



Graphes, Logique et Algorithmes

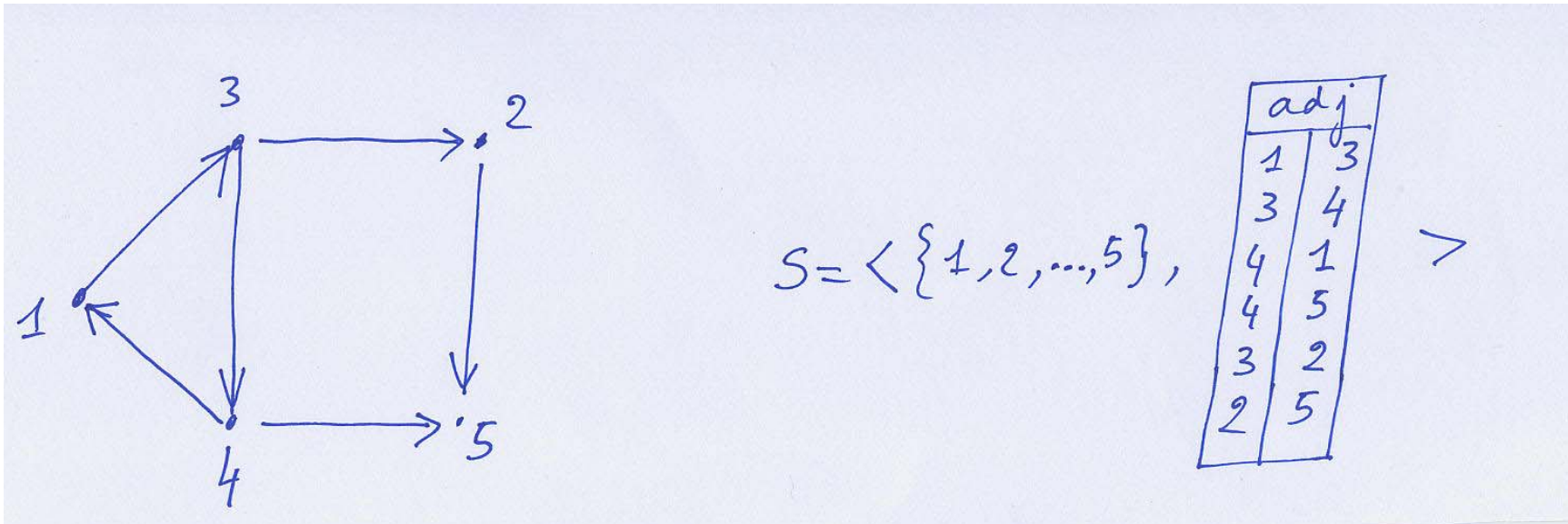
Bruno Courcelle

Equipe "Méthodes Formelles" du LaBRI

Logique : Calculabilité et complexité

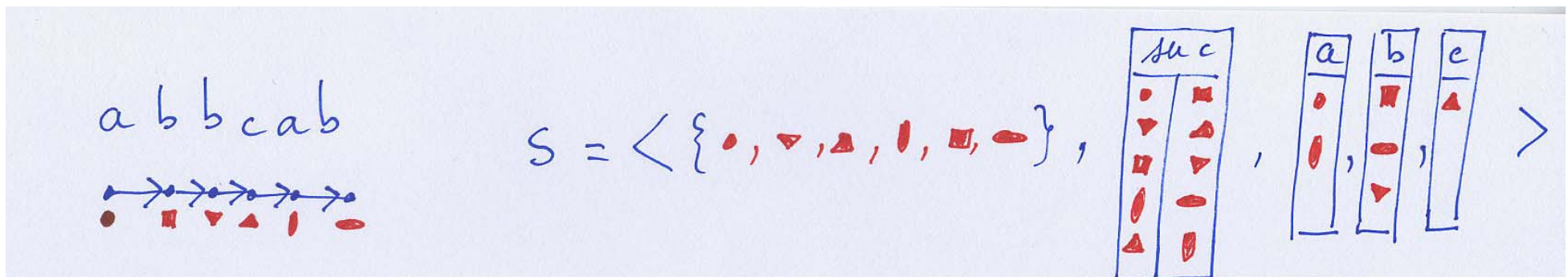
Démonstration : théorie et aide informatique

Théorie des modèles



Graphe

Structure représentative



Mot

Structure représentative

Propriétés d'objets combinatoires :

Expression en logique du premier ordre :

Ex. : tout sommet a deux voisins ; le graphe a diamètre au plus 5.

Expression en logique du second-ordre monadique :

Avec la construction : « il existe un ensemble de sommets ... »

Exemples : 3-colorabilité (Propriété NP-complète, “difficile” à tester):

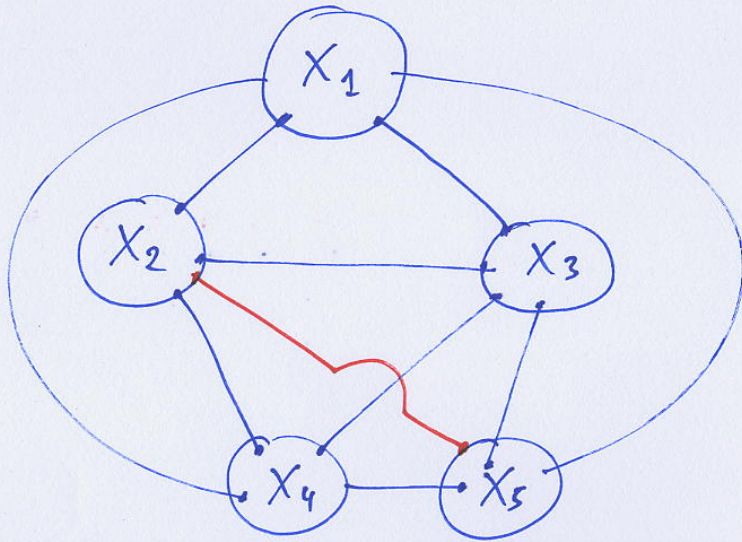
$$\exists X, Y (\text{“}X, Y \text{ sont disjoints”} \wedge \forall u, v \{ \text{edg}(u, v) \Rightarrow \\ [(u \in X \Rightarrow v \notin X) \wedge (u \in Y \Rightarrow v \notin Y) \wedge (u \in V - (X \cup Y) \Rightarrow v \notin V - (X \cup Y))] \})$$

Non connexité :

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

Non-planarité : $G \supseteq K_5$ ou $G \supseteq K_{3,3}$

$G \supseteq H$ de cette forme



$\exists X_1, X_2, X_3, X_4, X_5.$

[" $G[X_1]$ connexe" \wedge

\vdots

$G[X_5]$ connexe" \wedge

$\exists x, y. (x \in X_1 \wedge y \in X_2 \wedge "x-y")$
 \wedge

\vdots idem pour tous
les couples X_i, X_j
 $i \neq j$]

Problèmes algorithmiques :

Pour une classe C de graphes finis fixée :

1) *Décidabilité* logique :

Une formule donnée φ , est-elle vraie *pour tout graphe* de C ?

(Espoir de démonstration automatique).

2) *Vérification* : soit φ une formule fixée :

Peut-on vérifier efficacement (temps polynomial ou linéaire de préférence) si un graphe donné de C vérifie cette formule ?

(Espoir de construction automatique d'algorithmes).

Contrastes : Propriété du 1^{er} ordre : Polynomiale pour tout graphe.

Linéaire si degré borné fixé.

Prop. 2nd -ordre monadique : *3-coloriage* NP-complète, même si degré ≤ 4

Mais : Linéaire pour des graphes « arborescents ».

Cas des mots :

Equivalence entre « Automates finis »
et « Second-ordre monadique »

→ Solutions pour les deux problèmes de *décision* et de *vérification*.

Cas des graphes :

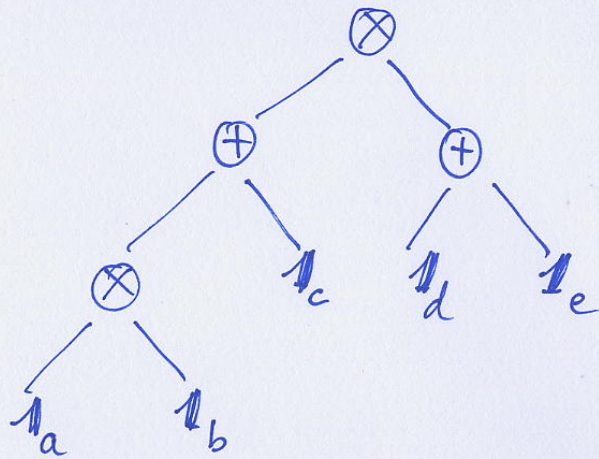
Extension aux graphes « arborescents » ou « décrits algébriquement ».

Cographe

$G \oplus H =$ union de G, H disjoints

$G \otimes H = G \oplus H +$ arêtes entre G et H

$\mathbb{1}_a =$ sommet isolé a

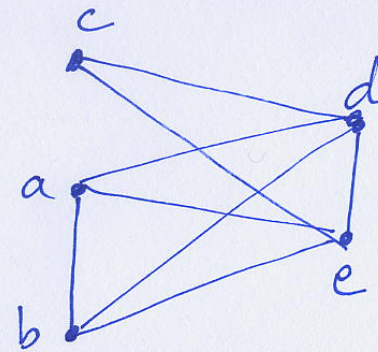


terme t

Calcul inductif

$\chi(G) =$ nombre chromatique

$$\left\{ \begin{array}{l} \chi(G(t_1 \oplus t_2)) = \max\{\chi(G(t_1)), \chi(G(t_2))\} \\ \chi(G(t_1 \otimes t_2)) = \chi(G(t_1)) + \chi(G(t_2)) \\ \chi(G(\mathbb{1})) = 1 \end{array} \right.$$



$G(t)$

Calcul du nombre d'arêtes $A(G)$:

$$A(G(t_1 \oplus t_2)) = A(G(t_1)) + A(G(t_2))$$

$$A(G(\perp)) = 0$$

$$A(G(t_1 \otimes t_2)) = A(G(t_1)) + A(G(t_2)) + \underline{N(G(t_1))} \cdot \underline{N(G(t_2))}$$

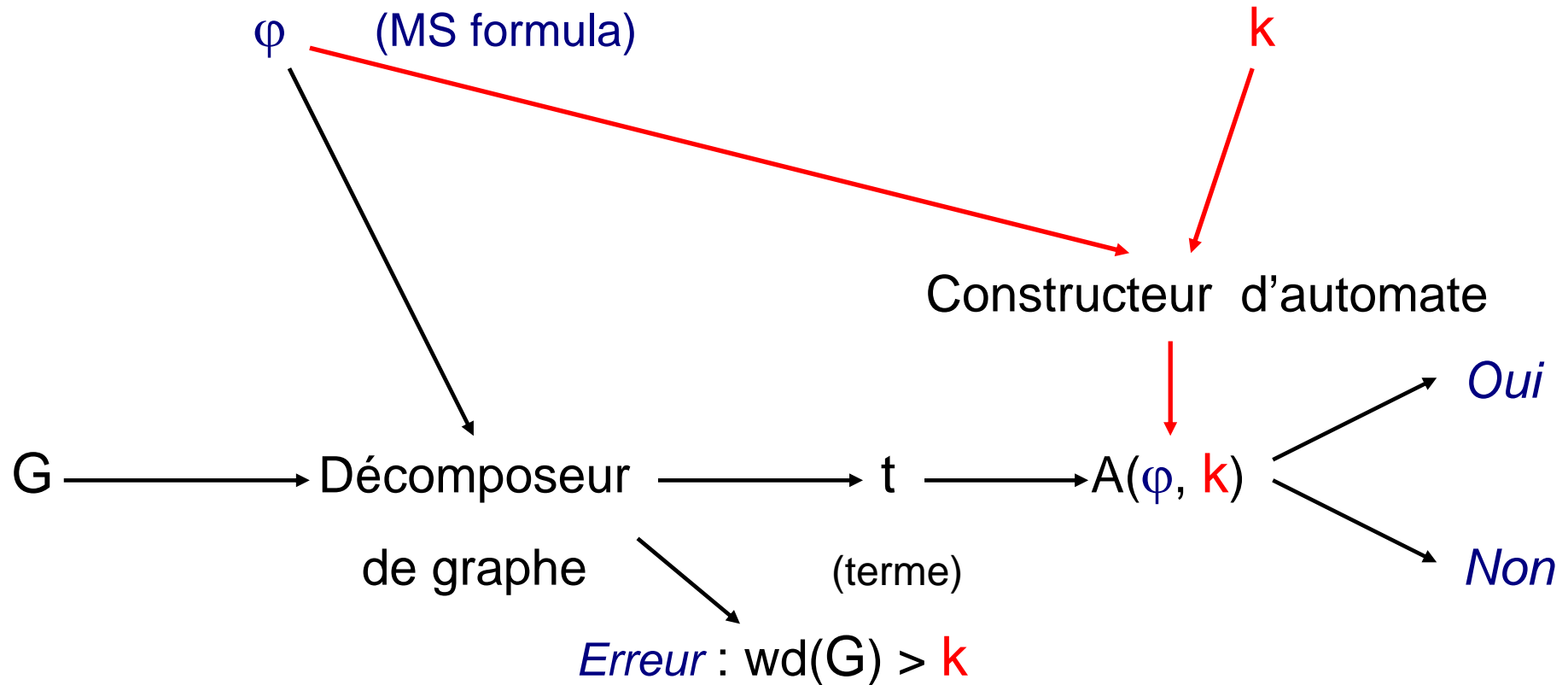
Valeur auxiliaire : Nombre de sommets :

$$N(G(t_1 \oplus t_2)) = N(G(t_1 \otimes t_2)) = N(G(t_1)) + N(G(t_2))$$

$$N(G(\perp)) = 1$$

Généralisation : Toute propriété monadique du second-ordre est vérifiable inductivement ~~avec~~ avec un (grand) nombre fini de propriétés auxiliaires (monadiques du second-ordre).
 \Rightarrow vérification en temps $O(|E|)$

Verification au moyen d'automates



→ fait 1 fois pour toutes, independant de G ;

$A(\varphi, k)$: automate de terme (wd = "largeur" du graphe)

Difficultés

1) *Taille des automates* : Calculs qui échouent par manque de mémoire

	<i>Cographe</i> s	<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

2) *Décomposition des graphes* : une algorithmique difficile.

Beaucoup de travail théorique et pratique, déjà fait (« largeur arborescente ») mais reste à faire (« largeur de clique »).

Remède pour la taille des automates

(travail en cours avec I. Durand) :

Automates « interprétés » plutôt que « compilés » :

Ne pas chercher à construire un automate de 2^{100} états pour l'utiliser sur un terme de taille 100, pour lequel seulement 99 transitions seront nécessaires (cas déterministe) :

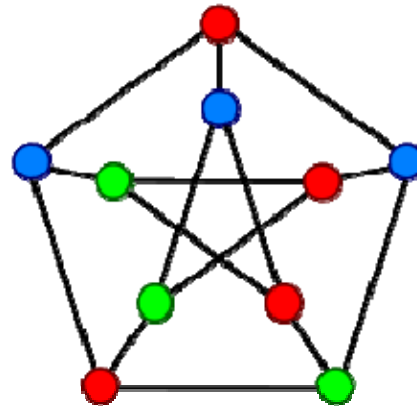
on les recalcule à chaque fois à partir d'une description combinatoire des états (paires d'ensembles, relations binaires).

Some experiments with fly-automata

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).



Autres sujets reliés :

Etiquetages des graphes pour test efficace de propriétés (« indexation » des sommets).

Liens entre *théorie des graphes* (propriétés de coloriations) et *logique*.

Méta-théorèmes algorithmiques fondés sur la *logique* (1^{er} ordre et 2nd ordre monadique) et la *théorie structurelle des graphes*

Descriptions *finies* et vérifiables de graphes infinis qui décrivent le comportement des programmes,

Grammaires de graphes, applications à des *visualisations structurées*.