd order logic

Transformations of graphs, trees, and other structures formalised in Monadic 2nd order logic.

Monadic 2nd order transductions (or "interpretations")

Transformation τ of logical structures such that :

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

where T is defined by **monadic 2nd order formulas**
inside the structure: $S \oplus S \oplus \dots \oplus S$

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

in terms of "parameters" i.e. subsets X_1, \dots, X_p
of the domain of S

Proposition : The composition of two monadic 2nd order transductions
is a **monadic 2nd order** transduction.

The inverse image of a monadic 2nd order definable set under
a monadic 2nd order transduction is monadic 2nd order definable

Examples of Monadic 2nd order transductions

The mappings from a graph to :

the forest of its biconnected components

the directed acyclic graph of its strongly connected components

all its spanning trees

all its minors

(by using the incidence structure $\langle V \cup E, \text{inc}(\cdot, \cdot) \rangle$)

Modular decomposition of (finite) graphs (Gallai 1967)

Graph substitution : $G[H/u]$: H replaces u and all neighbours of u in G are linked to all vertices of H

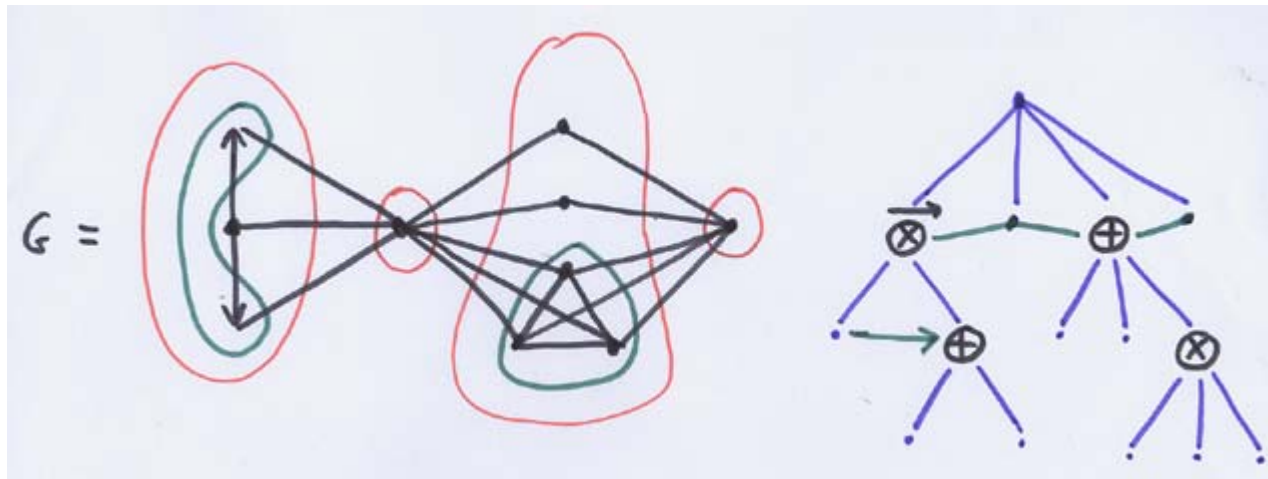
Module : M is a *module* of G if $G = K[H/u]$ with $M = \text{Vertices}(H)$.



Modular decomposition : expression of a graph in terms of substitutions

Every graph is partitioned into maximal proper strong modules.

Strong module : does not overlap any other module

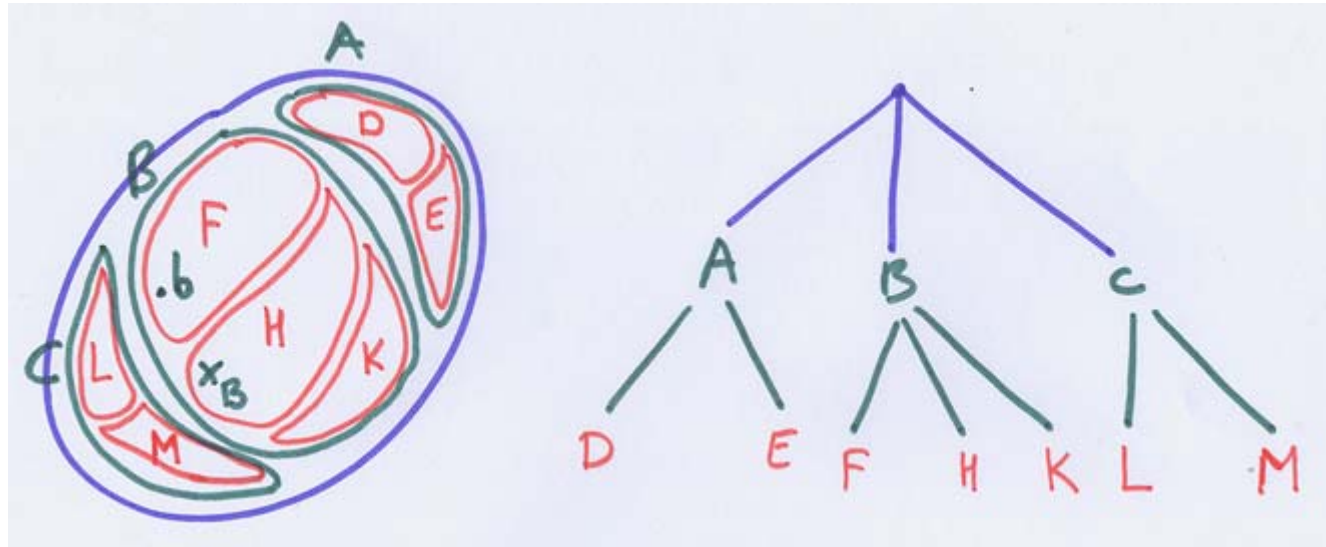


$$G = (u - v - w - x)[H / u, L / w]$$

$$H = (y \rightarrow z)[(\bullet \bullet) / z]$$

$$L = (a \bullet \quad b \bullet \quad c \bullet)[K_3 / c]$$

It can be constructed from G and an *arbitrary linear order* \leq of its vertices by fixed **Monadic 2nd** order formulas



Each module A, B, C , is represented by one of its elements ; each vertex represents at most one module.

For module M , $Min(M)$ is its \leq -minimal element ;

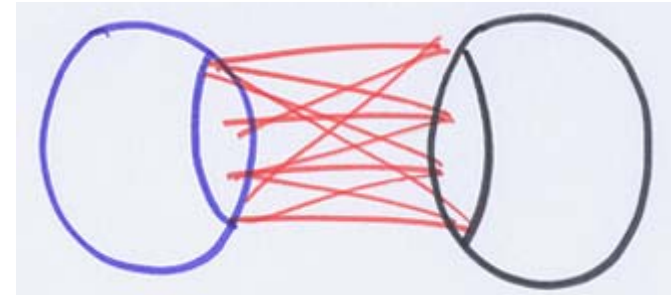
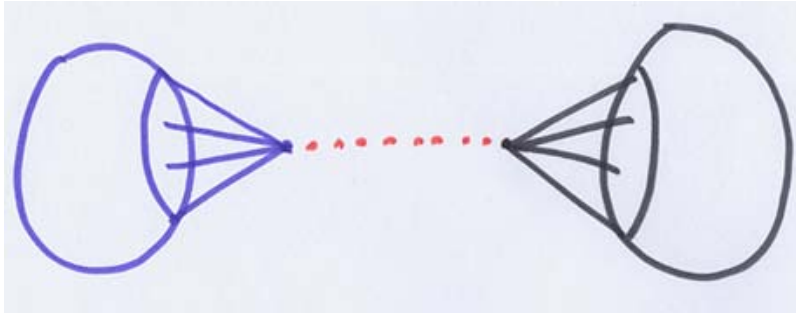
$MinSubMod(M)$ is its son submodule containing $Min(M)$;

M is represented by $x_M = Min(M - MinSubMod(M))$

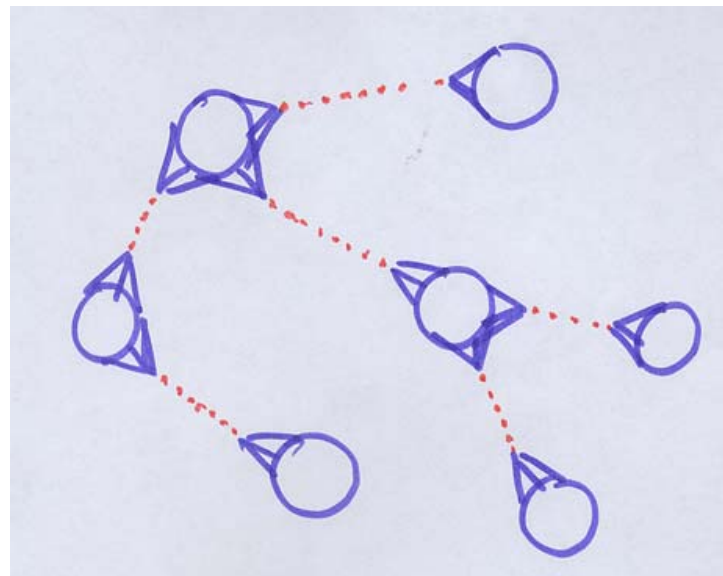
Example : $b = Min(B)$

Split decomposition (Cunnigham 1982)

Based on a different graph composition

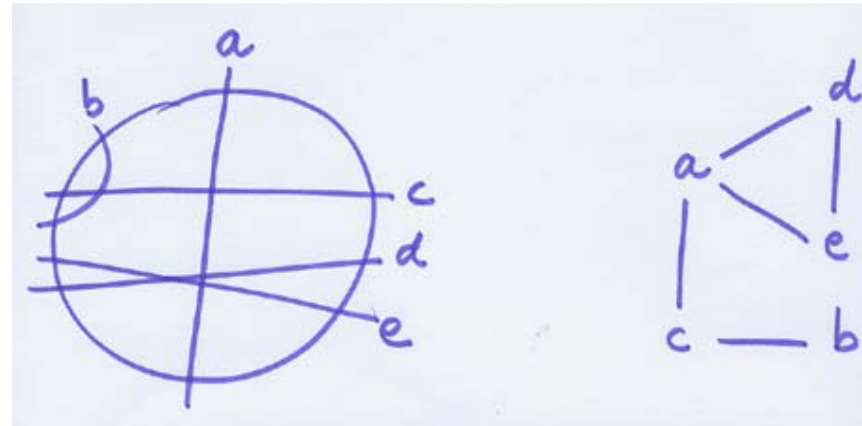


The canonical "split decomposition" is a tree of cliques, stars, *prime graphs*.

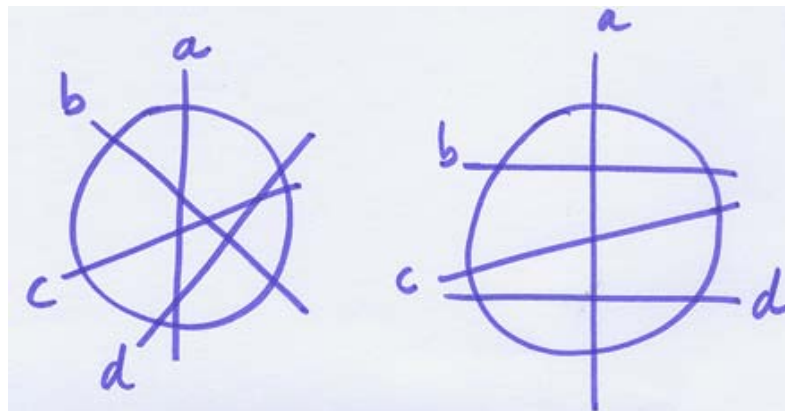


Circle graphs

Intersection graph of chords of a circle



Combinatorial description by **double occurrence words** : abcbedaedc

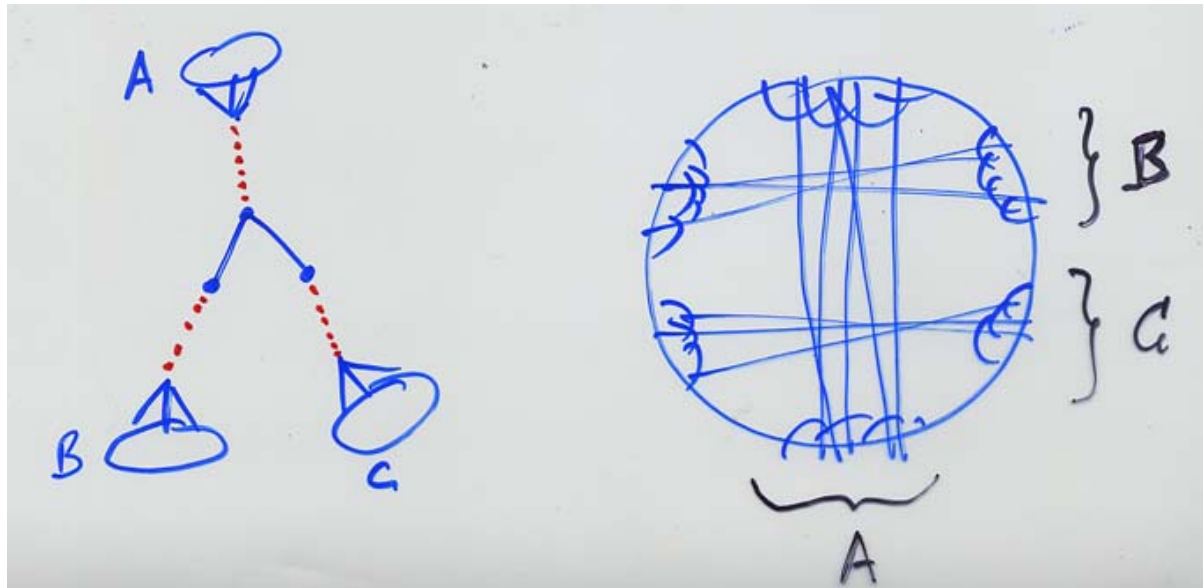


Word $abcdabcd$ defining K_4

Word $abcdadcb$ defining the star S_3

Theorem : 1) A graph is a circle graph iff the components of **its split decomposition** are circle graphs.

2) If a circle graph is **prime**, it has a unique representation as a double occurrence word (up to reversal and conjugacy) or as chords (up to homeomorphism).



Theorem : 1) Circle graphs are characterized by 3 **excluded vertex-minors** (Bouchet 1994).

2) This characterization is expressible in **Monadic 2nd order logic** with even cardinality set predicate (Courcelle, Oum 2006)



Vertex-Minors (Oum)

$G \rightarrow G \setminus x$: deletion of vertex x (and incident edges)

$G \rightarrow G * x$: **local complementation** at vertex x : one takes the edge complementation of the subgraph induced by the neighbours of x .

H is a **vertex-minor** of G if $G \rightarrow^* H$

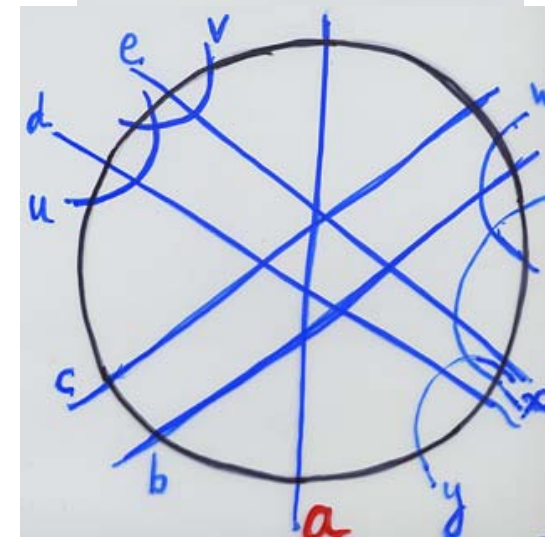
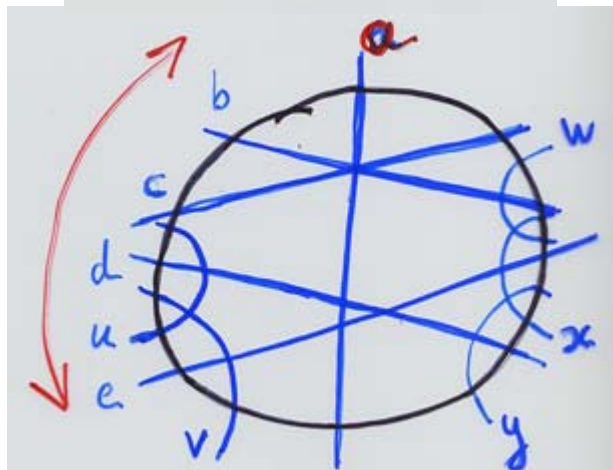
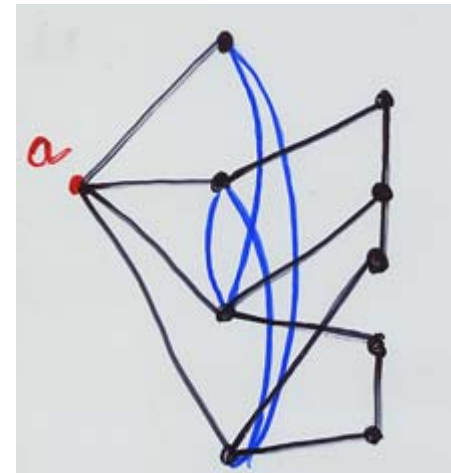
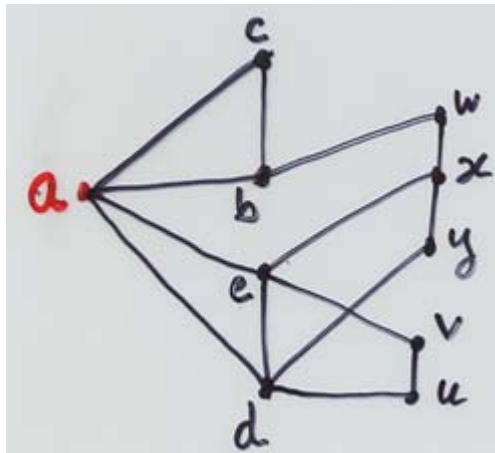
Some classes are characterized by excluded vertex-minors : distance hereditary, circle graphs, graphs of **rank-width** $\leq k$.

Question : Is vertex-minor inclusion a well-quasi order like is minor inclusion ?

Vertex-minors of circle graphs ?

$G \setminus a$: deletion of letter a in a word (of chord a in a chord) representation

$G * a$: replacing word $anaw$ by word $a\tilde{n}aw$;



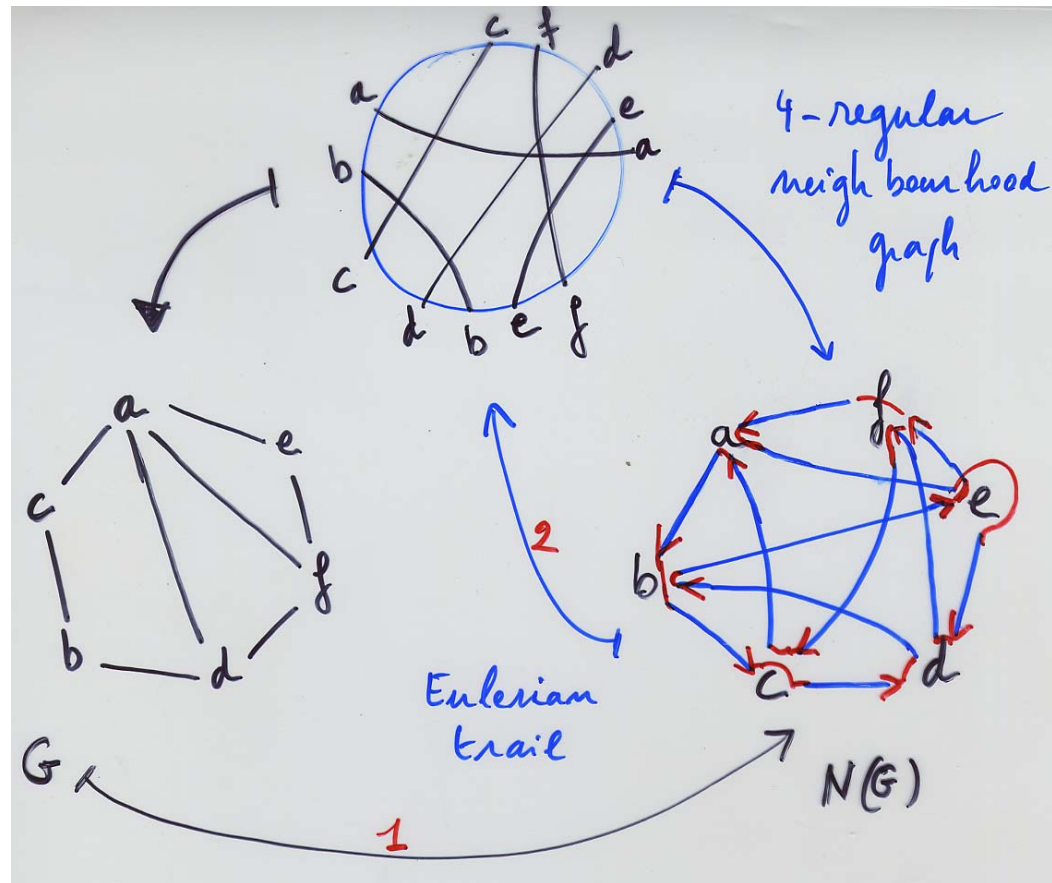
Theorem : There exists a Monadic 2^{nd} order transduction that defines, for every *linearly ordered* circle graph a representation of it by a double occurrence word.

Method : 1) We prove the case of a prime circle graph by using the unicity of the representation.

2) In the general case, we build the split decomposition of the given graph G by a Monadic 2^{nd} order transduction.

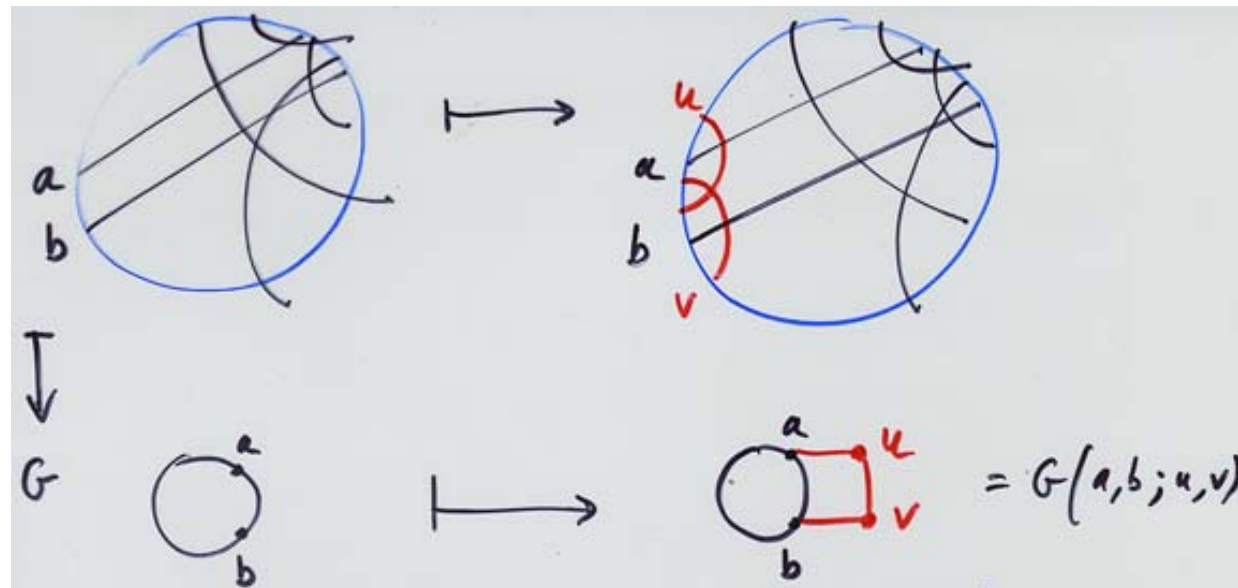
3) We combine the constructed representations of the components into one of the graph G , again by a Monadic 2^{nd} order transduction.

First step : We construct the unique representation of a prime CG



- 1) $N(G)$ the neighbourhood graph is definable in G by C_2MS formulas
- 2) The Eulerian trails of a 4-regular graph are also MS definable

Proof of 1):



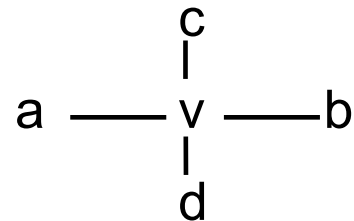
For a *prime* circle graph G , the graph $G(a,b; u,v)$ is a circle graph iff a and b are neighbours in the unique representation of G .

Hence, "neighbourhood" is expressible by a formula of Monadic 2nd order logic with even cardinality predicate by Courcelle-Oum.

Proof of 2) :

A 4-regular graph H has a 17 vertex-coloring such that vertices at distance 1 or 2 have distinct colors.

Let v be a vertex with 4 neighbours colored by colors a,b,c,d :



By attaching to v the 4-tuple (a,b,c,d) in $\{1,\dots,17\}^4$ we may stipulate that the Eulerian trail traverses v from a to b and from c to d ; by attaching to v the 4-tuple (a,c,b,d) we may stipulate that the trail traverses v from a to c and from b to d .

This can be done in Monadic 2^{nd} order logic by existential set quantifications. Hence, Eulerian trails can be specified.

The "good" one can be determined.

Second step : Constructing the canonical split decomposition.

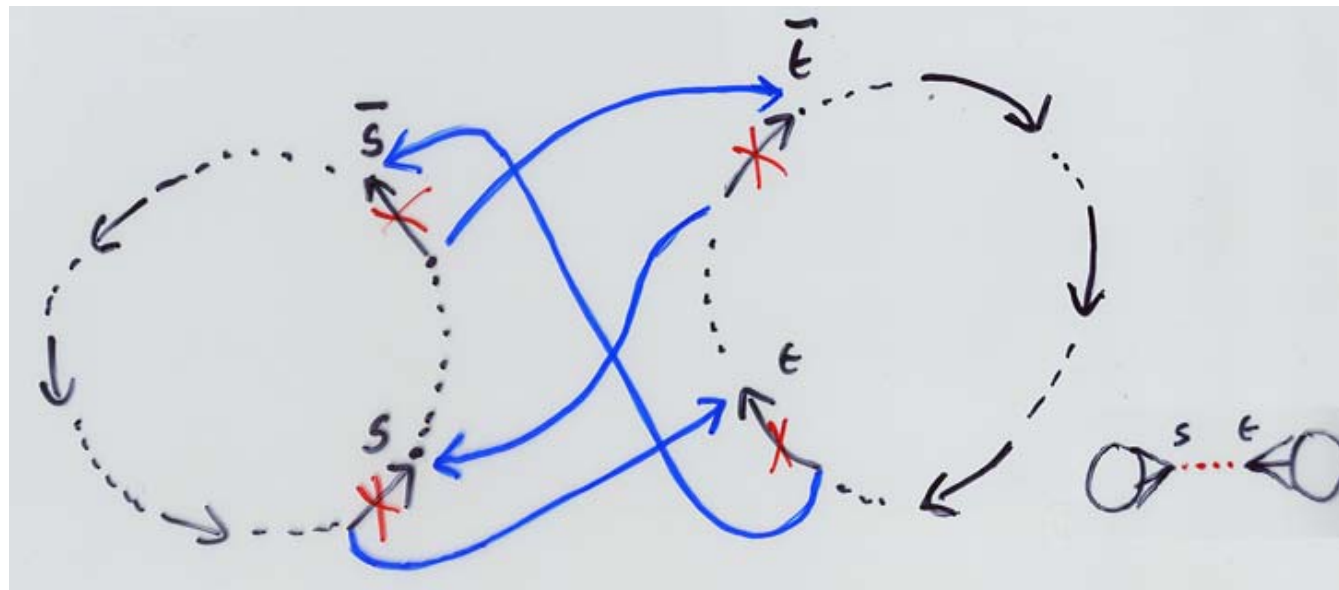
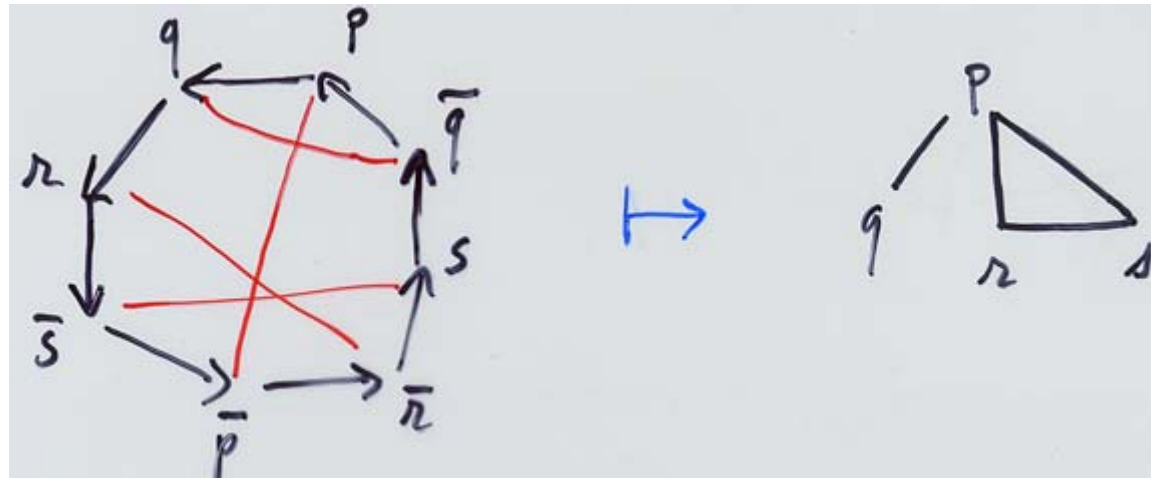
Can be done (like for the modular decomposition).

For each circle graph that is a component of the split decomposition, we construct a representation, either by using the first step or more easily (but with the help of the linear order) for *cliques* (square words : ww) and *stars* (words of the form anañ ; ñ is the reversal of word n).

Third step : It remains to combine the word representations of components into a word representation of the considered graph.

Remark : Double occurrence word = Hamiltonian circuit + edges that match positions with same letter.

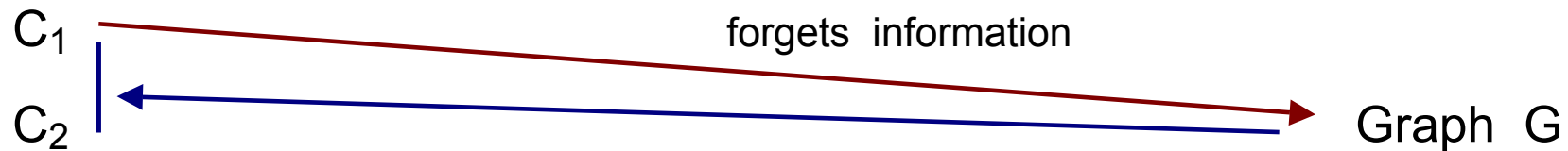
Combining representations of circle graphs



Overview of this method which applies to other cases

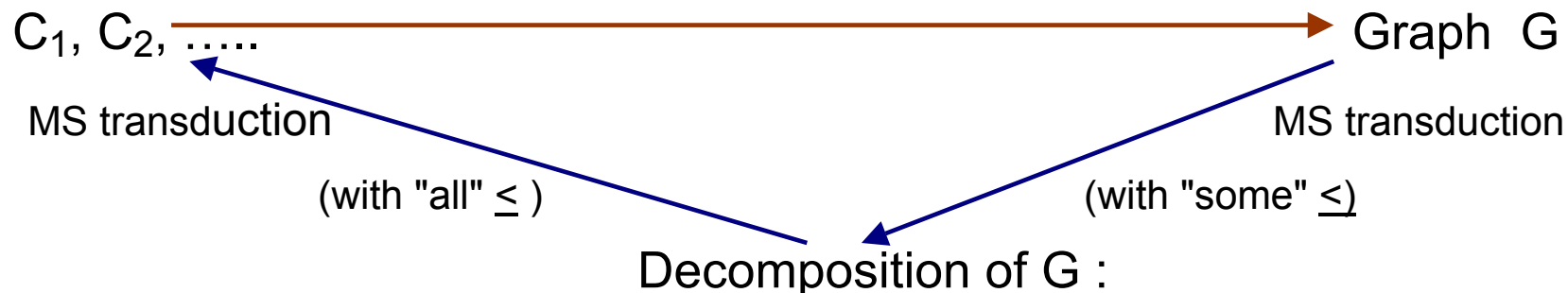
Geometrical configurations

(coded by relational structures)



.....

How can one reconstruct C_1, C_2, \dots from G ?



tree of basic blocks B with unique C_B

$B \rightarrow C_B$: Monadic 2^{nd} order transduction using the unicity