Graphes

4. Parcours, sommets ascendants/descendants et connexité

Solen Quiniou

solen.quiniou@univ-nantes.fr

IUT de Nantes

Année 2021-2022 – Info 1 (Semestre 2)

[Mise à jour du 27 janvier 2022]

Plan du cours

Parcours d'un graphe : algorithme général

```
pour chaque x ∈ S faire
    etat[x] = non_atteint;
fin
X = ∅ ; a_traiter = ∅;
pour chaque x ∈ S faire
    si etat[x] == non_atteint alors
        a_traiter = a_traiter ∪ {x};
        etat[x] = atteint;
        tant que a_traiter ≠ ∅ faire
            y = Choix(a_traiter);
            a_traiter = a_traiter \ {y};
            pour chaque z adjacent de y faire
                si etat[z] == non_atteint alors
                    T = T ∪ {(y, z)};
                    etat[z] = atteint;
                    a_traiter = a_traiter ∪ {z} ;
                fin
            fin
            etat[y] = examine ;
        fin
    fin
fin
```

Parcours en largeur : principe

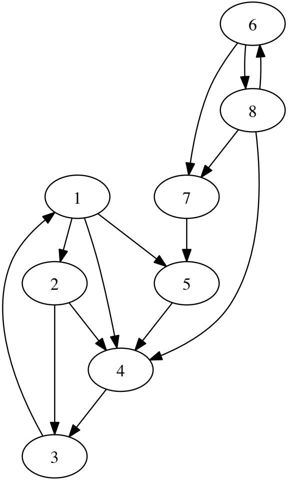
Le principe de choix d'un sommet sera le fait qu'un sommet atteint est toujours un sommet traité, c'est-à-dire que la variable $\mathrm{a\_traiter}$ sera une **file** (« First In First Out »).
Ensuite, on numérotera les sommets et on les parcourra dans l'ordre croissant.

Lorsque l'on fait un parcours en largeur à partir d'un sommet adjacent de x, on atteint d'abord les adjacents et ensuite les adjacents des adjacents (sauf ceux déjà atteints) et ainsi de suite.

Parcours en largeur

```
pour chaque x ∈ S faire
    etat[x] = non_atteint;
fin
index = 1 ; a_traiter = ∅;
pour chaque z ∈ S faire
    si etat[z] == non_atteint alors
        pere[z] = NIL;
        ordre[z] = index ; index = index + 1;
        a_traiter.AjouterEnQueue(z);
        tant que a_traiter ≠ ∅ faire
            x = Tete(a_traiter);
            pour chaque y adjacent de x faire
                si etat[y] == non_atteint alors
                    etat[y] = atteint ; pere[y] = x;
                    ordre[y] = index ; index = index + 1;
                    a_traiter.AjouterEnQueue(y) ;
                fin
            fin
            a_traiter.EnleverTete() ;
            etat[x] = examine;
        fin
    fin
fin
```

Exemple



Parcours en profondeur : principe

Le principe de choix d'un sommet sera le fait que le dernier sommet atteint sera traité, c'est-à-dire que la variable $\mathrm{a\_traiter}$ sera cette fois une **pile** (« Last In First Out »).

Lorsque l'on fait un parcours en profondeur à partir d'un sommet x, on tente d'avancer le plus loin possible dans le graphe et ce n'est que lorsque toutes les possibilités de progression sont bloquées que l'on revient (étape de « backtrack ») pour explorer un nouveau chemin ou une nouvelle chaîne.

Concrètement, on part d'un sommet puis on va voir ses adjacents puis un adjacent d'un adjacent et ainsi de suite jusqu'à être bloqué ; dans ce cas, on revient en arrière.

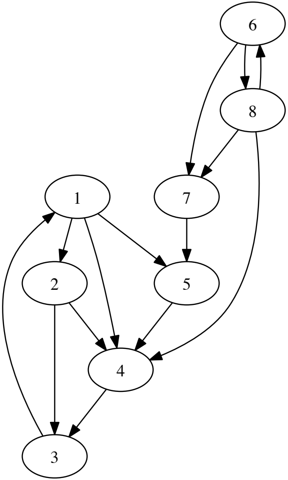
Parcours en profondeur : algorithme

```
pour chaque x ∈ S faire
    etat[x] = non_atteint;
fin
index = 1 ;

pour chaque x ∈ S faire
    si etat[x] == non_atteint alors
        pere[x] = NIL;
        ParcoursProfondeur(x);
    fin
fin
```

Procédure ParcoursProfondeur(x :sommet)

```
etat[x] = atteint;
ordre[x] = index ; index = index + 1;

pour chaque y adjacent de x faire
    si etat[y] == non_atteint alors
        pere[y] = x;
        ParcoursProfondeur(y) ;
    fin
fin
etat[x] = examine ;
```

Exemple



Plan du cours

Sommets ascendants, descendants

Définitions : descendants et ascendants

Soit $G = (S, A)$ un graphe orienté

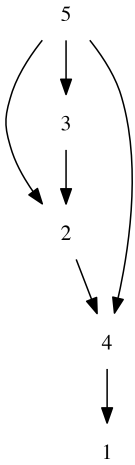
- Sommet x_k **descendant** du sommet x_i ssi il existe un chemin d'origine x_i et d'extrémité x_k

→ L'ensemble des descendants de x_i est noté :
$desc_G(x_i) = \{x_k \in S \mid \exists\, [x_i, x_k] \in G\}$

- Sommet x_k **ascendant** du sommet x_i ssi il existe un chemin d'origine x_k et d'extrémité x_i

→ L'ensemble des ascendants de x_i est noté :
$asc_G(x_i) = \{x_k \in S \mid \exists\, [x_k, x_i] \in G\}$

Exemple



- $desc_G(5) = \{1, 2, 3, 4\}$
- $asc_G(5) = \emptyset$

- $desc_G(1) = \emptyset$
- $asc_G(1) = \{2, 3, 4, 5\}$

Plan du cours

Connexité et forte connexité

Problème de la recherche de **composantes connexes** : problème d'un intérêt pratique dans de nombreuses applications où on veut savoir quels sommets sont reliés sans chercher à savoir explicitement comment

$\rightarrow$ Notion de **connexité** liée à l'existence de chaînes (graphes non orientés)
$\rightarrow$ Notion de **forte connexité** liée à l'existence de chemins (graphes orientés)

Graphes connexes

Définition : graphe connexe

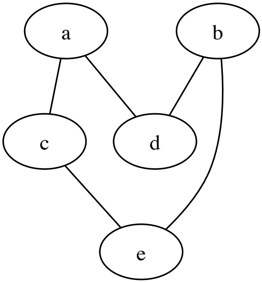
- **Graphe non orienté connexe** si, pour tout couple de sommets x et y, il existe une chaîne reliant x à y

$\rightarrow$ **Graphe orienté connexe** si son graphe non orienté associé est connexe

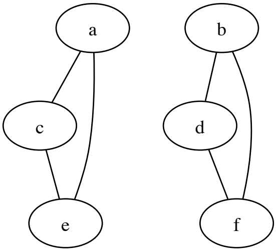
Théorème

Il existe une chaîne simple entre chaque paire de sommets (distincts) d'un graphe simple connexe

Exemples de graphes connexes et non connexes



Graphe connexe



Graphe non connexe

Composantes connexes

Définition : composante connexe

- **Composante connexe** d'un graphe $G = (S, A)$, notée C : sous-ensemble maximal de sommets tels que deux sommets quelconques du sous-ensemble sont reliés par une chaîne

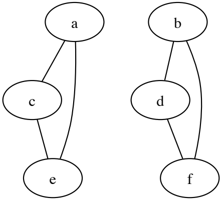
$\rightarrow$ si $x \in C$ alors

$$\forall y \in C, \text{ il existe une chaîne reliant } x \text{ à } y$$
$$\forall z \in S \setminus C, \text{ il n'existe pas de chaîne reliant } x \text{ à } z$$

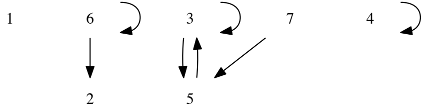
Remarques

- Ensemble des composantes connexes d'un graphe $G = (S, A)$: **partition** de l'ensemble des sommets S
- Graphe connexe $\Leftrightarrow$ une seule composante connexe

Exemples de composantes connexes



Graphe avec 2 composantes connexes : $\{a, c, e\}, \{b, d, f\}$



Graphe avec 4 composantes connexes : $\{1\}, \{2, 6\}, \{3, 5, 7\}, \{4\}$

Graphes fortement connexes

Définition : graphe fortement connexe

Graphe orienté fortement connexe si, pour tout couple de sommets x et y, il existe un chemin reliant x à y

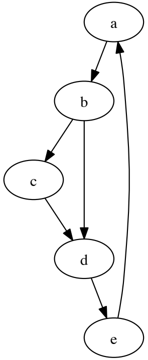
Théorème

Graphe fortement connexe ssi, pour tout couple de sommets x et y, il existe un circuit passant par x et y

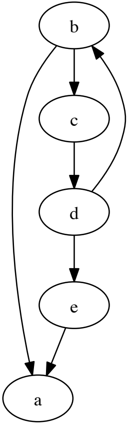
Théorème

Graphe orienté fortement connexe $\Rightarrow$ graphe connexe

Exemples de graphes fortement connexes et non fortement connexes



Graphe fortement connexe



Graphe non fortement connexe

Composantes fortement connexes

Définition : composante fortement connexe

- **Composante fortement connexe** d'un graphe $G = (S, A)$, notée C : sous-ensemble maximal de sommets tels que deux sommets quelconques du sous-ensemble sont reliés par un chemin

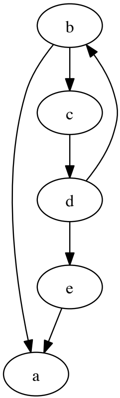
$\rightarrow$ si $x \in C$ alors

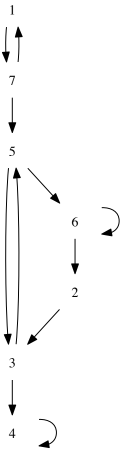
$$\forall y \in C, \text{ il existe un circuit passant par } x \text{ et } y$$
$$\forall z \in S \setminus C, \text{ il n'existe pas de circuit passant par } x \text{ et } z$$

Remarques

- Ensemble des composantes fortement connexes d'un graphe $G = (S, A)$: **partition** de l'ensemble des sommets de S
- Graphe fortement connexe $\Leftrightarrow$ une seule composante fortement connexe

Exemples de composantes fortement connexes





Graphe avec 3 composantes fortement connexes : $\{a\}, \{b, c, d\}, \{e\}$

Graphe avec 3 composantes fortement connexes : $\{1, 7\}, \{2, 3, 5, 6\}, \{4\}$

Calcul des composantes fortement connexes

Théorème : calcul d'une composante fortement connexe

$Y = (desc_G(x_i) \cap asc_G(x_i)) \cup \{x_i\}$: composante fortement connexe du graphe G

$\rightarrow$ Ce théorème donne un algorithme pour déterminer chacune des composantes fortement connexes d'un graphe

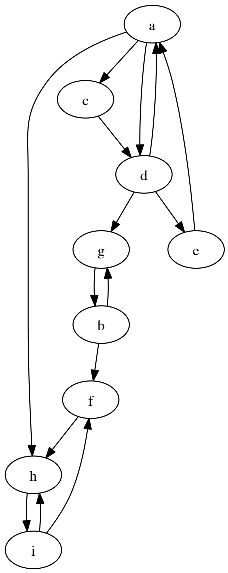
Exemple

- Une administration veut améliorer le cadre de vie de ses 9 employés qui travaillent dans la même salle

→ Objectif : répartir les employés dans plusieurs bureaux en ne séparant pas ceux qui s'échangent des documents

- Les 9 employés a, b, c, d, e, f, g, h et i échangent entre eux des documents tel qu'indiqué sur le graphe ci-contre

→ Solution : calculer les composantes fortement connexes du graphe



Exemple - suite

1. Calcul de la composante fortement connexe du sommet a
   - $desc_G(a) = \{a, b, c, d, e, f, g, h, i\}$
   - $asc_G(a) = \{a, c, d, e\}$
   - Composante fortement connexe 1 :
     $(\{a, b, c, d, e, f, g, h, i\} \cap \{a, c, d, e\}) \cup \{a\} = \{a, c, d, e\}$

2. Calcul de la composante fortement connexe du sommet b
   - $desc_G(b) = \{b, f, g, h, i\}$
   - $asc_G(b) = \{a, b, c, d, e, g\}$
   - Composante fortement connexe 2 :
     $(\{b, f, g, h, i\} \cap \{a, b, c, d, e, g\}) \cup \{b\} = \{b, g\}$

3. Calcul de la composante fortement connexe du sommet f
   - $desc_G(f) = \{f, h, i\}$
   - $asc_G(f) = \{a, b, c, d, e, f, g, h, i\}$
   - Composante fortement connexe 3 :
     $(\{f, h, i\} \cap \{a, b, c, d, e, f, g, h, i\}) \cup \{f\} = \{f, h, i\}$