Cours III
Cours MPRI 2009–20010

Michel Habib
habib@liafa.jussieu.fr
http://www.liafa.jussieu.fr/~habib

Chateau des rentiers, septembre 2009

Schedule

Some Applications of Chordal Graphs

Implémentation de LexBFS

Twin vertices

Reconnaissance des graphes d'intervalles

Research Problems

Propriété de Helly

Définition

Une famille de parties $\{T_i\}_{i\in I}$ satisfait la propriété de Helly si $J \subseteq I$ et $\forall i, j \in J \ T_i \cap T_j \neq \emptyset$ entraine $\cap_{i \in J} T_i \neq \emptyset$

Proposition

Les sous-arbres d'un arbre vérifient la propriété de Helly.

Application à un problème de recherche opérationnelle

Le stockage de produit dans des réfrigérateurs : partition en cliques maximales d'un graphe d'intervalles

Classes of twin vertices

Definition

x and y are called false twins, (resp. true twins) if $N(x) = N(y)$ (resp. $N(x) \cup \{x\} = N(y) \cup \{y\}$)

Algorithm Folklore

Data: $G = (V, E)$ a graph with n vertices and m edges
Result: The classes of false twin vertices
$Q \leftarrow \{V\}$
for *Every* $x \in V$ **do**
    $Q \leftarrow Refine(Q, N(x))$
end

Partition Refinement

If $Q = \{C_1, \ldots C_k\}$
$Refine(Q, S) = \{C_1 \cap S, C_1 - S, \ldots, C_k \cap S, C_k - S\}$

At the end, parts of Q have no splitter outside and therefore are modules.
Furthermore they have no splitter inside the part.
They are made up with false twins (non connected).

Complexity
$\Sigma_{x \in V} |N(x)| \in O(n+m)$

Implementation

$V = \{x_1, \ldots, x_7\}$
$P = \{C1, C2, C3\}$ and $S = \{x_3, x_4, x_5\}$

Data Structure

$$\begin{array}{cccccccccccccccccccc}
 & & & \mathrm{C1} & & & & & & & \mathrm{C2} & & & & & & \mathrm{C3} & & \\
x_1 & \nearrow\swarrow & \rightleftarrows & x_3 & \rightleftarrows & \nwarrow\searrow & x_5 & \rightleftarrows & x_2 & \nearrow\swarrow & \rightleftarrows & \nwarrow\searrow & x_7 & \rightleftarrows & x_4 & \nearrow\swarrow & \rightleftarrows & \nwarrow\searrow & x_6
\end{array}$$

First Step

Processing x_3

1,C1 C2 C3

$$x_3 \nearrow\swarrow \rightleftarrows x_1 \rightleftarrows \nwarrow\searrow x_5 \rightleftarrows x_2 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_7 \rightleftarrows x_4 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_6$$

Processing x_4

1, C1 C2 1, C3

$$x_3 \nearrow\swarrow \rightleftarrows x_1 \rightleftarrows \nwarrow\searrow x_5 \rightleftarrows x_2 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_7 \rightleftarrows x_4 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_6$$

x_5

2, C1 C2 1, C3

$$x_5 \nearrow\swarrow \rightleftarrows x_3 \rightleftarrows \nwarrow\searrow x_1 \rightleftarrows x_2 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_7 \rightleftarrows x_4 \nearrow\swarrow \rightleftarrows \nwarrow\searrow x_6$$

Second step

Maintain a list of the $Ci's$ that intersect S. List bounded by $|S|$.

Result

$$
\begin{array}{ccccccccccccccccc}
 & & \text{C'1} & & & & \text{C"1} & & & & \text{C2} & & & \text{C'3} & & \text{C"3} & \\
x_5 & \nearrow\swarrow & \rightleftarrows & \nwarrow\searrow & x_3 & \rightleftarrows\uparrow & x_1 & \downarrow\rightleftarrows & x_2 & \nearrow\swarrow & \rightleftarrows & \nwarrow\searrow & x_7 & \rightleftarrows\uparrow & x_4 \downarrow & \rightleftarrows\uparrow & x_6 \downarrow
\end{array}
$$

$Refine(P, S) = \{C'1, C"1, C2, C'3, C"3\}$
computed in $O(|S|)$

Exercise

- Propose an implementation in $O(|S|)$ of Refine stable which preserves a given initial ordering of the vertices of the elements x_i's. of the parts.
- Propose an implementation in an array.

1. C'est une structure de données parfaitement adaptée aux graphes. On affine à l'aide des voisinages ouverts ou fermés.
2. Très efficace (en théorie et en pratique)
3. Permet de travailler sur G ou son complémentaire, sans calculer le complémentaire
4. . . .

Some comments

- To obtain true twins (i.e. connected twins) either we study the complement or we filter with $N[x] = N(x) \cup \{x\}$
- It is not enough to obtain a linear time cograph recognition algorithm, it only yields an $O(n(n+m))$ time algorithm.
- Same problem for linear arrays has no obvious linear solution. Roughly same complexity as sorting.

Applications

Detection of multi-occurency in a list of subsets

Just consider the bipartite elements–subsets.

Constraint Satisfaction problems

It is a usual filtering technique.

Les questions

Ce que nous avons compris

Pourquoi LexBFS fournit un schéma simplicial

Ce qu'il reste à comprendre

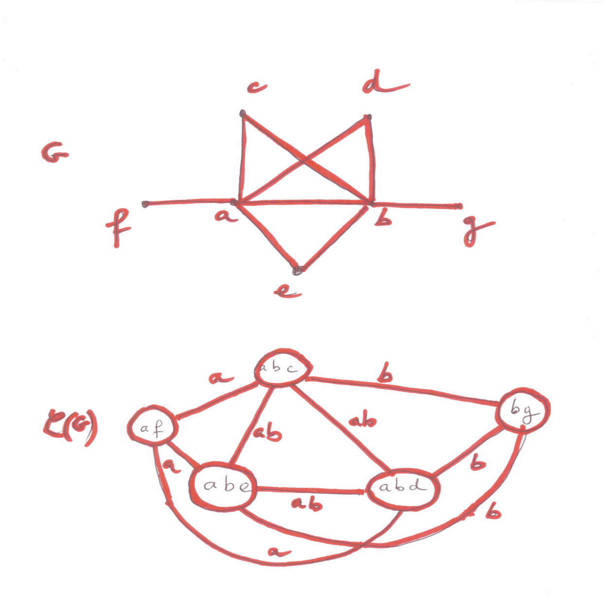
Pourquoi LexBFS termine sur une extrémité possible pour les graphes d'intervalles ?
Pourquoi LexBFS exécuté sur le complémentaire, termine sur une source possible d'un graphe de comparabilité ?

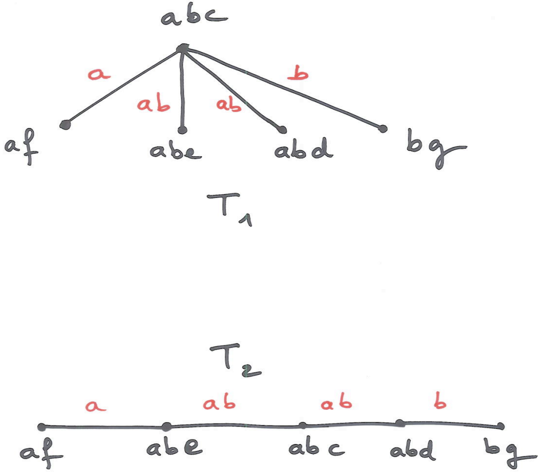
Ce qu'il reste à prouver

Pourquoi LexBFS fournit une bonne extrémité sur un path graphe ?

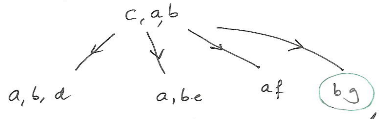
Un algorithme de partitionnement sur les cliques maximales du graphe

1. Calculer un arbre T de cliques maximales à l'aide d'un parcours lexBFS. Si T n'est pas un arbre de cliques maximales, G n'est pas chordal et donc pas d'intervalle.
2. Partir de la dernière clique maximale du parcours, affiner les cliques à l'aide du séparateur de l'arête pendante.
3. Continuer l'affinage jusqu'à ce que chaque classe soit un singleton
4. Si une classe n'est pas réduite à un singleton, recommencer récursivement à partir de la dernière clique maximale suivant LexBFS de la classe.


G
c
d
f
a
b
g
e
C(G)
abc
a
b
af
bg
ab
ab
a
abe
abd
b
ab
b
a


abc
a
ab
ab
b
af
abe
abd
bg
T1
T2
a
ab
ab
b
af
abe
abc
abd
bg

Lex BFS c a b | d | e | f | g



arbre de cliques maximales

af ⋮ abc, abd | bg

abe

a b

a n'apporte rien

af | abd abc | abe | bg

b n'apporte rien

af | abc | abd | abe | bg

Research Problems

1. For any graph G find in $O(n+m)$ $Reduce(G)$ obtained from G by deleting all pending vertices and keeping one vertex for each pair of twins.
2. G a bipartite graph, compute the vertices with maximal neighbourhood. (Another filtering scheme in CSP)
3. Transversal minimal $\in P$?
   Data : G a bipartite graph, $\mathcal{T}$ a set of transversal of G
   Question : Is $\mathcal{T}$ the set of all minimal transversals of G ?

Comments

- A transversal is simply a set of vertices intersecting all edges.
- Transversal minimal $\in NP \cap co - NP$?
- Kachyan proposed an algorithm in $O(n^{logn})$.
- A hot subject ! ! !