

Graphes orientés & Recherche Arborescente

Maria Malek

9 mars 2009

Définition & Généralités

Notations, Représentation & Terminologie

Notions dans le contexte orienté

Degrés intérieurs et extérieurs

Composantes fortement connexes

Représentation en machine

Graphes orientés sans circuits

Tri Topologique

Arborescences

Caractéristiques

Recherche Arborescente

Parcours en Profondeur d'abord dans une arborescence

Parcours en Profondeur d'abord dans un graphe orienté strict

Parcours en Profondeur d'abord étendu

Algorithme de tri topologique

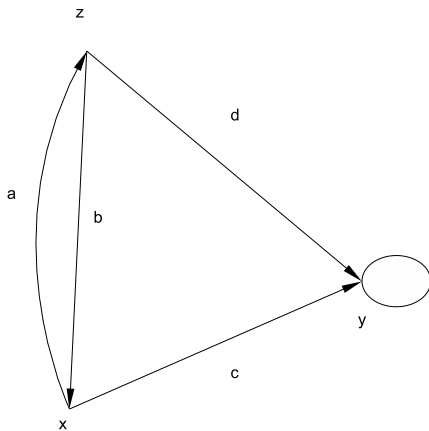
Un graphe (orienté) G est défini par :

- ▶ Un ensemble X de sommets (non vide).
- ▶ Un ensemble A d'arcs (arêtes orientées).
- ▶ A chaque arc a sont associés un couple de sommets (x,y) appelés extrémités de a . x est appelé l'origine.

Notations

- ▶ On note $G=(X,E)$, $X(G)$, $A(G)$.
- ▶ On note n le cardinal de G : le nombre de sommets.
- ▶ On note m le cardinal de A .

Représentation d'un graphe orienté



Terminologie

- ▶ (x,y) est le couple associé à un arc a :
 - ▶ a va du sommet x au sommet y , a sort de x et entre dans y .
 - ▶ L'arc a est incident au sommet x et au sommet y .
 - ▶ y est appelé successeur de x . x est prédécesseur de y .
 - ▶ Si $x=y$ a est une boucle.
 - ▶ Deux arcs a et a' ayant le même couple (x,y) sont dites parallèles (ou multiples).
 - ▶ Deux arcs ayant comme couple de sommets (x,y) et (y,x) sont dits opposés.
- ▶ Un graphe est dit **strict** s'il n'a ni boucles ni arcs multiples.
- ▶ Dans un graphe **strict** un arc s'identifie au couple de ses deux extrémités : $a=(x,y)$
- ▶ Un graphe orienté strict est dit **symétrique** si pour tout arc (x,y) il existe l'arc opposé (y,x) .

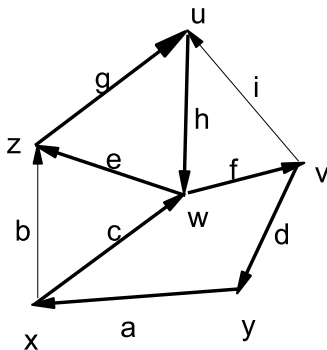
Graphe non orienté associé

- ▶ Étant donné un graphe orienté G , le graphe non orienté associé est obtenu en remplaçant chaque arc dans G par une arête
- ▶ Étant donné un graphe non orienté G , il y a 2^m graphes orientés dont le graphe non orienté associé est G .

Notions de graphes orientés

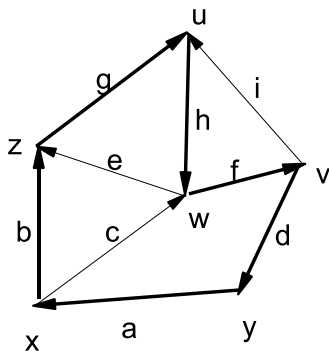
- ▶ Même notions vues en non orientés se transposent ici :
 - ▶ Sous graphes, Sous graphes engendrés, graphes partiels ;
 - ▶ chemins (chaînes orientées), circuits (cycles orientés).
 - ▶ Un chemin est simple s'il ne passe pas deux fois par un même arc.
 - ▶ Un chemin est élémentaire s'il ne passe pas deux fois par un même sommet.

Exemple d'un circuit - 1



Circuit non élémentaire :
 (x,c,w,e,z,g,u,h,w,f,v,d,y,a,x)

Exemple d'un circuit - 2



Circuit élémentaire (x,b,z,g,u,h,w,f,v,d,y,a,x)

Degrés intérieurs et extérieurs

- ▶ Le degré intérieur d'un sommet x d'un graphe orienté G est le nombre d'arcs entrants dans x $d_G^-(x)$.
- ▶ Le degré extérieur d'un sommet x d'un graphe orienté G est le nombre d'arcs sortants de x $d_G^+(x)$.
- ▶ $\sum_{x \in X} d_G^-(x) = \sum_{x \in X} d_G^+(x) = m$
- ▶ Remarquer que $d_G(x) = d_G^-(x) + d_G^+(x)$
- ▶ $\sum_{x \in X} d_G(x) = \sum_{x \in X} d_G^-(x) + \sum_{x \in X} d_G^+(x) = 2m$

Composantes fortement connexes

- ▶ Un graphe orienté G est fortement connexe si quels que soient deux sommets x et y , il existe un chemin allant de x à y .
- ▶ Une composante fortement connexe de G est un sous graphe engendré de G fortement connexe et maximal.
- ▶ Les composantes fortement connexes d'un graphe sont à deux à deux disjointes.
- ▶ Des sommets appartenant à un même circuit sont dans une même composante fortement connexe (pourquoi?).

Représentation en machine - 1

► Trois représentations possibles :

1. Tableau ou liste des arcs :

arcs	couples des extrémités
1	(2,1)
2	(1,3)
3	(2,3)
4	(2,4)
5	(4,3)

2. Matrice d'adjacence.

3. Liste des voisins (successeurs ou prédécesseurs).

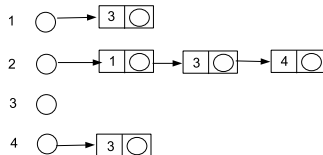
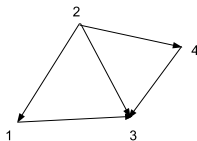
Représentation en machine - 2

- ▶ Trois représentations possibles :
 1. Tableau ou liste des arcs.
 2. Matrice d'adjacence :

	1	2	3	4
1	0	0	1	0
2	1	0	1	1
3	0	0	0	0
4	0	0	1	0

3. Liste des voisins (successeurs ou prédécesseurs).

Représentation en machine - 3



Listes de successeurs

Graphes Orientés Valués

- ▶ Un graphe dont les arcs sont valués.
- ▶ Un graphe valué $G(X,E)$ est un graphe avec une application :
 $v : A \rightarrow R$
- ▶ Un graphe valué *peut être représenté* par une matrice d'adjacence.
- ▶ L'absence d'une arc entre deux sommets x_i et x_j est représentée par : $v(x_i, x_j) = \infty$ ssi $(x_i x_j) \notin E$
- ▶ Par extension, $v(x_i, x_i) = 0$

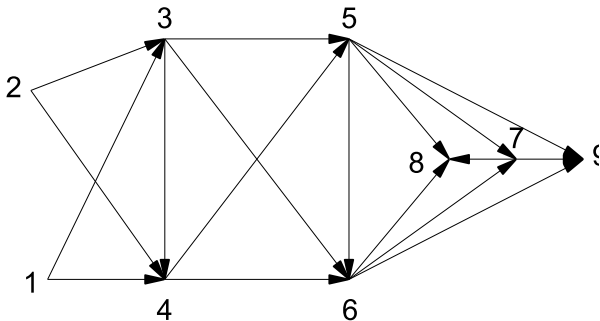
Graphes Orientés sans circuits

- ▶ Un sommet **source** d'un graphe orienté est un sommet dont le degré intérieur est nul.
- ▶ Un sommet **puits** d'un graphe orienté est un sommet dont le degré extérieur est nul.
- ▶ **LEMME I** Dans un graphe orienté sans circuit il existe un sommet source et un sommet puits.
- ▶ **PREUVE** (idée) Par construction du chemin des prédécesseurs (res. successeurs) d'un sommet donné x .

Tri topologique

- ▶ On appelle tri topologique des sommets d'un graphe $G=(X,A)$ connexe et strict :
 - ▶ une bijection f de X sur l'intervalle des entiers de 1 à n (n étant le nombre de sommets) telle que pour un arc (x,y) on $f(x) < f(y)$.
- ▶ **PROPOSITION I** Un graphe orienté est sans circuits ssi il admet un tri topologique de ses sommets.
- ▶ En pratique, on associe au tri topologique la notion de niveaux, les prédécesseurs d'un sommet donné sont dans des niveaux inférieurs.

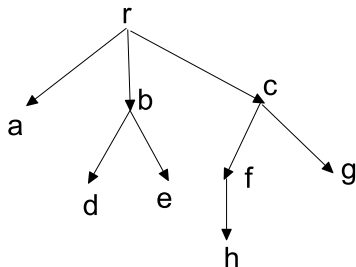
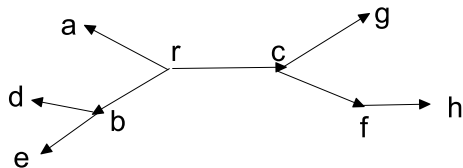
Exemple : Tri Topologique



Arborescence - 1

- ▶ On appelle **racine** d'un graphe orienté G un sommet r tel qu'il existe pour tout sommet x de G un chemin de r à x .
- ▶ Une arborescence est un graphe orienté qui admet une racine et dont le graphe non orienté associé est un arbre.
- ▶ On représente une arborescence avec la racine en haut (voir figure sur le transparent suivant). On omet parfois les flèches indiquant les orientations.
- ▶ Terminologies :
 - ▶ La profondeur d'un sommet est sa distance à la racine
 - ▶ un sommet puits est appelé feuille, sinon, un sommet est appelé nœud.
 - ▶ Un successeur d'un sommet x est appelé fils ou enfant de x .
 - ▶ Un prédécesseur d'un sommet x est appelé parent ou père de x .
 - ▶ D'autres termes sont utilisés : frère, descendant, etc.

Arborescence : exemple



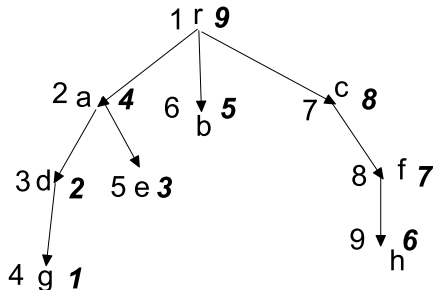
Arborescence - 2

- ▶ **THÉORÈME I** Les conditions suivantes sont équivalentes :
 1. G est une arborescente.
 2. G a une racine et son graphe non orienté associé est acyclique.
 3. G a une racine et on a $m=n-1$.
 4. Il existe un sommet r tel que pour tout sommet x de G il existe un chemin unique de r à x .
 5. G est connexe et on $d^-(x) = 1$ pour tout sommet x sauf pour un sommet r pour lequel $d^-(r) = 0$ (condition des degrés intérieurs)
 6. Le graphe non orienté associé de G est acyclique et on a la condition des degrés intérieurs.
 7. G est sans circuits et on a la condition des degrés intérieurs.
- ▶ **PREUVE** (idée)
 - ▶ Equivalence de 1,2 et 3.
 - ▶ Equivalence de 6 et 7.
 - ▶ $1 \Rightarrow 4, 4 \Rightarrow 5, 5 \Rightarrow 6, 6 \Rightarrow 1$

Parcours en profondeur d'abord - 1

- ▶ PROCÉDURE `parcoursArbre (T,s)`
 - ▶ POUR $t \in \text{fils}(s)$ FAIRE
 - ▶ `parcoursArbre (T,t)`
- ▶ Complexité : $O(m)$, avec $m=n-1$, donc $O(n)$. Ou bien $O(k^P)$, k étant le nombre maximum de fils d'un sommet x et p étant la profondeur de l'arbre.

Parcours d'Arborescence : exemple



Ordre de prévisites et de *postvisites*

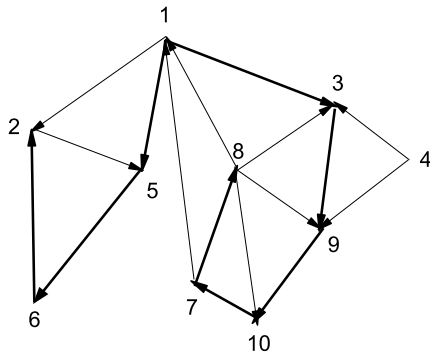
Parcours en profondeur d'abord - 2

- ▶ **procédure** `parcoursGrapheRécursif(G,s)`
 - visite(S) ← vrai*
 - tantque** *succ(S) <> null* **faire**
 - t ← suc(S).som*
 - suc(S) ← suc(s).suiv*
 - si** non *visite(t)* **alors**
 - Prévisite de t
 - `parcoursGrapheRécursif(G,t)`
 - Postvisite de t
 - sinon**
 - Revisite de t
 - fin si**
 - fin tantque**

Parcours en profondeur d'abord - 3

- ▶ PROCEDURE `parcoursGraphe (G,r)`
 - ▶ Prévisite de r
 - ▶ `parcoursGrapheRécursif (G,r)`
 - ▶ Postvisite de r
- ▶ Exemple (voir transparent suivant)
 - ▶ Prévisite de 1,3,9,10,7, revisite de 1
 - ▶ Prévisite de 8, revisite de 1,3,9,10
 - ▶ Postvisite de 8. etc.

Parcours en profondeur : exemple



Parcours en profondeur d'abord - 4

- ▶ **PROPOSITION II** Lors d'un parcours en profondeur d'un graphe orienté strict, tout sommet accessible par un chemin depuis le sommet origine r du parcours est visité, en prévisite, puis en postvisite, éventuellement aussi en revisite.
- ▶ Complexité de l'algorithme :
 - ▶ Complexité du parcours : $O(m)$
 - ▶ Complexité de traitement du tableau `visite()` : $O(n)$

Parcours en profondeur étendu - 1

- ▶ Objectif : Tous les sommets du graphe orienté strict doivent être visités.
- ▶ Le tableau `visite()` est une variable globale.
- ▶ **procedure** `parcoursEtendu(G)`
 - $r \leftarrow 1$
 - tantque** `non visite(r)` **faire**
 - `parcoursGraphe(G,r)`
 - tantque** $r \leq n$ ET `visite(r)` **faire**
 - $r \leftarrow r + 1$
 - fin tantque**
- fin tantque**

Parcours en profondeur étendu - 2

- ▶ **PROPOSITION III** Lors d'un parcours en profondeur étendu d'un graphe orienté strict, tout sommet est visité, en prévisite, puis en postvisite, éventuellement aussi en revisite.
- ▶ **PROPOSITION IV** Application au tri topologique
 1. Un graphe orienté strict G est sans circuits ssi lors d'un parcours en profondeur étendu de ce graphe, lorsqu'un sommet est revisité, il est alors déjà postvisité.
 2. Si le graphe G est sans circuits, le postordre des sommets dans un parcours en profondeur étendu est l'inverse de celui d'un tri topologique.

Algorithme de tri topologique

- ▶ Effectuer un parcours en profondeur du graphe G
 - ▶ En postvisite de s marquer s postvisité et le numéroter en décroissant de n à 1 .
 - ▶ En revisite de t : si t n'est pas postvisité, arrêter car il existe un circuit dans le graphe.