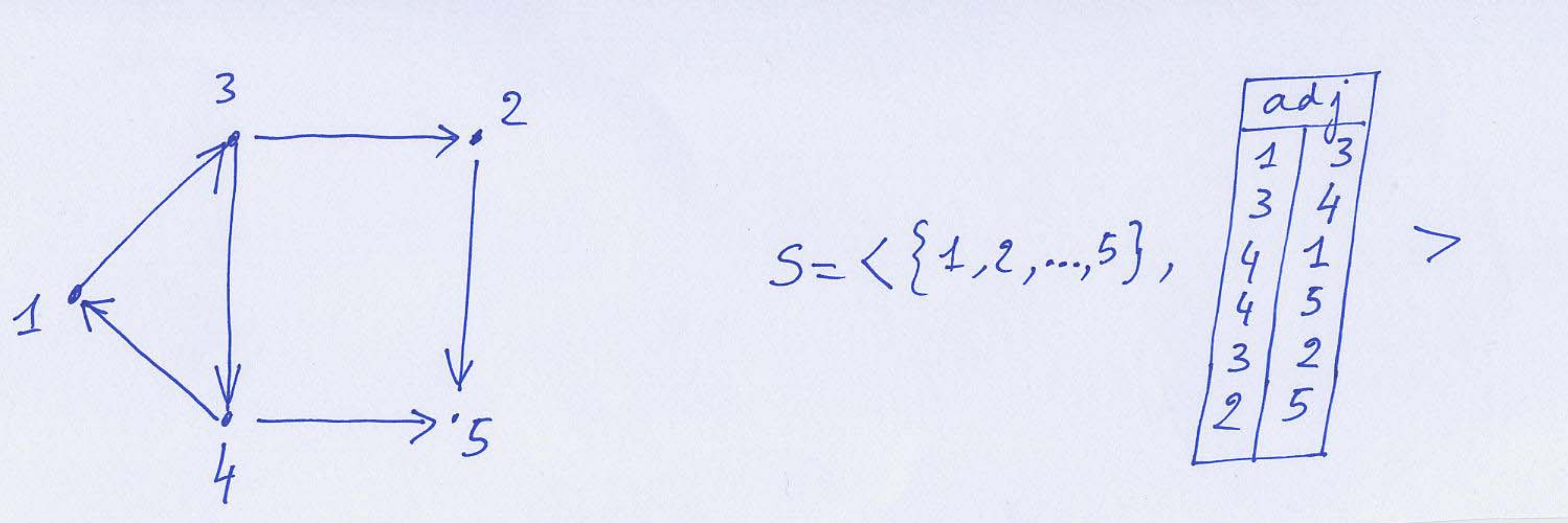
Graphes, Logique et Algorithmes

Bruno Courcelle

Logique : Calculabilité et complexité

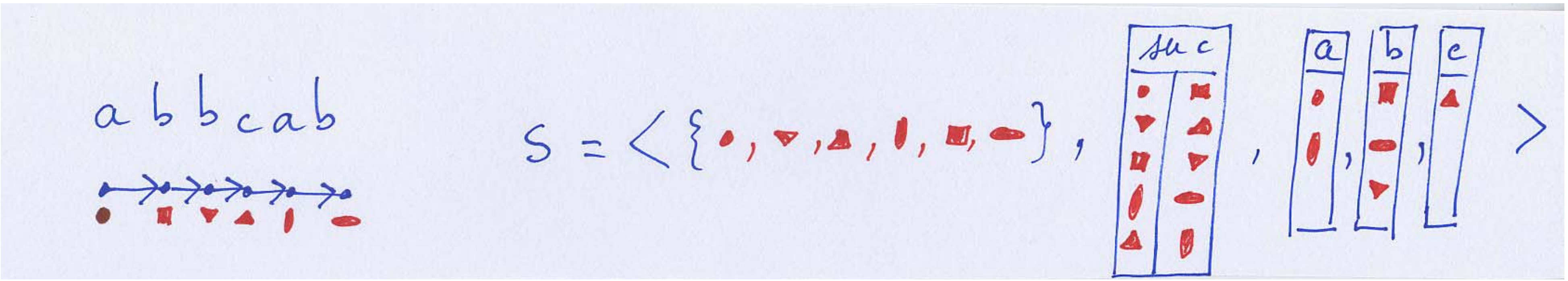
Démonstration : théorie et aide informatique

Théorie des modèles



Graphe

Structure représentative



Mot

Structure representative

Propriétés d'objets combinatoires :

Expression en logique du premier ordre :

Exemple : Tout sommet a deux successeurs

Expression en logique du second-ordre monadique :

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

Exemple : 3-colorabilité *(Propriété "difficile" à tester, NP-complète):*

$$\exists X,Y \Big(\text{"}X,Y \text{ are disjoint"} \wedge \forall u,v \{ edg(u,v) \Rightarrow$$

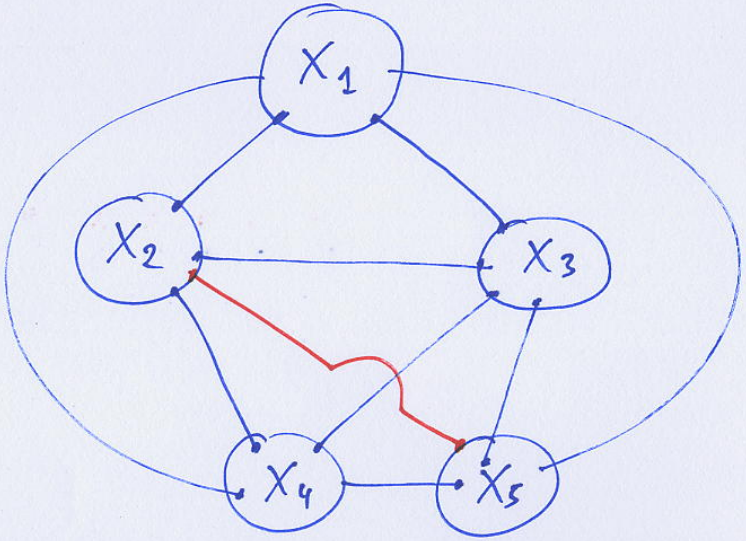
$$[(u \in X \Rightarrow v \notin X) \wedge (u \in Y \Rightarrow v \notin Y) \wedge (u \in V-(X \cup Y) \Rightarrow v \notin V-(X \cup Y)]\} \Big)$$

Non connexité :

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

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

$G \supseteq H$ de cette forme



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

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

⋮

"$G[X_5]$ connexe" $\wedge$

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

$\wedge$

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

Problèmes algorithmiques :

Pour une classe de graphes finis fixée *C* :

1) *Décidabilité* logique :

Etant une formule φ , est-elle vraie pour tout graphe de *C* ?

(Espoir de démonstration automatique).

2) *Vérification* : soit une formule φ 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 : Premier-ordre : "Polynomial" pour tout graphe.

"Linéaire" si degré borné fixé.

Second-ordre monadique : « NP-complet » général ou degré <4 (3-coloriage)

« Linéaire » pour graphe « arborescent ».

Cas des mots :

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

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

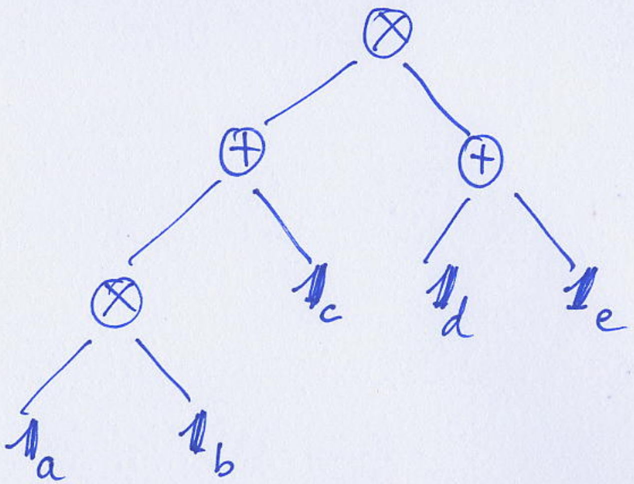
Extension aux graphes « arborescents » ou « décrits algébriquement ».

Cographes

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

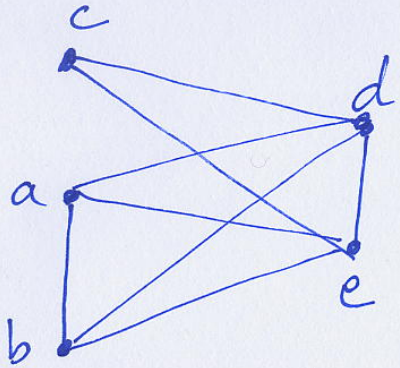
$G \otimes H$ = $G \oplus H$ + arêtes entre G et H

$\mathbb{1}_a$ = sommet isolé a



terme t

$\longrightarrow$



$G(t)$

Calcul inductif

$$\begin{cases} \chi(G(t_1 \oplus t_2)) = \operatorname{Max}\{\chi(G(t_1)), \chi(G(t_2))\} \\ \chi(G(t_1 \otimes t_2)) = \chi(G(t_1)) + \chi(G(t_2)) \\ \chi(G(\mathbb{1})) = 1 \end{cases}$$

$\chi(G)$ = nombre chromatique

Calcul du nombre d'arêtes $A(G)$:

$$A(G(t_1 \oplus t_2)) = A(G(t_1)) + A(G(t_2))$$
$$A(G(\mathbb{1})) = 0$$
$$A(G(t_1 \otimes t_2)) = A(G(t_1)) + A(G(t_2)) + \underline{N(G(t_1))} \cdot \underline{N(G(t_2))}$$

Valeur auxiliaire : Nombre de sommets :

$$N(G(t_1 \oplus t_2)) = N(G(t_1 \otimes t_2)) = N(G(t_1)) + N(G(t_2))$$
$$N(G(\mathbb{1})) = 1$$

Généralisation : Toute propriété monadique du second-ordre est vérifiable inductivement avec un (grand) nombre fini de propriétés auxiliaires (monadiques du second-ordre).

$\Rightarrow$ vérification en temps $O(|t|)$

Difficultés

Taille des automates : Calculs qui échouent par manque de mémoire

	Cographes	*Clique-width 3*
MaxDegree≤3	91 states	Space-Out
Degree≤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

Remèdes envisagés : Constructions directes; « interprétation des automates » plutôt que compilation.

Autres sujets reliés :

Etiquetages pour test efficace de propriétés,

Liens *dans les 2 sens* avec la théorie des graphes,

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

Grammaires de graphes, applications à des visualisations structurées.