esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

THEORIE DES GRAPHES

6) CONNEXITE, CONNEXITE FORTE ET GRAPHE REDUIT

6.1. Connexité

Un graphe G = [X, U] est dit connexe si, pour tout couple de sommets xj et xk, il existe une chaîne joignant xj et xk. Autrement dit, un graphe connexe est un graphe formé d'un seul bloc.

La relation R de connexité définie par :

xj R xk ⇔ ∃ (il existe) une chaîne joignant xj et xk ou bien xj = xk

est une relation d'équivalence, à savoir, une relation qui vérifie les trois propriétés qui suivent.

- Réflexive. Autrement dit, ∀ xj∈ X, xj R xj.
- Symétrique. Autrement dit, ∀ (xj, xk) ∈ X^2: si xj R xk => xk R xj.
- Transitif. Autrement dit, ∀ (xj, xk, xh) ∈ X^3: (si (xj R xk) et si (xk R xh)) => (xj R xh).

Ainsi, comme la relation R de connexité est une relation d'équivalence, alors elle induit une partition de l'ensemble des sommets X en p sous-ensembles X1, X2, …Xp, où p dénote le nombre de connexité.

Les sous-ensembles X1, X2, …Xp sont dits les composantes connexes de G.

Remarques

Un graphe est dit connexe s'il ne possède qu'une composante connexe.

Chaque sous-graphe Gi de G engendré par une composante connexe est un graphe connexe.

Exemple

Considérons le graphe G donné par la figure 1.

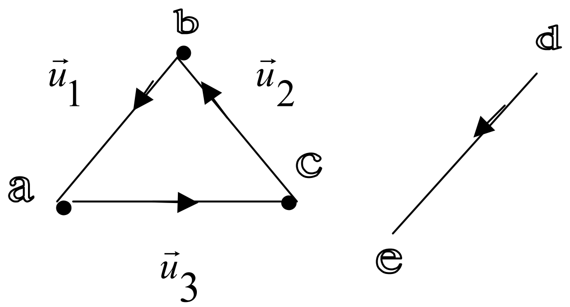


Figure 1.

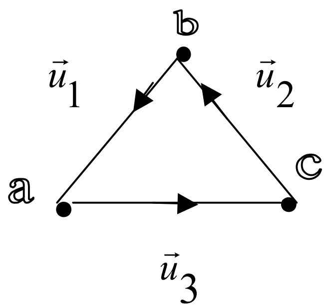
Le graphe de la figure 1 ci-dessus n'est pas connexe. Il a deux composantes connexes {a, b, c} et {d, e}.

Ainsi, le nombre de connexité p vaut 2.

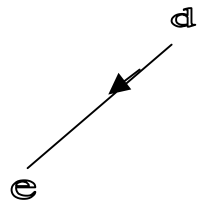
X1= {a, b, c} et X2 = {d, e} sont alors les deux composantes connexes de G.

Comme vous pouvez le constater, le sous-graphe G1 de G engendré par X1 est un graphe connexe.

De même, le sous-graphe G2 de G engendré par X2 est un graphe connexe.



Le sous-graphe G1 engendré par X1.



Le sous-graphe G2 engendré par X2.

Remarque

Le parcours en profondeur pour un graphe **non orienté** G donne les composantes connexes de G.

Les composantes connexes sont obtenues en considérant les sommets des arborescences de la forêt couvrante associée au parcours en profondeur du graphe G.

Ainsi, chaque sous-graphe engendré par les sommets d'une arborescence de la forêt couvrante est un graphe connexe.

Chaque sous-graphe Gi de G engendré par une composante connexe est un graphe connexe.

Exemple

Considérons le graphe non orienté G donné par la figure 2.

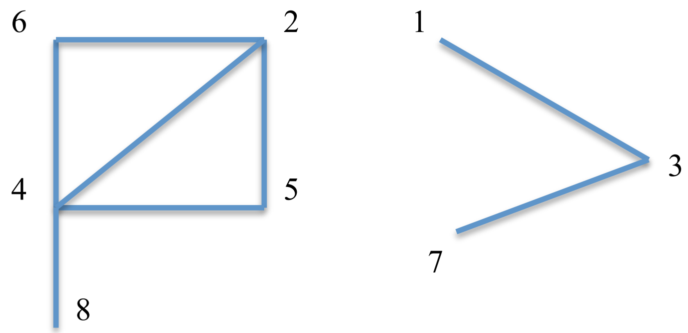
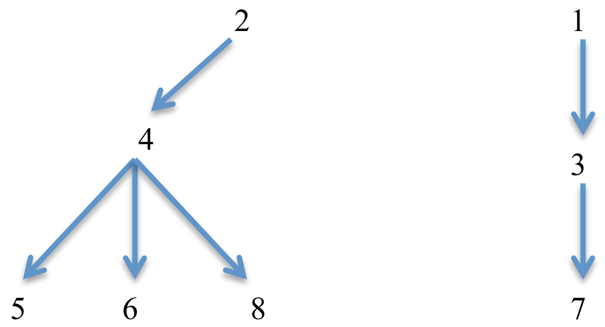


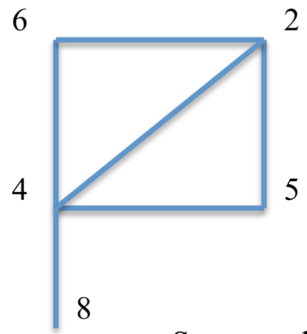
Figure 2.

La forêt couvrante associée au parcours en profondeur du graphe G de la figure 2 est celle qui suit.

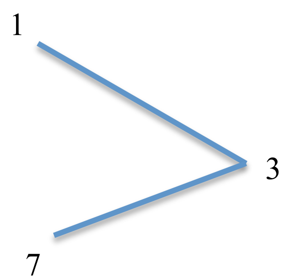


Les composantes connexes sont donc X1 = {2, 4, 5, 6, 8} et X2 = {1, 3, 7}.

Aussi, le graphe G admet deux sous-graphes connexes, le sous-graphe G1 engendré par X1 et le sous-graphe G2 engendré par X2.



Sous-graphe G1 engendré par X1



Sous-graphe G2 engendré par X2

6.2. Connexité forte

Un graphe G = [X, U] est dit fortement connexe si, pour tout couple de sommets xj et xk, il existe un chemin de xj à xk **et** un chemin de xk à xj.

La relation Ŕ de connexité forte définie par :

xj Ŕ xk ⇔ (∃ un chemin de xj à xk **et** un chemin de xk à xj) ou bien xj = xk

est une relation d'équivalence, à savoir, une relation qui vérifie les trois propriétés qui suivent.

- Réflexive. Autrement dit, ∀ xj∈ X, xj R xj.
- Symétrique. Autrement dit, ∀ (xj, xk) ∈ X^2: si xj Ŕ xk => xk Ŕ xj.
- Transitif. Autrement dit, ∀ (xj, xk, xh) ∈ X^3: (si (xj Ŕ xk) et si (xk Ŕ xh)) => (xj Ŕ xh).

Ainsi, la relation Ŕ de connexité forte induit une partition de l'ensemble des sommets X en q sous-ensembles X'1, X'2, …X'q, où q dénote le nombre de connexité forte.

Les sous-ensembles X'1, X'2, …X'q sont dits les composantes fortement connexes de G.

Remarques

Un graphe est dit fortement connexe s'il ne possède qu'une composante fortement connexe.

Un graphe fortement connexe est connexe.

Théorème : Si deux sommets x1 et x2 d'un graphe G appartiennent à un même circuit, alors ils appartiennent à une même composante fortement connexe.

Exemple

Considérons le graphe non orienté G donné par la figure 3.

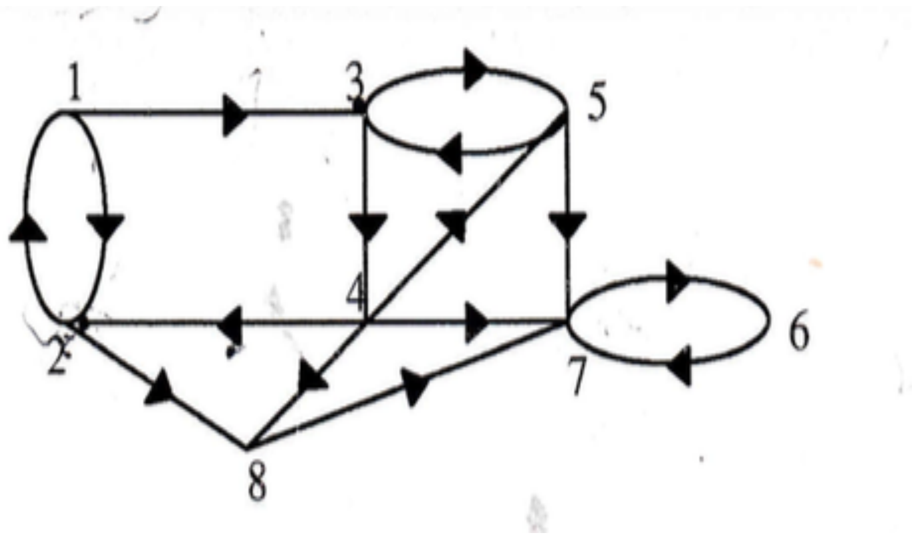


Figure 3.

Le graphe G de la figure 3 possède trois composantes fortement connexes X'1 = {1, 2, 3, 4, 5}, X'2 = {6, 7} et X'3 = {8}.

6.3. Algorithme de détermination des composantes fortement connexes

Soient G un graphe orienté et G^{-1} le graphe obtenu à partir de G en inversant le sens de tous les arcs. Le principe de l'algorithme de détermination des composantes fortement connexes est le suivant.

On effectue un parcours en profondeur de G en construisant une liste L des sommets en ordre suffixe.

Ensuite, on effectue un parcours en profondeur de G^{-1} en commençant par le dernier sommet de L, c.-à-d., par le sommet ayant le numéro le plus élevé dans la numérotation suffixe de G, et en considérant les sommets selon l'ordre décroissant de la numérotation suffixe de G.

Si, à l'issue de cet appel, tous les sommets de G ne sont pas marqués, on recommence le parcours en profondeur avec le dernier sommet non marqué de L, c.-à-d., avec le sommet qui, parmi les sommets non marqués, a le numéro le plus élevé dans la numérotation suffixe de G.

Ensuite, on continue de la même manière et ce, jusqu'à ce que tous les sommets de G soient marqués.

Les composantes fortement connexes (cfc) de G sont alors obtenues en considérant les sommets des arborescences de la forêt couvrante associée au parcours en profondeur du graphe G^{-1} tout entier.

Le schéma de l'algorithme est le suivant.

```
Procédure comp-fort-connexe (d G : GRAPHE , r comp : tableau [1 ... n] d'entiers) ;
{Cette procédure associe à chaque sommet i le numéro de la composante fortement
connexe à laquelle il appartient, soit comp[i]}
Var i, h, k : entier ;
marque : tableau [1 ... n] de booléens ;
L : tableau [1 ... n] d'entiers;
Début
        pour i = 1 à n faire
                marque [i] := faux
                comp [i] := 0
        Fin pour
        k :=0  {k est le nombre de cfc détectées jusque là}
        pour h = 1 à n faire
                L[h] := 0  {h dénote le nombre de sommets déjà visités}
                           {L[h] retourne le hème sommet visité}
        Fin pour
        h :=0
        pour i = 1 à n faire {construction de la liste L des sommets en ordre suffixe
                             de G}
                Si (marque [i] = faux) alors
                        profsuff (i, G, marque, L, h)
        Fin pour
        pour i = 1 à n faire { parcours en profondeur de G^-1}
                marque [i] := faux
        Fin pour
        pour i = n à 1 faire
                Si (marque(L[i]) = faux) alors
                        k :=k+1
                        parcours (L[i], G, marque, comp, k)
 Fin ;
```

Où,

profsuff(i, G, marque, L, h) est une procédure qui, lors du parcours en profondeur du graphe G, construit la liste L des sommets i de G en ordre suffixe. h dénote le nombre de sommets déjà visités,

parcours(L[i], G, marque, comp, k) est une procédure qui parcours en profondeur le graphe G^{-1} et remplie le tableau comp par la valeur k pour tous les sommets L[i] de l'arborescence.

```
Procédure profsuff (d x: entier, d G: GRAPHE , r M : tableau [1 ... n] de booléens, r
L : tabteau [1 ... n] d'entiers, r h : entier) ;

{...}

Var j : entier ; {j et x sont des sommets}

Début

  M[x] := vrai ; {1ère rencontre de x, on le marque} {écriture en ordre préfixe}

  Pour j := 1 à n faire {tous les j : G[x, j] = vrai, c.-à-d., tous les successeurs j de x}

       Si (G[x, j] = vrai) et (M[j] = faux) alors

                 profsuff (j, G, M, L, h) ;

  {traitement de x à la dernière rencontre} {écriture en ordre suffixe}

  h :=h+1   {on incrémente le nombre de sommets déjà visités de la valeur 1}

  L[h] :=x

  Fin pour

Fin ;
```

```
Procédure parcours (d x: entier, d G: GRAPHE, r M: tabteau [l ... n] de booléens, r
cp : tableau [l ... 11] d'entiers, d k : entier) ;

{...}

Var j : entier ; {j et x sont des sommets}

Début

  M[x] := vrai ; {1ère rencontre de x, on le marque} {écriture en ordre préfixe}

  cp[x] := k
```

Pour j := 1 à n faire {tous les j : G[j, x] = vrai, c.-à-d., tous les prédécesseurs j de x}

Si (G[j, x] = vrai) et (M[j] = faux) alors

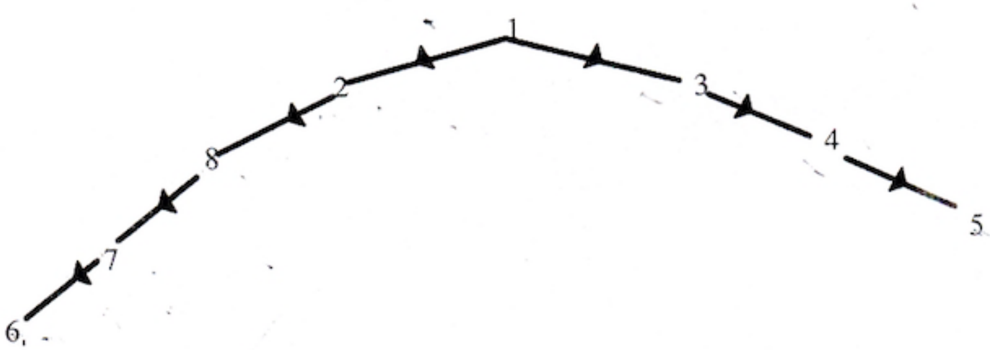
parcours (j, G, M, cp, k) ;

Fin pour

Fin ;

Exemple

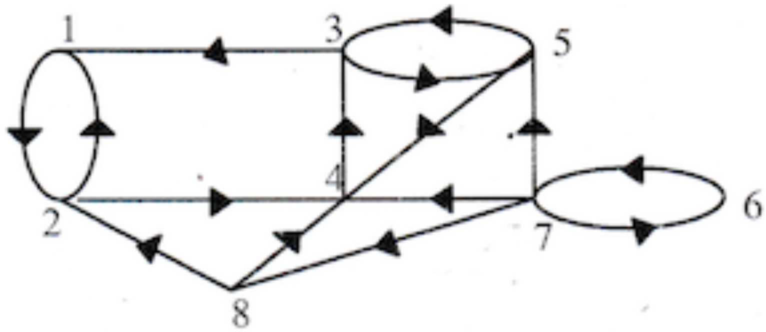
Reprenons le graphe de la figure 3. Pour ce graphe, la forêt couvrante est celle qui suit



Pour le graphe G, la liste L des sommets en ordre suffixe est alors celle donnée par le tableau suivant

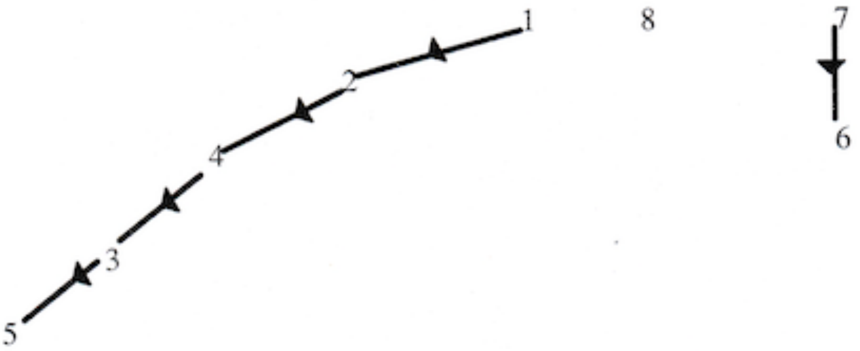
i	1	2	3	4	5	6	7	8
L[i]	6	7	8	2	5	4	3	1

Le graphe G^{-1} est celui donné par la figure ci-dessous.



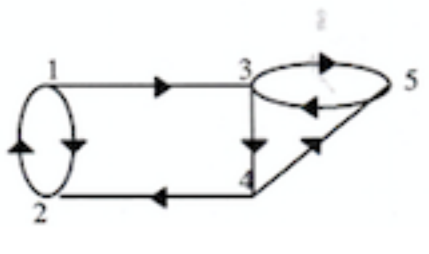
Pour effectuer le parcours en profondeur de G^{-1}, on commence par le sommet 1, car c'est le sommet de numéro suffixe le plus élevé dans L. Après avoir obtenu l'arborescence de racine 1, c.-à-d., celle qui contient les sommets 2, 4, 3 et 5, on repart avec le sommet qui, parmi les sommets non marqués, a le numéro le plus élevé dans la numérotation suffixe de G, soit avec le sommet 8. Enfin, après avoir obtenu l'arborescence réduite au sommet 8, on repart avec le sommet 7.

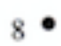
La forêt couvrante associée au parcours en profondeur de G^{-1} est celle donnée par la figure ci-dessous.



Les composantes fortement connexes du graphe G de la figure 3 sont donc X'1 = {1, 2, 3, 4, 5}, X'2 = {6, 7} et X'3 = {8}.

Aussi, le graphe G de la figure 3 admet trois sous-graphes fortement connexes, le sous-graphe engendré par X'1, le sous-graphe engendré par X'2 et le sous-graphe engendré par X'3.





Sous- graphe engendré par X'1

Sous- graphe engendré par X'2

Sous- graphe engendré par X'3

Pour le graphe G^{-1}, Le tableau comp obtenu est le suivant

i	1	2	3	4	5	6	7	8
comp[i]	1	1	1	1	1	3	3	2

6.4. Graphe réduit

On appelle graphe réduit du graphe G = [X, U] le graphe G' = [X', U'] dont l'ensemble des sommets X' est l'ensemble des composantes fortement connexes et u' = (X'1, X'2) ∈ U' <=> il existe au moins un arc u dans G dont l'extrémité initiale (resp. l'extrémité terminale) appartient à la composante fortement connexe X'1 (resp. X'2).

Exemple

Reprenons le graphe G donné par la figure 3.

Le graphe réduit de la figure 3 est celui qui suit: