

# Automata based graph algorithms for logically defined problems

## *Part 2*

Monadic Second-Order logic : definitions, examples.

The main construction : from MSO formulas to automata  
(accepting clique-width terms).

Existential quantifications and nondeterministic automata.

Tested examples : colorability problems.

# Monadic Second-Order logic

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

MSO (expressible) properties : transitive closure, properties of paths,  
connectivity, planarity (via Kuratowski, uses connectivity), p-colorability.

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

*G is 3-colorable :*

$$\begin{aligned} \exists X, Y ( X \cap Y = \emptyset \wedge \\ \forall u, v \{ \text{edg}(u, v) \Rightarrow \\ [(u \in X \Rightarrow v \notin X) \wedge (u \in Y \Rightarrow v \notin Y) \wedge \\ (u \notin X \cup Y \Rightarrow v \in X \cup Y) ] \\ \} ) \end{aligned}$$

$G$  (undirected) is *not* connected :

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

*Transitive and reflexive closure* : **TC**( $R, x, y$ ) :

$$\forall X \{ \text{“}X \text{ is } R\text{-closed”} \wedge x \in X \Rightarrow y \in X \}$$

where “ $X$  is  $R$ -closed” is defined by :

$$\forall u,v ( u \in X \wedge R(u,v) \Rightarrow v \in X )$$

The relation  $R$  can be defined by a formula as in :

$$\forall x,y ( x \in Y \wedge y \in Y \Rightarrow \mathbf{TC}(\text{“}u \in Y \wedge v \in Y \wedge \text{edg}(u,v)\text{”}, x, y)$$

expressing that  $G[Y]$  is connected (note that  $Y$  is free in  $R$ ).

*Application* :  $G$  contains (fixed)  $H$  as a minor where  $V_H = \{1, \dots, p\}$  : there exist disjoint sets of vertices  $X_1, \dots, X_p$  in  $G$  such that each  $G[X_i]$  is connected and, for every edge  $i - j$  of  $H$ , there is an edge in  $G$  between  $X_i$  and  $X_j$  (denoted by  $\text{Link}(X_i, X_j)$  ).

*Consequence* : planarity is MSO-expressible (no minor  $K_5$  or  $K_{3,3}$ ).

## Provably non-expressible properties

$G$  is *isomorphic to  $K_{p,p}$*  for some  $p$  (*not fixed*; needs equipotence of two sets, hence quantification over binary relations to say that there is a **bijection**).

$G$  has a *nontrivial automorphism*, or is *regular*, i.e., has all its vertices of same degree.

## Exercises

- 1) Write an MSO-sentence expressing that the considered *simple* graph is a tree. (“Simple” is not MSO expressible).
- 2) Write an MSO-formula with free variables  $x, y, z$  expressing that, in a tree (undirected and unrooted), if one takes  $x$  as root, then  $y$  is an ancestor of  $z$ .
- 3) A nonempty word over alphabet  $\{a, b\}$  can be considered as a directed path with unary relations  $lab_a$  and  $lab_b$  on vertices representing the sets of occurrences of letters  $a$  and  $b$ . Prove that every regular language over  $\{a, b\}$  is MSO-definable.
- 4) A complete bipartite graph  $K_{n,m}$  has a Hamiltonian cycle if and only if  $n=m$ . Construct such a graph “over” any word in  $\{a, b\}^+$  having at least one  $a$  and at least one  $b$ . Deduce from the (well-known) converse of 3) that Hamiltonicity is not MSO-expressible.

## From MS formulas to automata for “clique-width” terms

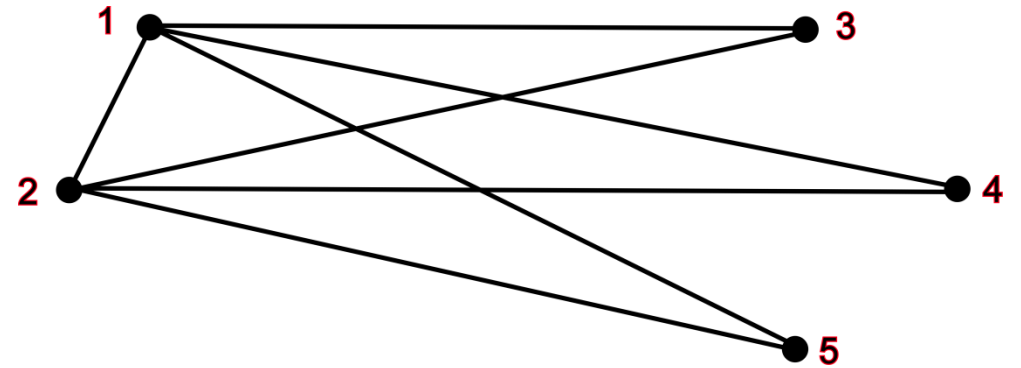
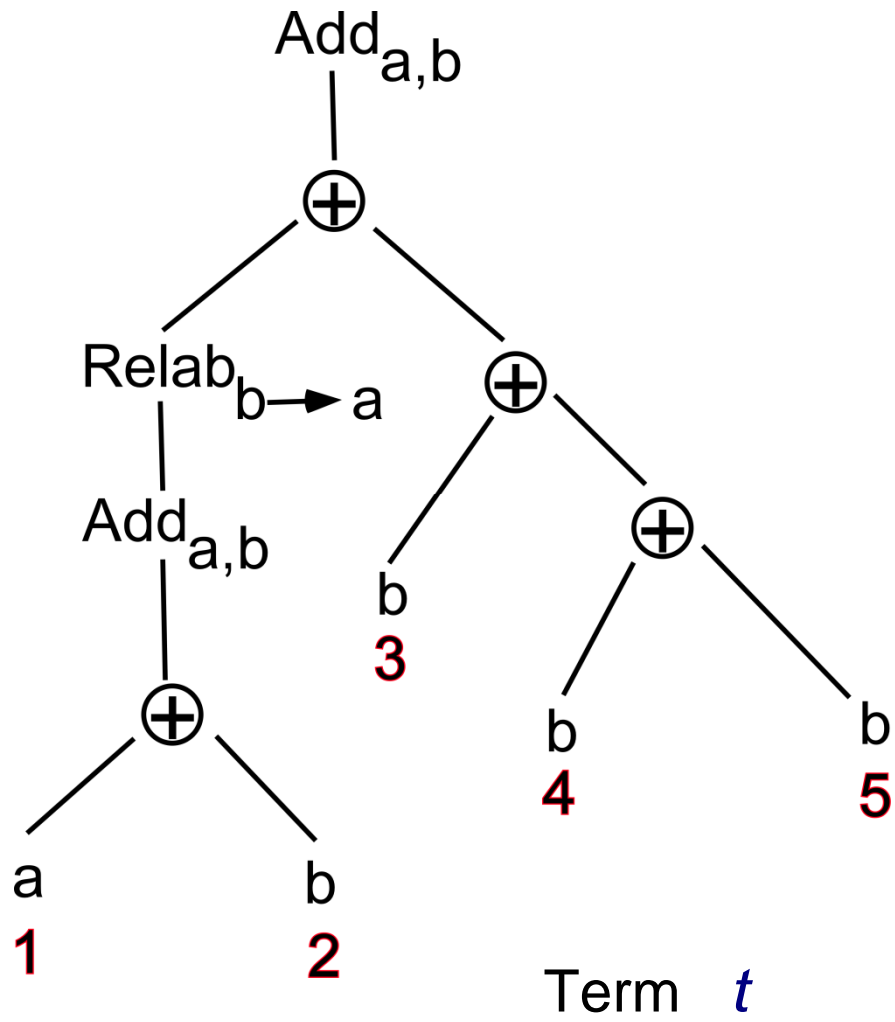
F = the countable set of operations and constants :

$$\mathbf{a}, \emptyset, \oplus, \overrightarrow{Add}_{\mathbf{a},\mathbf{b}}, Add_{\mathbf{a},\mathbf{b}}, Relab_{\mathbf{a} \longrightarrow \mathbf{b}}$$

$G(t)$  = the graph defined by a term  $t$  in  $T(F)$ .

Its vertices are (in bijection with) the occurrences of the constants  $\mathbf{a}$  in  $t$  (those not  $\emptyset$ ).

# Example



Terms are equipped with Booleans that encode assignments of vertex sets  $V_1, \dots, V_p$  to the free set variables  $X_1, \dots, X_p$  of MSO formulas (*formulas are written without first-order variables*):

1) we replace in  $F$  each constant  $\mathbf{a}$  by the constants

$(\mathbf{a}, (w_1, \dots, w_p))$  where  $w_i \in \{0, 1\}$  : we get  $F^{(p)}$

(only constants are modified);

2) a term  $\mathbf{s}$  in  $T(F^{(p)})$  encodes a term  $t$  in  $T(F)$  and an assignment of sets  $V_1, \dots, V_p$  to the set variables  $X_1, \dots, X_p$  :

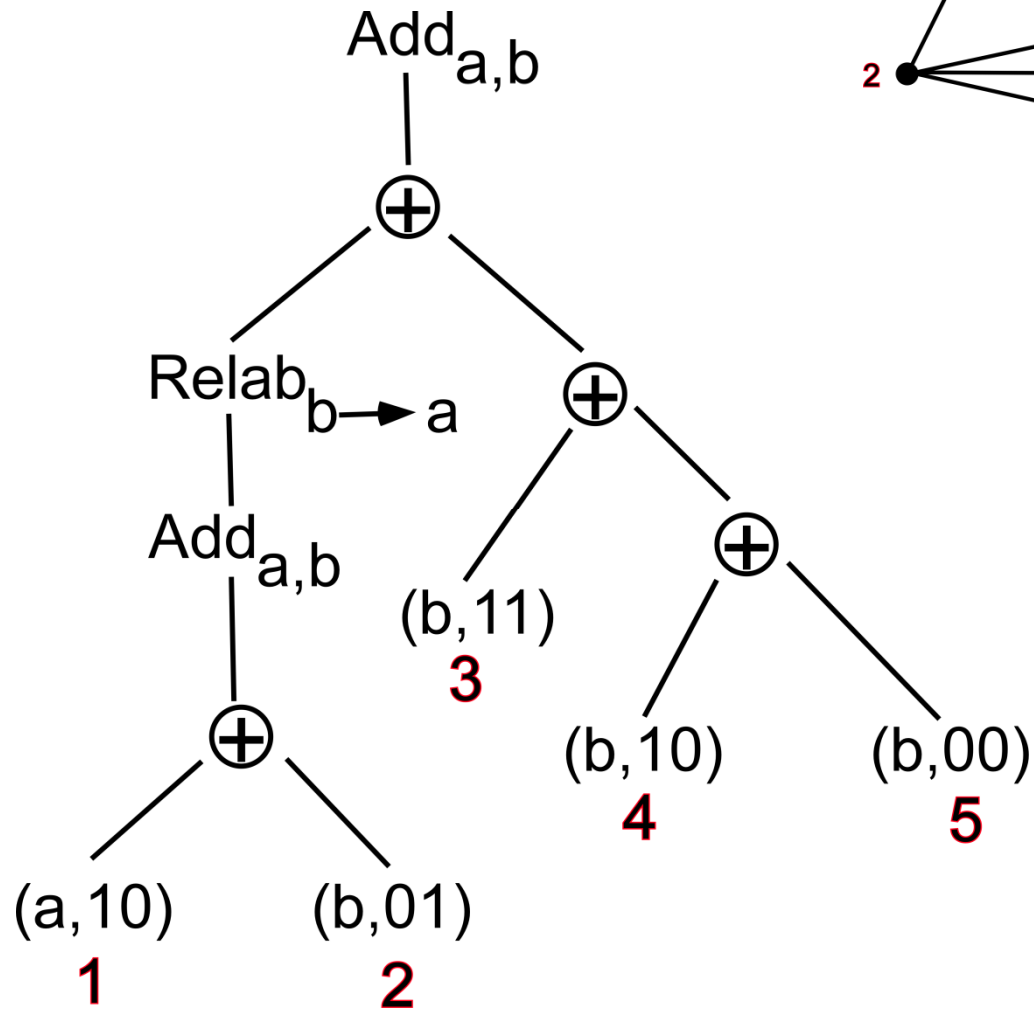
if  $u$  is an occurrence of  $(\mathbf{a}, (w_1, \dots, w_p))$ , then

$w_i = 1$  if and only if  $u \in V_i$ .

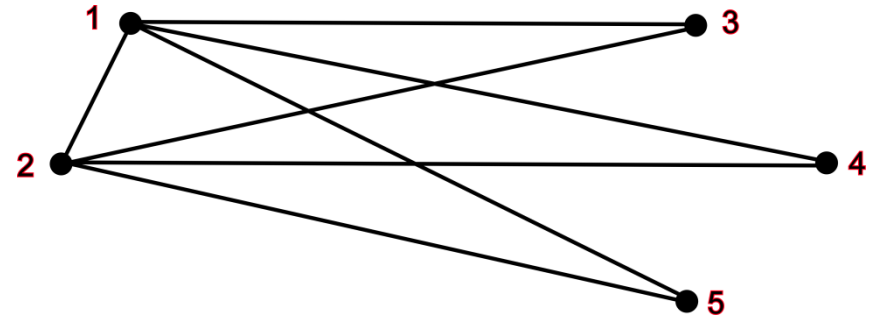
3)  $\mathbf{s}$  is denoted by  $t^*(V_1, \dots, V_p)$ .



Example (continued)



Term  $t^*(V_1, V_2)$



$$V_1 = \{1,3,4\}, V_2 = \{2,3\}$$

By an induction on  $\varphi$ , we construct for each  $\varphi(X_1, \dots, X_p)$  a finite (bottom-up) deterministic fly-automaton  $A(\varphi(X_1, \dots, X_p))$  that recognizes:

$$L(\varphi(X_1, \dots, X_p)) := \{ t^* (V_1, \dots, V_p) \in T(F^{(p)}) \mid (G(t), (V_1, \dots, V_p)) \models \varphi \}$$

*Theorem*: For each MS-sentence  $\varphi$ , the FPT-fly-automaton  $A(\varphi)$  accepts in time  $f(\varphi, k) \cdot |t|$  the terms  $t$  in  $T(F_k)$  such that  $G(t) \models \varphi$ .

## The inductive construction of $A(\varphi)$

Atomic formulas : discussed below.

For  $\wedge$  and  $\vee$  : product of two automata with appropriate accepting states.

For *negation* : exchange accepting / non-accepting states  
for a complete *deterministic* automaton;  
determinization preserves fly-automata.

(Recall that Fly-automata are run deterministically by means of a bottom-up computation of *finite* sets of states).

**Quantifications:** Formulas are written without  $\forall$

$$L( \exists X_{p+1} . \varphi(X_1, \dots, X_{p+1}) ) = \text{pr}( L ( \varphi(X_1, \dots, X_{p+1}) ) )$$

$$A( \exists X_{p+1} . \varphi(X_1, \dots, X_{p+1}) ) = \text{pr}( A ( \varphi(X_1, \dots, X_{p+1}) ) )$$

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

Yields a *non-deterministic* automaton even if  $A( \varphi(X_1, \dots, X_{p+1}) )$  is deterministic .

**Consequence** : Over a finite signature, the number of states of  $A( \varphi(\underline{X}) )$  is finite, but is an **h-iterated exponential**, where **h** = maximum nesting of negations. (There is no alternative construction, Frick and Grohe, 2004).

## Some tools for constructing automata

Substitutions and inverse images (“cylindrifications”).

1) If we know  $A(\varphi(X_1, X_2))$ , we can get easily  $A(\varphi(X_4, X_3))$ :

$$L(\varphi(X_4, X_3)) = h^{-1}(L(\varphi(X_1, X_2))) \quad \text{where}$$

$h$  maps  $(\mathbf{a}, (w_1, w_2, w_3, w_4))$  to  $(\mathbf{a}, (w_4, w_3))$

We take

$$A(\varphi(X_4, X_3)) = h^{-1}(A(\varphi(X_1, X_2)))$$

This construction preserves determinism and the number of states.

2) From  $A(\varphi(X_1, X_2))$ , we can get  $A(\varphi(X_3, X_1 \cup (X_2 \setminus X_4)))$  by  $h^{-1}$  with  $h$  mapping  $(\mathbf{a}, (w_1, w_2, w_3, w_4))$  to  $(\mathbf{a}, (w_3, w_1 \vee (w_2 \wedge \neg w_4)))$

*Set term*

## Relativization to subsets by inverse images.

If  $\varphi$  is a closed formula expressing a graph property  $P$ , its relativization  $\varphi[X_1]$  expresses that the subgraph induced on  $X_1$  satisfies  $P$ . To construct it, we can replace recursively

$$\exists y. \theta \quad \text{by} \quad \exists y. y \in X_1 \wedge \theta, \text{ etc...}$$

However, there is an easy transformation of automata :

Let  $h$  map  $(\mathbf{a}, 0)$  to  $\emptyset$  and  $(\mathbf{a}, 1)$  to  $\mathbf{a}$ .

$$L(\varphi[X_1]) = h^{-1}(L(\varphi))$$

Hence:

$$A(\varphi[X_1]) := h^{-1}(A(\varphi))$$

## The inductive construction (continued)

Complete *deterministic* FA for atomic formulas and basic graph properties : automaton over  $F^{(p)}$  recognizing the set of terms

$$t^* (V_1, \dots, V_p) \text{ in } L(\varphi(X_1, \dots, X_p))$$

*Atomic and basic formulas* :

$X_1 \subseteq X_2$ ,  $X_1 = \emptyset$ ,  $\text{Sgl}(X_1)$  ( $X_1$  is singleton)

$\text{Card}_{p,q}(X_1)$  : cardinality of  $X_1$  is  $\equiv p \pmod{q}$ ,

$\text{Card}_{<q}(X_1)$  : cardinality of  $X_1$  is  $< q$ .

→ Easy constructions with small numbers of states independent on  $F$ :  
respectively 2, 2, 3,  $q$ ,  $q+1$ .

*Example* : for  $X_1 \subseteq X_2$ , the term must have no constant (**a**, 10).

*Atomic formula* :  $\text{edg}(X_1, X_2)$  for directed edges

$\text{edg}(X_1, X_2)$  means :  $X_1 = \{x\} \wedge X_2 = \{y\} \wedge x \longrightarrow y$

The restriction to a set  $C$  of  $k$  labels uses  $k^2+k+3$  *states* :

0, Ok,  $a(1)$ ,  $a(2)$ ,  $ab$ , Error, for  $a, b$  in  $C$ ,  $a \neq b$

*Meaning of states* (at node  $u$  in  $t$ ; its subterm  $t/u$  defines  $G(t/u) \subseteq G(t)$ ).

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

Ok *Accepting state* :  $X_1 = \{v\}$ ,  $X_2 = \{w\}$ ,  $\text{edg}(v, w)$  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



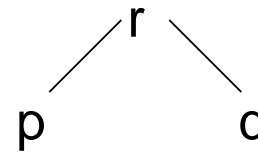
## Transition rules

For the constants based on **a** :

$(\mathbf{a},00) \rightarrow 0$  ;  $(\mathbf{a},10) \rightarrow \mathbf{a}(1)$  ;  $(\mathbf{a},01) \rightarrow \mathbf{a}(2)$  ;  $(\mathbf{a},11) \rightarrow \text{Error}$

For the binary operation  $\oplus$ :

(p,q,r are states)



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

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

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

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

Otherwise  $r := \text{Error}$

For unary operations  $\overrightarrow{Add}_{a,b}$

r  
|  
p

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

For unary operations  $Relab_{a \rightarrow b}$

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  $r := p$

# Number of states of some deterministic FA

restricted to  $k$  labels ;  $k$  = bound on clique-width.

| Property                      | Partition<br>( $X_1, \dots, X_p$ ) | edg(X,Y)  | NoEdge | Connected,<br>NoCycle<br>for degree $\leq d$ | Path(X,Y)    | Connected,<br>Nocycle |
|-------------------------------|------------------------------------|-----------|--------|--|--------------|-----------------------|
| Number of<br>states<br>$N(k)$ | 2                                  | $k^2+k+3$ | $2^k$  | $2^{O(d.k.k)}$                               | $2^{O(k.k)}$ | $2^{2^{O(k)}}$        |

We construct directly FA for important graph properties : NoEdge, Connectedness, NoCycle, Path(X,Y) (there is a path with vertices in Y between the two vertices of X).

This construction works well for formulas with no quantifier alternation, but that use “powerful atomic formulas”.

*Examples* :  $p$ -acyclic colorability

$$\exists X_1, \dots, X_p \left( \text{Partition}(X_1, \dots, X_p) \wedge \text{NoEdge}(X_1) \wedge \dots \wedge \text{NoEdge}(X_p) \wedge \dots \right. \\ \left. \dots \wedge \text{NoCycle}(X_i \cup X_j) \wedge \dots \right)$$

*Minor inclusion* :  $H$  simple, loop-free.  $\text{Vertices}(H) = \{v_1, \dots, v_p\}$

$$\exists X_1, \dots, X_p \left( \text{Disjoint}(X_1, \dots, X_p) \wedge \text{Conn}(X_1) \wedge \dots \wedge \text{Conn}(X_p) \wedge \dots \right. \\ \left. \dots \wedge \text{Link}(X_i, X_j) \wedge \dots \right)$$

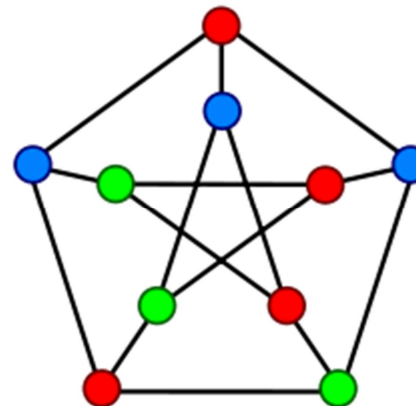
*Existence of “holes”* : odd induced cycles (to check *perfectness* ; one checks “anti-holes” on the edge-complement of the given graph).

## Some experiments using FA (by Irène Durand, LaBRI)

3-colorability of the 6 x 300 grid (of clique-width 8) in less than 2 hours,

4-acyclic-colorability of the **Petersen graph** (clique-width 7) in 17 minutes.

(3-colorable but not acyclically;  
**red** and **green** vertices  
induce a cycle).

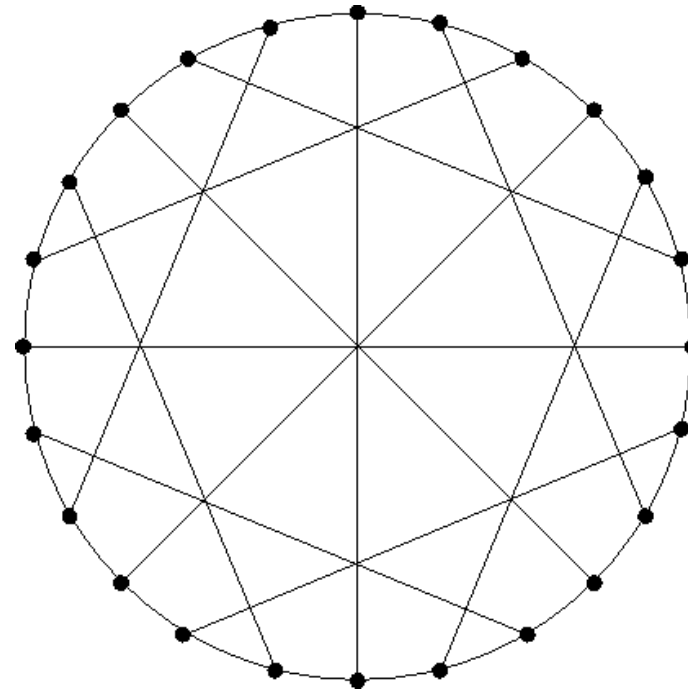


## The McGee graph

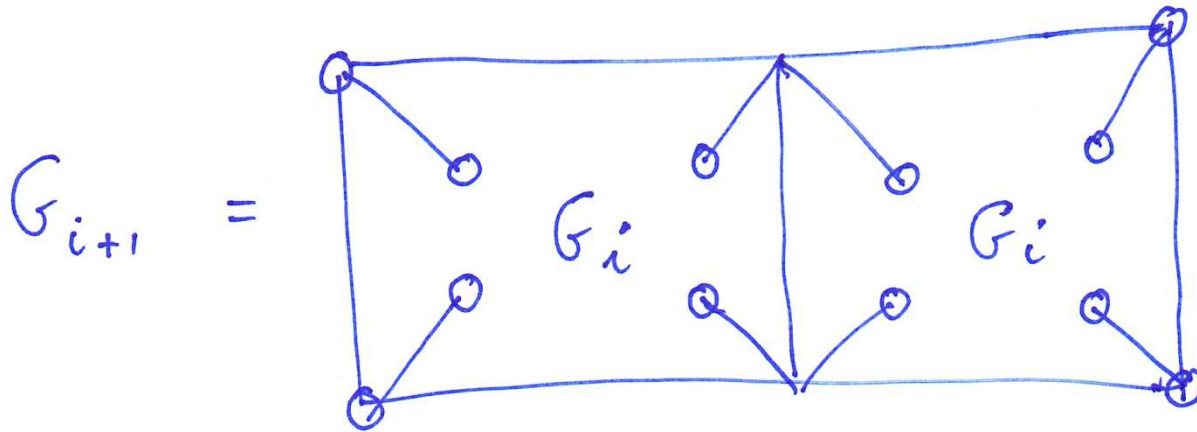
24 vertices on the external cycle.

Defined by a term  
with 10 labels (optimal ?)  
of size 99 and height 76.

It is 3-acyclically colourable.  
Checked in 11 hours with  
an *annotated term*  
(a preprocessing defined below).



## Recursively defined graphs



## Verification of 4-colorability

| $i$ | term  | dag    |
|-----|-------|--------|
| 9   | 88 mn | 1.5 mn |
| 20  | ??    | 4 mn   |
| 30  | ??    | 146 mn |

## Other tool : Annotations

At some positions in the given term, we attached some (finite) *contextual information*.

*Example :*

At position  $u$  in a term  $t$ , we attach the set

$\text{ADD}_t(u)$  = the set of pairs  $(a,b)$  such that some operation

$\text{Add}_{c,d}$  *above*  $u$  (hence, in its “context”) adds edges between the (eventual) vertices *below*  $u$  labelled by  $a$  and  $b$ .

These sets can be computed in *linear time* by means of a top-down traversal of  $t$ .



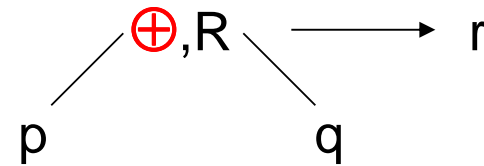
Certain automata on **annotated terms** may have less states.

*Example* :  $edg(X_1, X_2)$  :  $2k+3$  states instead of  $k^2 + k + 3$  :

0, Ok, **a**(1), **a**(2), Error, for **a** in C.

Transitions for  $\oplus$  annotated by R :

(p, q, r are states)



if  $p = 0$  then  $r := q$  ; if  $q = 0$  then  $r := p$  ;

if  $p = \mathbf{a}(1)$ ,  $q = \mathbf{b}(2)$  and  $(\mathbf{a}, \mathbf{b}) \in R \wedge$  then  $r := \text{Ok}$  ;

and if  $(\mathbf{a}, \mathbf{b}) \notin R \wedge$  then  $r := \text{Error}$  ;

if  $p = \mathbf{b}(2)$ ,  $q = \mathbf{a}(1)$  : *idem* ;

otherwise  $r := \text{Error}$ .

## Other examples :

For *Clique*[X] meaning that X induces a clique :

$$2^k + 2 \text{ states instead of } 2^{O(k.k)} .$$

For *Connectedness* : same states but they “shrink” quicker :

cf. the rules for  $Add_{a,c}$  in the first part of the lecture.

*Recall the example state* :  $q = \{ \{a\}, \{a,b\}, \{b,c,d\}, \{b,d,f\} \}$ .

And the transition :

$$Add_{a,c} : q \longrightarrow \{ \{a,b,c,d\}, \{b,d,f\} \}.$$