TD11 - Composantes fortement connexes et 2-SAT

On travaille sur des graphes *orientés*, que l'on suppose représentés par des *listes d'adjacence* : soit S l'ensemble des sommets et A l'ensemble des arcs, on associe à chaque sommet u dans S la liste $Adj[u]$ des éléments v tels qu'il existe un arc de u à v.

Exercice 1. *Parcours en profondeur*

Le parcours en profondeur d'un graphe G fait usage des constructions suivantes :
- A chaque sommet du graphe est associée une *couleur* : au début de l'exécution de la procédure, tous les sommets sont blancs. Lorsqu'un sommet est rencontré pour la première fois, il devient gris. On noircit enfin un sommet lorsque l'on est en fin de traitement, et que sa liste d'adjacence a été complètement examinée.
- Chaque sommet est également *daté* par l'algorithme, et ceci deux fois : pour un sommet u, $d[u]$ représente le moment où le sommet a été rencontré pour la première fois, et $f[u]$ indique le moment où l'on a fini d'explorer la liste d'adjacence de u. On se servira par conséquent d'une variable entière "temps" comme compteur événementiel pour la datation.
- La manière dont le graphe est parcouru déterminera aussi une relation de paternité entre les sommets, et l'on notera $\pi[v] = u$ pour dire que u est le père de v (selon l'exploration qui est faite, c'est à dire pendant le parcours v a été découvert à partir de u).

On définit alors le parcours en profondeur (**PP**) de la manière suivante :

PP(S,$Adj[]$,temps) :=
 `pour chaque` sommet u de S *Initialisation*
  `faire` couleur$[u] \leftarrow$ BLANC
  $\pi[u] \leftarrow$ NIL
 temps$\leftarrow 0$
 `pour chaque` sommet u de S *Exploration*
  `faire si` couleur$[u]$=BLANC
  `alors` **Visiter_PP**(u)

Visiter_PP(u) :=
 couleur$[u] \leftarrow$ GRIS
 $d[u] \leftarrow$temps $\leftarrow$temps$+1$
 `pour chaque` $v \in Adj[u]$
  `faire si` couleur$[v]$=BLANC
  `alors` $\pi[v] \leftarrow u$
  Visiter_PP(v)
 couleur$[u] \leftarrow$ NOIR
$f[u] \leftarrow$ temps$\leftarrow$ temps$+1$

1. Quelle est la complexité de PP ?

2. Que peut-on dire de $d[u]$, $f[u]$, $d[v]$ et $f[v]$, pour deux sommets u et v du graphe ?

3. Montrer que v est un descendant de u si et seulement si à l'instant $d[u]$, il existe un *chemin blanc* de u à v (*Lemme du chemin blanc*).

Exercice 2. *Composantes fortement connexes*

On définit la relation binaire R des sommets d'un graphe orienté par : xRy si et seulement s'il existe un chemin orienté de x à y et un chemin orienté de y à x.

1. Montrer que c'est une relation d'équivalence.

Les classes d'équivalence sont appelées *composantes fortement connexes* du graphe orienté. Ce sont les sous-ensembles maximaux du graphe tels qu'il existe un chemin allant d'un sommet u à un sommet v pour tous sommets u et v de l'ensemble.

2. Montrer que lors d'un parcours en profondeur, tous les sommets appartenant à la même composante fortement connexe sont dans le même arbre en profondeur (issu de l'application de PP sur G).

Pour tout sommet u, et pour une exécution d'un parcours en profondeur, on appelle $\phi(u)$ le sommet w tel que $u \rightsquigarrow w$ (i.e. il existe un chemin éventuellement nul de u à w) et $f[w]$ est maximal. On appelle $\phi(u)$ l'*aïeul de* u.

3. Montrer que $\phi(u)$ est un ancêtre de u.

Remarque : par conséquent, u et $\phi(u)$ sont dans la même composante fortement connexe, et plus globalement, deux sommets sont dans la même composante fortement connexe si et seulement si ils ont le même aïeul.

Pour calculer les composantes fortement connexes d'un graphe G on considère l'algorithme suivant :
– Exécuter le parcours en profondeur de G pour calculer f sur l'ensemble des sommets.
– Calculer le graphe ${}^tG$ qui a les mêmes sommets que G mais où le sens des arcs est inversé.
– Exécuter le parcours en profondeur de ${}^tG$ en appelant dans la boucle principale les sommets par f décroissant.
Les composantes fortement connexes de G (et aussi de ${}^tG$) sont les arbres ainsi obtenus.

4. Montrer la correction de l'algorithme précédent (par exemple en montrant par récurrence sur le nombre d'arbres déjà calculés qu'une composante fortement connexe, et donc l'ensemble des sommets de même aïeul selon G, est un arbre).

5. Quelle est sa complexité ?

Exercice 3. *2-SAT*

Montrer que 2-SAT peut être résolu en temps $O(n+m)$ où n est le nombre de variables et m le nombre de closes.
Indication. On crée le graphe orienté suivant :
Pour chaque variable x, on crée deux sommets correspondant aux littéraux x et $\neg x$.
Pour chaque close $\neg x \vee y$, on ajoute une arête allant de x vers y.