



On several proofs of the Recognizability Theorem

Bruno Courcelle

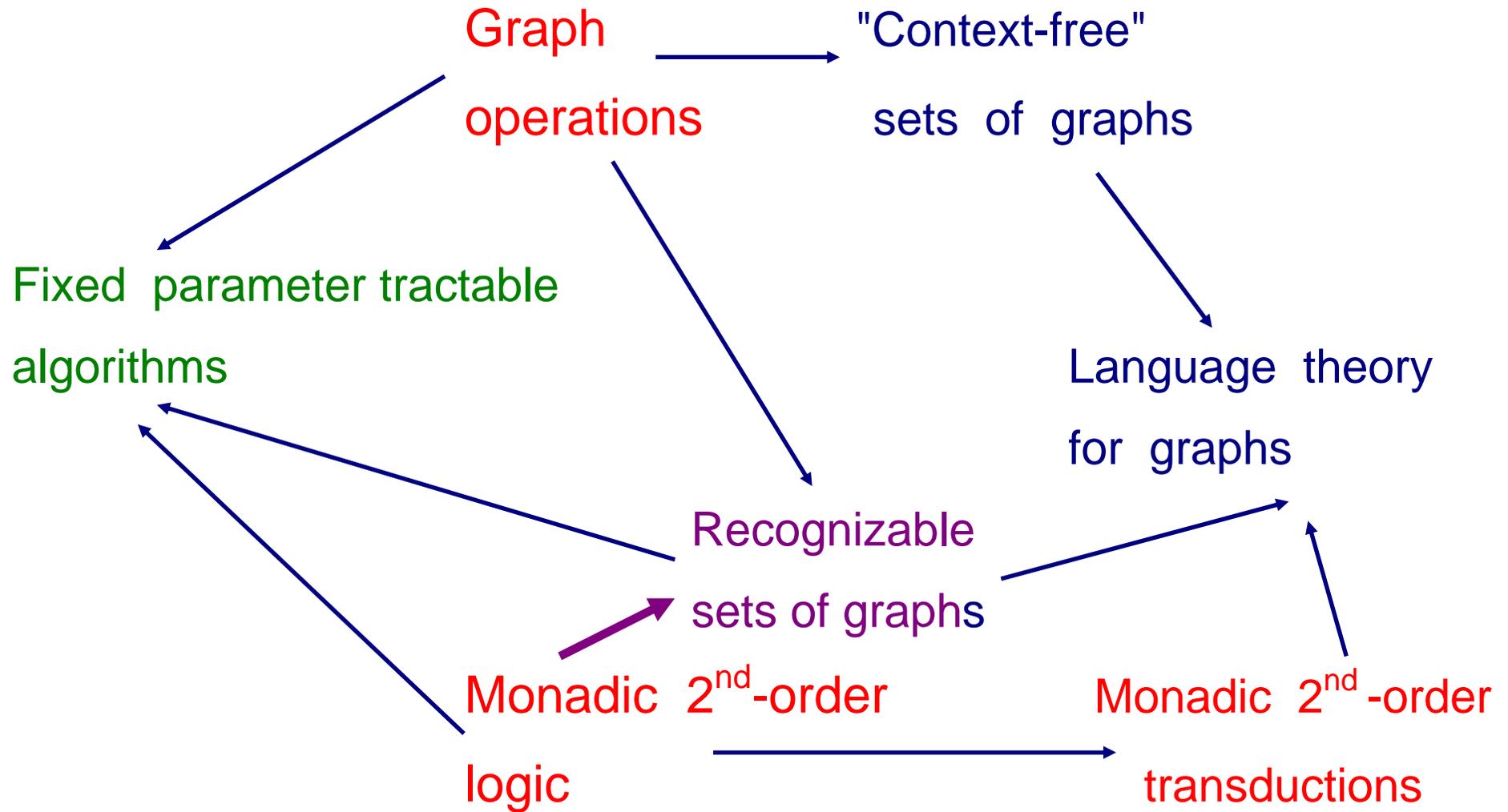
Université Bordeaux 1, LaBRI, and Institut Universitaire de France



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>

An overview chart



Key concepts of Language Theory and their extensions

| Languages | Graphs |
|--|---|
| 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 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 | Monadic Second-order transductions |

Recognizability Theorem : the inclusion \cup

Summary

1. The graph algebra **VR**.
2. Recognizability : an algebraic notion
3. Monadic second-order logic
4. The Recognizability Theorem
5. The case of terms (proof by Büchi)
6. **VR_k**- Recognizability after Büchi.
7. **VR**- Recognizability using the Feferman-Vaught paradigm
8. Extension of the Büchi-style proof after Engelfriet
9. Comparisons

1. The graph algebra **VR**

Origin: Vertex Replacement *context-free* graph grammars

Associated complexity measure: **clique-width**.

Graphs are defined in terms of **very simple graph operations**,
giving easy inductive proofs.

Graphs are simple, loop-free, directed or not.

Vertex labels : $1, \dots, k$. Each vertex has one and only one label.

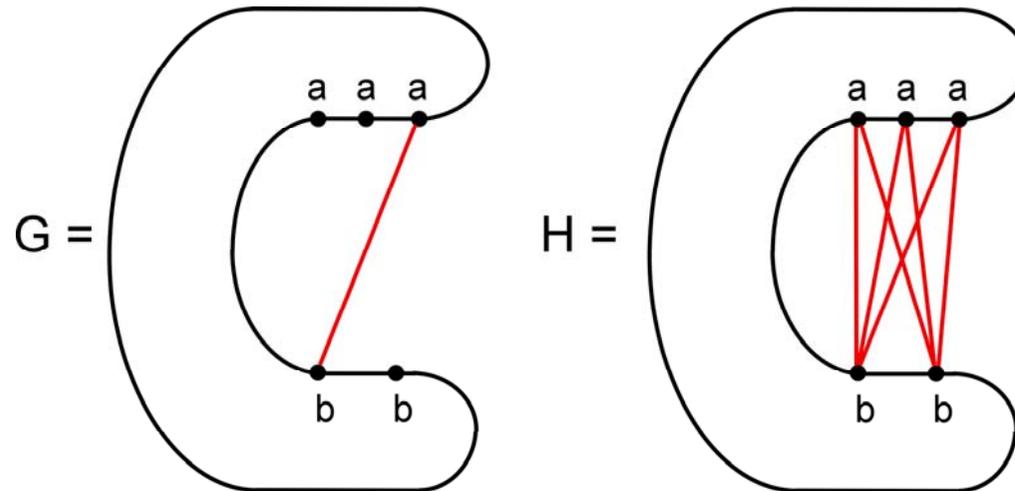
One binary operation: **disjoint union** : \oplus

Well-defined up to isomorphism : one takes disjoint copies ;

$G \oplus G$ is not equal to G

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

$Add-edg_{a,b}(G)$ is G augmented with undirected edges between every a -labelled vertex and every b -labelled vertex



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

The number of added edges depends on the argument graph.

The directed version of $Add-edg_{a,b}$ adds directed edges from every a -labelled vertex to every b -labelled vertex

Vertex relabellings :

$Relab_{a \rightarrow b}(G)$ is G with every (vertex) label a changed into b

Basic graphs : a one vertex labelled by a .

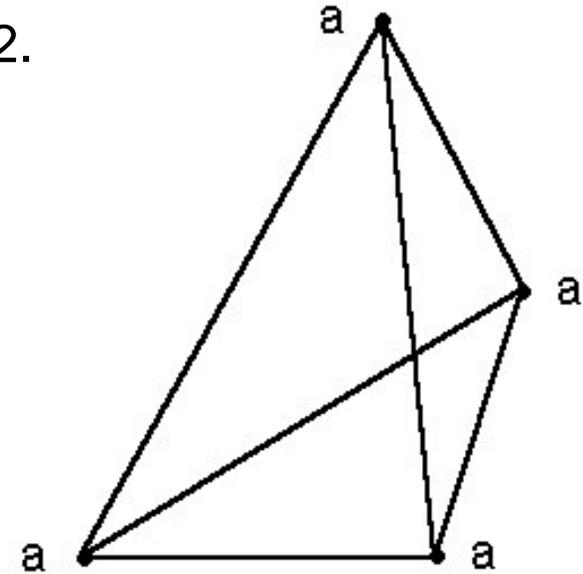
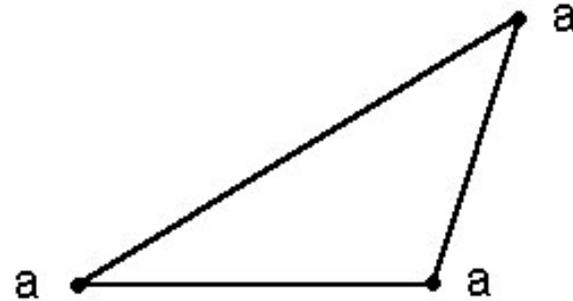
Definition: A graph G has **clique-width** $\leq k$

\Leftrightarrow it can be constructed from basic graphs with the operations

\oplus , $Add-edge_{a,b}$ and $Relab_{a \rightarrow b}$ with labels a, b in $1, \dots, k$.

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

Example : Cliques (*a*-labelled) have clique-width 2.



K_n is defined by t_n where $t_1 = \mathbf{a}$

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

Example : Cographs (*a*-labelled) are generated by \oplus and \otimes defined by:

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

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

2. Recognizable sets : an algebraic notion

$\mathbf{M} = \langle M, (f_{\mathbf{M}})_{f \in F} \rangle$: an F-algebra where F is a *finite* signature.

$L \subseteq M$ is **(M-)recognizable** if it is a union of equivalence classes for a finite congruence \approx on M.

Congruence = equivalence relation such that :

$$m \approx m' \text{ and } p \approx p' \Rightarrow f_{\mathbf{M}}(m,p) \approx f_{\mathbf{M}}(m',p').$$

Finite means that M / \approx is finite, i.e., \approx has finitely many classes.

Equivalently, $L = h^{-1}(D)$ for a homomorphism $h : \mathbf{M} \rightarrow \mathbf{A}$, where

\mathbf{A} is a *finite* F-algebra and $D \subseteq A$.

Rec(M) = the recognizable subsets of M .

This notion is relative to the algebra \mathbf{M} .

Classical examples

Algebra

$\langle A^*, \cdot, \varepsilon, a, b, \dots, d \rangle$

$\langle A^*, \varepsilon, (\lambda u \in A^*. ua)_{a \in A} \rangle$

$\mathbf{T}(F)$, terms over F , (initial F -algebra)

On terms, h is the run of a *finite deterministic bottom-up automaton*

$\langle \mathbf{N}^k, +, (0, \dots, 0), \dots (0, \dots, 1, 0, \dots, 0) \dots \rangle$

Finite unions of Cartesian products of k sets $\{ \mathbf{u} + n \cdot \mathbf{v} \mid n \in \mathbf{N} \}$ for $\mathbf{u}, \mathbf{v} \in \mathbf{N}$

Recognizable sets

Regular languages
(syntactic monoid)

Regular languages
(Myhill-Nerode)

Regular sets of terms

Finite unions of Cartesian

The algebra **VR** has an *infinite* signature

We introduce a notion of **type** (or *sort* in a many-sorted framework).

The **type** of G is : $\pi(G)$ = the set of vertex labels having an occurrence.

π has a homomorphic behaviour :

$$\pi(G \oplus H) = \pi(G) \cup \pi(H) \quad ; \quad \pi(\text{Add-edg}_{a,b}(G)) = \pi(G) \quad ;$$

$$\pi(\text{Relab}_a \rightarrow b(G)) = \pi(G)[b/a].$$

For defining the recognizability of set L we require that the congruence \approx

is *type preserving* : $G \approx H$ implies $\pi(G) = \pi(H)$,

locally finite : it has finitely many classes of each type,

L is a union of classes (possibly of different types).

Properties of recognizable sets that follow from the algebraic setting :

Closure under \cup , \cap and $-$ (difference),
under inverse homomorphisms and
under inverse unary derived operations.

Filtering Theorem: The intersection of an equational
(i.e., “context-free”) set and a recognizable one is equational
With *effective constructions*.

Properties of recognizable sets of graphs that do not follow “algebraically”

Closure under the binary operation \oplus ,

under the unary *relabelling* operations.

(false for Add-edg but true if some “harmless” restriction of the use of this operation is made.)

(*Fact*: It is *more difficult* to prove the closure under concatenation of regular languages than their closure under Boolean operations ; this is reflected by the sizes of syntactic monoids).

Properties that do not hold as we could wish or expect.

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

Rec and **Equat** are **incomparable**.

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

There are **uncountably** many recognizable sets, hence *no characterization by finite automata or logical formulas*.

Proposition : If F is *finite* and generates \mathbf{M} , i.e., if every element of \mathbf{M} is the value $\text{val}_{\mathbf{M}}(t)$ of some term t in $T(F)$, then :

a subset L of \mathbf{M} is **\mathbf{M} -recognizable** if and only if

the set of terms $\text{val}_{\mathbf{M}}^{-1}(L)$ is **$T(F)$ -recognizable**, if and only if

$\text{val}_{\mathbf{M}}^{-1}(L)$ is defined by a finite (deterministic) automaton.

Two possible proofs that a set L is **\mathbf{M} -recognizable** :

Construction of a finite congruence on \mathbf{M} that saturates L ,

Construction of an automaton that defines $\text{val}_{\mathbf{M}}^{-1}(L)$.

3. Monadic Second-Order (MS) Logic

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

3-colorability (NP-complete property)

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

Non connectivity (whence connectivity, more generally transitive closure) :

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

Planarity : no K_5 or $K_{3,3}$ minor.

Having a K_5 minor :

$$\begin{aligned} \exists X_1, \dots, X_5 (& \text{“the } X_i\text{'s are pairwise disjoint”} \\ & \wedge \text{“each } X_i \text{ induces a connected graph”} \\ & \wedge \text{“for every } i < j, \text{ there is an edge between some } u \text{ in } X_i \text{ and some } v \text{ in } X_j\text{”}) \end{aligned}$$

4. The recognizability theorem

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

$$L = \text{Models}(\varphi) := \{ S \text{ finite} \mid S \models \varphi \} \text{ for a fixed MS sentence } \varphi$$

The Recognizability Theorem :

Every **MS**- definable set of graphs is **VR**-recognizable

Every **MS**- definable set of relational structures is recognizable with respect to an **algebra** that generalizes **VR** (using \oplus and unary quantifier-free definable operations generalizing **edge addition**, **vertex relabelling**, edge complement).

Consequences

Fixed Parameter Tractable algorithms for checking MS graph properties, for clique-width (or tree-width) as parameter.

Filtering Theorem : if L is **VR**-equational (“context-free”), the set $L \cap \text{Models}(\varphi)$ is (effectively) **VR**-equational for each MS sentence φ .

Example: for φ expressing planarity, a direct proof would be unwritable.

Decidability of monadic theories : given k and φ , one can decide if φ is true in *all* graphs of tree-width or clique-width at most k .

5. Büchi-style proof for the case of terms

Terms in $T(F)$ where F is a finite signature

Logical representation $S(t)$ of a term t :

nodes of the syntactic tree, *son* binary relation,

unary predicates lab_f for “my label is f ” and br_i for “I am the i -th brother”.

Example : $t = f_1(g_2(a_3, b_4), b_5)$ (integers 1,2,...,5 designate nodes = occurrences)

$S(t)$ has domain $\{1,2,3,4,5\}$,

son relation $\{(1,2), (1,5), (2,3), (2,4)\}$,

unary predicates : $lab_f = \{1\}$, $lab_g = \{2\}$, $lab_a = \{3\}$, $lab_b = \{4,5\}$,

$br_1 = \{2,3\}$, $br_2 = \{4,5\}$.

Construction : for each sentence φ an automaton $A(\varphi)$ that defines

$\text{Models}(\varphi) \cap T(F)$ by induction on the structure of sentence φ

- Sentences are without first-order variables and \forall .
- For $\exists X. \varphi(X)$, we need $A(\varphi(X))$; more generally $A(\varphi(X_1, \dots, X_n))$:
 f in F is replaced by $(f, (w_1, \dots, w_n))$ in $F \times \{0, 1\}^n$ of same arity,

a term $t*v$ in $T(F \times \{0, 1\}^n)$ encodes a term t in $T(F)$ and an

assignment $v : \{X_1, \dots, X_n\} \rightarrow P(\text{Nodes}(t))$

(if u is an occurrence of $(f, (w_1, \dots, w_n))$ then $w_i = 1$ iff $u \in X_i$)

- One constructs $A(\varphi(X_1, \dots, X_n))$ that defines

$$\{ t*v \in T(F \times \{0, 1\}^n) \mid (t, v) \models \varphi \}$$

Then :

$$L(A(\exists X_{n+1} \cdot \varphi(X_1, \dots, X_{n+1}))) = \text{pr}(L(A(\varphi(X_1, \dots, X_{n+1}))))$$

where **pr** is the *projection* that eliminates the last Boolean.

One obtains a *nondeterministic* automaton.

For \wedge and \vee : product of two automata.

For negation : exchange accepting/non-accepting states
for a *deterministic* automaton.

The case of atomic formulas is easy.

The number of states is an **h-iterated exponential**,
where **h** = maximum nesting of negations. This is not avoidable.

6. Büchi-style proof for VR_k -recognizability

We fix k , a bound on clique-width of the considered graphs

F_k consists of binary operation \oplus and the $O(k^2)$ unary operations $(Add-edg_{a,b}, Relab_{a \rightarrow b})$, the constant $\mathbf{1}$ defining the vertex labelled 1.

(1 can be changed into another label).

A term t in $T(F_k \cup \{\mathbf{1}\})$ defines a graph $G(t)$ with vertex set = the set of occurrences of $\mathbf{1}$. For representing assignments

$$v: \{X_1, \dots, X_n\} \rightarrow P(\text{Vertices}(G(t)))$$

we replace $\mathbf{1}$ by constants $(\mathbf{1}, (w_1, \dots, w_n))$, $w_i \in \{0, 1\}$ as for terms

A term $t*v$ in $T(F_k \cup (\{\mathbf{1}\} \times \{0, 1\}^n))$ defines the graph $G(t)$ and

some assignment $v: \{X_1, \dots, X_n\} \rightarrow P(\text{Vertices}(G(t)))$

As in the case of terms, we construct for all relevant k, n and φ a finite deterministic automaton $A(\varphi(X_1, \dots, X_n))$ that defines

$$\{ t*v \in T(F_k \cup (\{1\} \times \{0,1\}^n)) \ / \ (G(t), v) \mid = \varphi \}$$

Inductive steps :

$\varphi \wedge \psi$, $\varphi \vee \psi$: product of two automata

$\neg \varphi$: exchange of accepting / non-accepting states of a deterministic automaton

$\exists X_{n+1} \varphi$: “projection” **pr** : makes automata non-deterministic

Basic cases : atomic formulas.

Example : The automaton $A(\text{edg}(X_1, X_2))$ with k^2+k+3 states

Graph labels are in $[k] = \{1, \dots, k\}$.

States : 0, Ok, $a(1)$, $a(2)$, ab , Error, for a, b in $[k]$, $a \neq b$

Meanings of states

(at occurrence u in t ; its subterm t/u defines a graph $G(t/u) \subseteq G(t)$).

0 : $X_1 = \emptyset$, $X_2 = \emptyset$

Ok : $X_1 = \{v\}$, $X_2 = \{w\}$, $\text{edg}(v, w)$ in $G(t/u)$

$a(1)$: $X_1 = \{v\}$, $X_2 = \emptyset$, v has label a in $G(t/u)$

$a(2)$: $X_1 = \emptyset$, $X_2 = \{w\}$, w has label a in $G(t/u)$

ab : $X_1 = \{v\}$, $X_2 = \{w\}$, v has label a , w has label b (hence $v \neq w$)
and $\neg \text{edg}(v, w)$ in $G(t/u)$,

Error : all other cases.

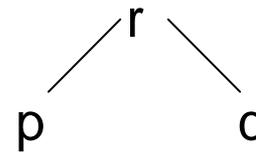
Accepting state : Ok

Transition rules

For the constant **1** :

(1,00) \rightarrow 0 ; **(1,10)** \rightarrow 1(1) ; **(1,01)** \rightarrow 1(2) , **(1,11)** \rightarrow Error

For the binary operation \oplus :



If $p = 0$ then $r := q$

If $q = 0$ then $r := p$

If $p = a(1)$, $q = b(2)$ and $a \neq b$ then $r := ab$

If $p = b(2)$, $q = a(1)$ and $a \neq b$ then $r := ab$

Otherwise $r :=$ Error

For unary operations *Add-edge*_{a,b}

r
|
p

If $p = ab$ then $r := Ok$ else $r := p$

For unary operations *Relab*_{a→b}

If $p = a(i)$ where $i = 1$ or 2 then $r := b(i)$

If $p = ac$ where $c \neq a$ and $c \neq b$ then $r := bc$

If $p = ca$ where $c \neq a$ and $c \neq b$ then $r := cb$

If $p = Error$ or 0 or Ok or $c(i)$ or cd or dc where $c \neq a$
then $r := p$

End of example ; end of proof sketch.

We have **only** proved that for every k and MS sentence φ

1) $\{ t \in T(F_k \cup \{1\}) \mid G(t) \models \varphi \}$ is $T(F_k \cup \{1\})$ -recognizable

2) The set of models of φ of clique-width $\leq k$, is VR_k -recognizable

This is **NOT** the **Recognizability Theorem** which needs a type-preserving, locally finite congruence working on ALL graphs (of unbounded clique-width).

This weak form is sufficient for **FPT algorithms** and for the **Filtering Theorem** because :

- graphs of **clique-width $\leq k$** are generated by $F_k \cup \{1\}$
- **VR-equational** implies VR_k -equational, for $k =$ maximum label

occurring in the defining system.

Although the signature VR is the union of the signatures VR_k , we may have a set L that is **not** VR -recognizable, whereas :

$L \cap \{ G / \text{cwd}(G) \leq k \}$ **is** VR_k -recognizable for each k .

Example : The set of duplicated square grids $G_{m \times m} \oplus G_{m \times m}$ is **not** VR -recognizable, but each of its restrictions to graphs of clique-width $\leq k$ is finite, hence MS-definable, whence VR_k -recognizable (by the proved special case of the Recognizability Theorem.)

VR_k-recognizability proved with Backwards Translation:

The mapping : t in $\mathbf{T}(F_k \cup \{\mathbf{1}\}) \mid \longrightarrow G(t) = \text{val}_{\text{VR}}(t)$ is an MS transduction.

The set L of terms t in $\mathbf{T}(F_k \cup \{\mathbf{1}\})$ such that $G(t) \models \varphi$ is defined by an MS formula $\varphi^\#$ obtained by *Backwards Translation*.

By the **Recognizability Theorem** for terms, L is recognizable in $\mathbf{T}(F_k \cup \{\mathbf{1}\})$, hence definable by a finite automaton.

Short proof, but $\varphi^\#$ has **larger quantifier-height than** φ , hence bad in view of a concrete implementation.

The Feferman-Vaught paradigm yields VR-recognizability

The *result* of the query defined by φ is :

$$\text{Sat}(G, \varphi, X_1, \dots, X_n) = \{ (A_1, \dots, A_n) \mid G \models \varphi(A_1, \dots, A_n) \}$$

Lemma 1 : If f is a quantifier-free mapping on graphs (edge-addition, vertex relabeling, edge complement) , every φ has a *Backwards*

Translation $f^\#(\varphi)$ relative to f such that for all G :

$$\text{Sat}(f(G), \varphi, X_1, \dots, X_n) = \text{Sat}(G, f^\#(\varphi), X_1, \dots, X_n)$$

$f^\#(\varphi)$ has no larger quantifier-height than φ

Proof : Routine.

Splitting Theorem : One can construct formulas ψ_i, θ_i , $i = 1, \dots, p$, of no larger quantifier-height than φ such that for all disjoint G and H : $\text{Sat}(G \oplus H, \varphi, X_1, \dots, X_n)$ is the disjoint union of the sets

$$\text{Sat}(G, \psi_i, X_1, \dots, X_n) \diamond \text{Sat}(H, \theta_i, X_1, \dots, X_n), \quad i = 1, \dots, p,$$

where \diamond combines “partial answers” as follows :

$$\underline{A} \diamond \underline{B} = \{ (A_1 \cup B_1, \dots, A_n \cup B_n) \mid (A_1, \dots, A_n) \in \underline{A}, (B_1, \dots, B_n) \in \underline{B} \}$$

Proof : Induction on the structure of φ and set manipulations.

Lemma 2 : For each n and h there are **finitely many** formulas

$\varphi(X_1, \dots, X_n)$ of quantifier-height $\leq h$, up to a **decidable and sound**

equivalence. *Proof sketch* : $\varphi \sim \psi$ iff ψ is obtained from φ by Boolean laws

and renamings of bound variables.

Proof of the Recognizability Theorem (general form) :

For each h , the equivalence relation such that

$$G \approx H \iff \text{Sat}(G, \psi) = \text{Sat}(H, \psi) \quad (= \emptyset \text{ or } (), \text{ the empty sequence})$$

for every sentence ψ of quantifier-height $\leq h$

is a type-preserving, locally finite congruence on **VR** that saturates

$\text{Models}(\varphi)$ for each every sentence φ of quantifier-height $\leq h$

(The same proof works for the algebra of relational structures).

For each k and MS sentence φ one obtains an automata that recognizes the set of terms in $\mathbf{T}(F_k \cup \{1\})$ that define graphs satisfying φ .

Its states are *theories* : finite sets of **reduced** sentences of quantifier-height $\leq h$ that are true in some structure.

(**Reduced** in some canonical form, using Boolean laws and renamings of bound variables, cf. Lemma 2, slide 31)

No good bound on automata sizes and no possible implementation can be obtained from this proof.

Extensions of the proof :

1) For *counting* valid assignments, i.e., for computing, for a fixed MS formula φ and a given G , the cardinality of $\text{Sat}(G, \varphi, X_1, \dots, X_n)$, the Splitting Theorem gives (*because of disjoint unions*) the recursion :

$$\begin{aligned} |\text{Sat}(G \oplus H, \varphi, X_1, \dots, X_n)| &= \\ \sum_{i=1, \dots, p} |\text{Sat}(G, \psi_i, X_1, \dots, X_n)| \cdot |\text{Sat}(H, \theta_i, X_1, \dots, X_n)| \end{aligned}$$

2) Similar fact for *optimizing functions*, defined by :

$$\text{MaxSat}(G, \varphi, X) = \text{Max} \{ |A| \mid G \models \varphi(A) \}$$

Remark : In these two cases, one could also use the deterministic automata constructed during the Büchi-style proof, with better practical perspectives.

8. A Büchi-style proof of *The Theorem*,

based on J. Engelfriet : *A regular characterization of MS definable graph languages*, Theor. Comput. Sci. Vol. 88 (1991) 139-150

For set of labels A : $G(A) = \{ \text{graphs with vertices labelled in } A \}$

For graph G and $v: \{ X_1, \dots, X_n \} \longrightarrow P(\text{Vertices}(G))$,

the graph $G*v$ in $G(\{ 0,1 \}^n)$ encodes G and v as in the “Büchi-style” proof.

For $\varphi(X_1, \dots, X_n)$ MS formula, we define :

$L(\varphi(X_1, \dots, X_n)) = \text{the set of } G*v \text{ such that } (G, v) \models \varphi$

Proposition : These sets are defined by **regular expressions** constructed from certain **local sets** by Boolean operations and **relabellings** (analogous to the “projections” of the “Büchi-style” proof).

Local sets are associated with atomic formulas

$L(X_i \subseteq X_j)$ = graphs without label (w_1, \dots, w_n) such that $w_i=1, w_j=0$

$L(\text{edg}(X_i, X_j))$ = graphs with at least one edge between a vertex labelled in B_i and a vertex labelled in B_j , where

B_i = set of labels (w_1, \dots, w_n) such that $w_i = 1$.

Relabelling based on $h : A \longrightarrow B$, replaces every label a by $h(a)$.

Useful to define $L(\exists X_n. \varphi(X_1, \dots, X_n)) = h(L(\varphi(X_1, \dots, X_n)))$ where h deletes the last Boolean in each label.

Boolean (set) operations give the cases of \wedge, \vee, \neg .

Proof of the Recognizability Theorem :

1) Each local set is recognizable : not hard to construct congruences witnessing this fact.

2) $\text{Rec}(\mathbf{VR})$ is closed under Boolean operations (general fact),

3) and under relabellings $h : G(A) \longrightarrow G(B)$:

if $L \subseteq G(A)$ is recognizable for congruence \sim ,

then $h(L) \subseteq G(B)$ is recognizable for congruence \approx such that :

$$G \approx H \iff k(G) = k(H) \text{ where } k(G) = \{ [G']_{\sim} \mid h(G') = G \}$$

The number of classes may increase exponentially from \sim to \approx .

9. Comparing the proofs

E(ngelfriet's) extends B(üchi's); F-V = Feferman-Vaught paradigm

| | B | E | F-V |
|--------------------------|----------------------|-----------|--------|
| Rec Thm for VR | NO | YES | YES |
| Rec Thm for STR | NO | Technical | YES |
| Implementability | YES "Small" cases | as for B | NO |
| Extension to counting | YES | NO | YES |
| Other algo. applications | YES | NO | ?? |
| Teachability | Easy | Technical | Doable |

About implementation: The automaton constructed from φ and k may be too large for being practically compiled.

Problems with size of memory for intermediate automata, even if the *unique minimal deterministic* automaton has manageable number of states.

Hopes come from precomputed automata for subformulas.

Soguet *et al.*, using MONA, have constructed automata for the following cases :

| | <i>Clique-width 2</i> | <i>Clique-width 3</i> |
|-----------------------|-----------------------|-----------------------|
| MaxDegree ≤ 3 | 91 states | Space-Out |
| Degree $\leq 4(x)$ | 48 states | 233 states |
| Path(x,y)in(X) | 26 states | Space-Out |
| Connectedness | 11 states | Space-Out |
| IsConnComp(X) | 48 states | Space-Out |
| Has ≤ 4 -VertCov | 111 states | 1037 states |
| HasClique ≥ 4 | 21 states | 153 states |
| 2-colorable | 11 states | 57 states |

Open questions :

To what extent can one overcome the difficulties that come from **state explosion** of the constructed automata ?

Another difficulty : the *parsing problem* : construction of an appropriate term witnessing clique-width $\leq k$ (also arise for graphs of bounded tree-width).

Bonus : Monadic Second-order logic with edge set quantifications and the graph algebra **HR**

Edge set quantifications increase the expressive power of MS logic

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

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

Monadic second-order (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 MS_2 formula, but **not by an MS** formula.

The Recognizability Theorem :

Every **MS**- definable set of finite graphs is **VR**-recognizable

Every **MS₂**- definable set of finite graphs is **HR**-recognizable

where **HR**- is another graph algebra defined below

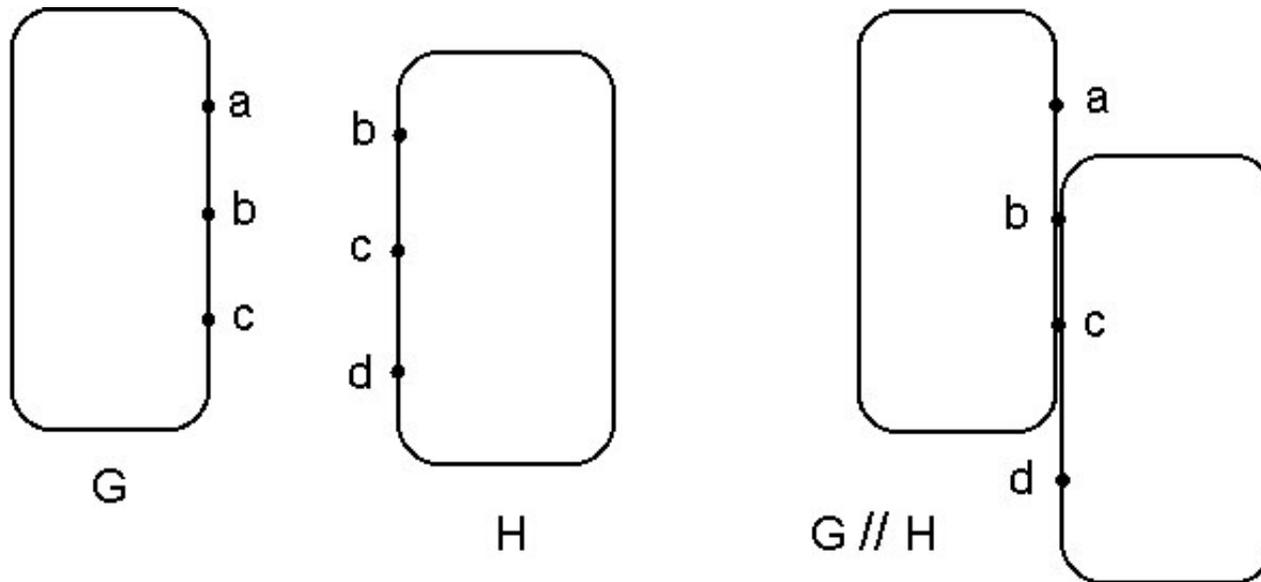
HR operations : Origin : **H**yperedge **R**eplacement hypergraph grammars ; associated complexity measure : **tree-width**

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.

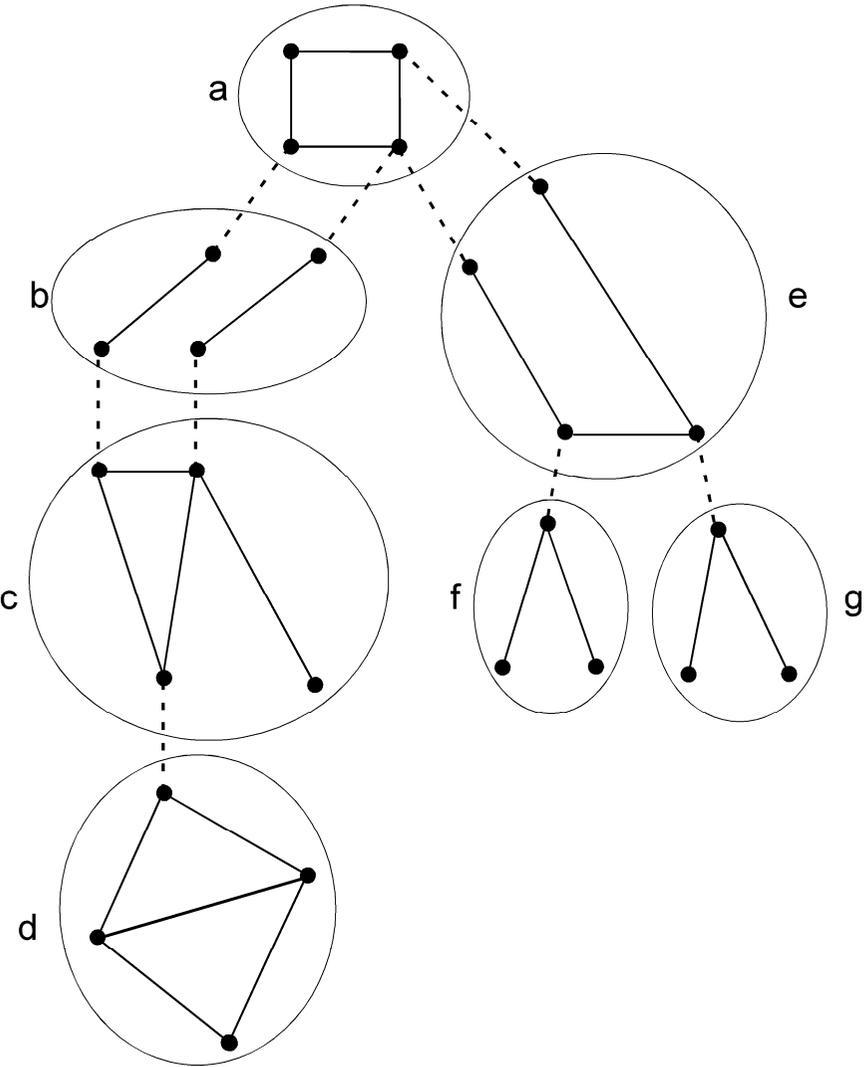
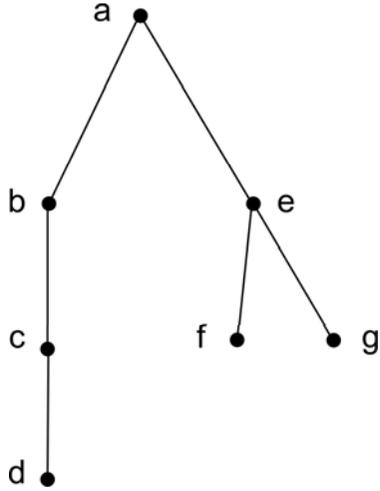
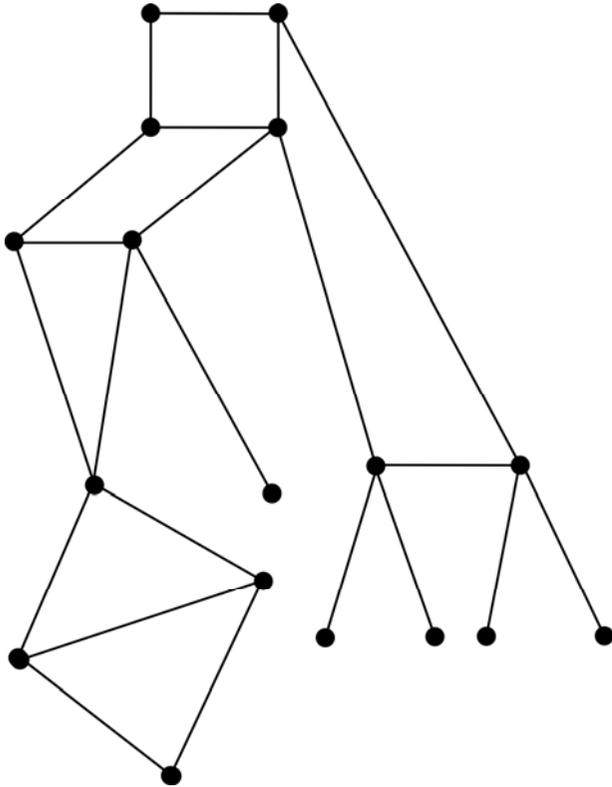
For dealing with hypergraphs one takes more nullary symbols for denoting hyperedges.

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

The type function has a homomorphic behaviour :

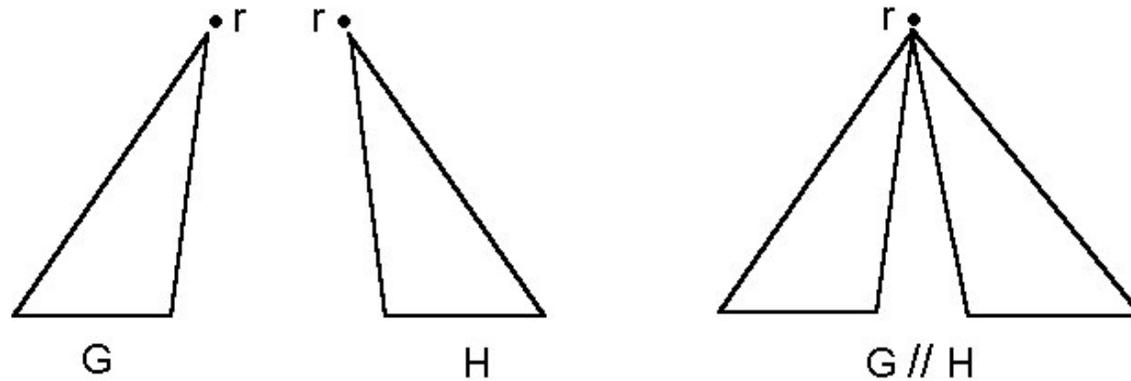
$$\tau(G//H) = \tau(G) \cup \tau(H) ; \tau(Forget_a(G)) = \tau(G) - \{a\} ; \tau(Rena_{a \leftrightarrow b}(G)) = \tau(G)[a/b, b/a].$$

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 $Forget_a$.

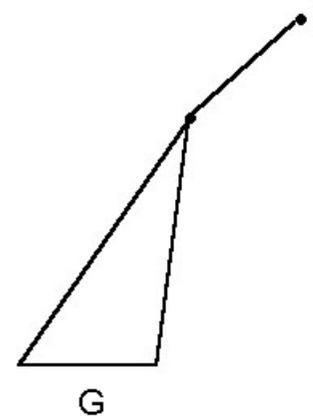
Example : Trees are of tree-width 1, constructed with two source labels, r (root) and n (new root): Fusion of two trees at their roots :



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

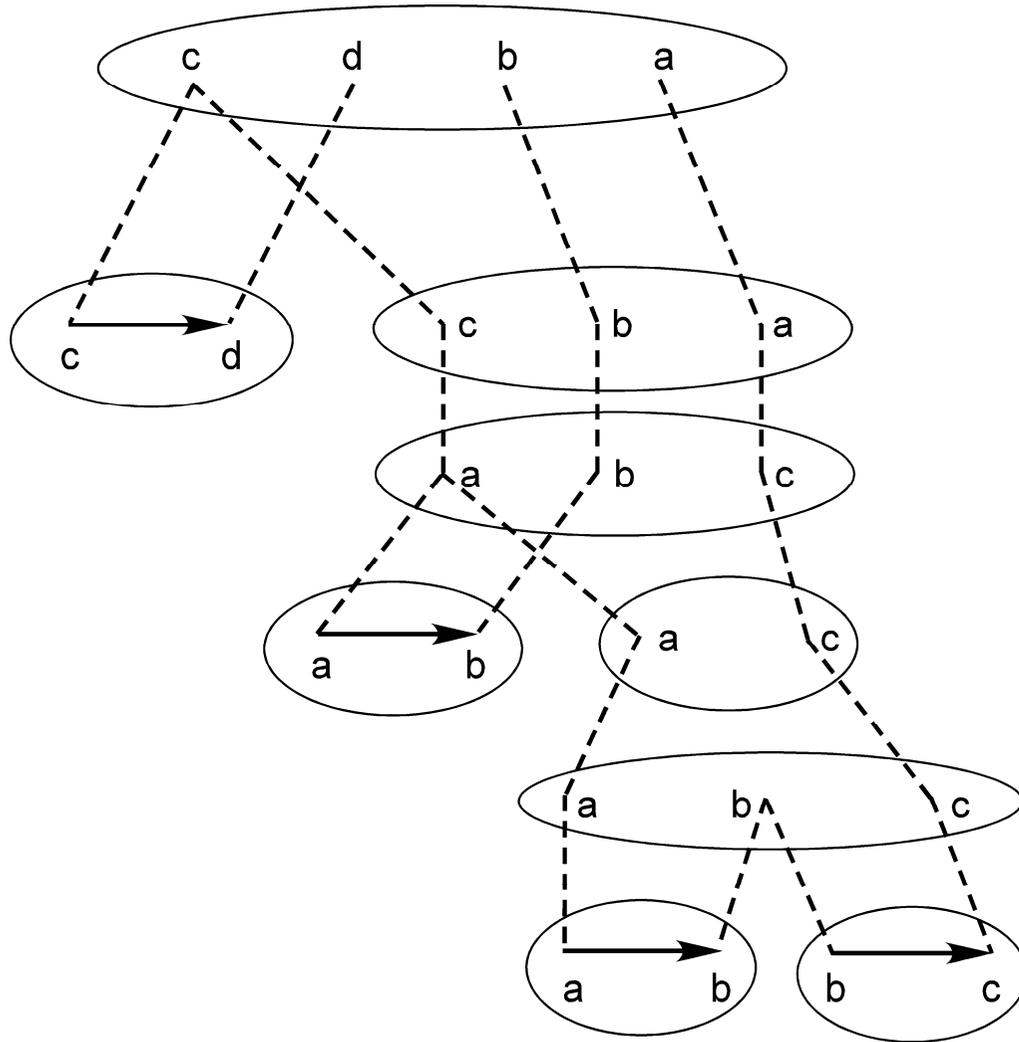
$$e = r \bullet \text{---} \bullet n$$

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



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.