Chapitre 1 : Notions de base sur les graphe

1. Introduction

Un petit dessin vaut mieux qu'un grand discours,
Napoléon.

Pour résoudre de nombreux problèmes concrets, on est amené à tracer sur le papier des petits dessins qui représentent (partiellement) le problème à résoudre. Bien souvent, ces petits dessins se composent de points et de lignes continues reliant deux à deux certains de ces points. On appellera ces petits dessins des **graphes**, les points des **sommets** et les lignes des **arcs** ou **arêtes**, selon que la relation binaire sous-jacente est orientée ou non.

2. Notions de base

Définition 1. Un **graphe G** est défini par : un ensemble **S** appelé **ensemble des sommets** {x1,x2,x3,...} un sous-ensemble **A** du produit cartésien S×S appelé **ensemble d'arcs** (ou arêtes) A = {(x,y) ∈ S ×S / le sommet x est en relation avec le sommet y}. On note **G = (S,A).**

Remarques.

- Si S est un ensemble fini, alors le graphe **G** est dit **fini**.
- Un sommet est **isolé** lorsqu'aucune arête ne le relie aux autres sommets

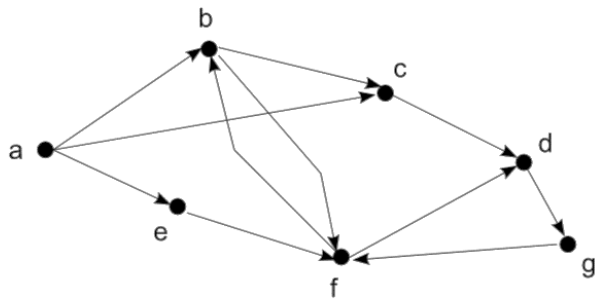
Définition 2. On appelle **ordre** d'un graphe fini le nombre de ses sommets.
On note : **n=|S|**

Définition 3. On appelle **taille** d'un graphe fini le nombre de ses arêtes.
On note : **m=|A|**

Notation 1. Un **arc** u = (x,y) est noté **x → y** , x est appelée **origine** ou extrémité initiale y est appelée **destination** ou extrémité finale. On parle alors de **graphe orienté (GO)**.

Notation 2. Lorsque le sens de parcours n'a pas de signification on dit que le graphe est **non orienté (GNO)**. u = (x,y) est alors appelée une **arête**, notée **x – y**

Exemple.



G=(S,A)
S={a,b,c,d,e,f,g}
A={(a,b),(a,c)},(a,e),(b,c),(b,f),
(c,d),(d,g),(g,f),(f,b),(f,d),(e,f)

Définition 4. Successeur et prédécesseur d'un sommet

- On note $\boldsymbol{\Gamma^+}$ l'application :

  $\Gamma^+ : S \rightarrow S$

  $x \rightarrow \Gamma^+(x) = \{y \in S \,/(x,y) \in A\}$

  où $\Gamma^+(x)$ représente l'ensemble des successeurs de x.

- On note $\boldsymbol{\Gamma^-}$ l'application :

  $\Gamma^- : S \rightarrow S$

  $x \rightarrow \Gamma-(x) = \{y \in S \,/(y,x) \in A\}$

  où $\Gamma^-(x)$ représente l'ensemble des prédécesseurs de x.

Remarque Un graphe G = (S,A) est entièrement déterminé par chacun des couples (S,Γ^+) et (S,Γ^-).

Définition 5. Incidence et adjacence

Soit u = (x,y) une arête.

- On dit que u est **incidente** à x (et à y). Soit u = (x,y) un arc. On dit que u est incident à x **vers l'extérieur** et incident à y **vers l'intérieur**.
- On dit que les sommets x et y sont **adjacents** à l'arc ou arête u.

Définition 6. Degré d'un sommet et degré d'un graphe

- On appelle demi degré intérieur d'un sommet x (***degré sortant***), et on note **d−(x)**, le nombre d'arcs incidents à x vers l'intérieur.
- On appelle demi degré extérieur d'un sommet x (***degré entrant***), et on note **d+(x)**, le nombre d'arcs incidents à x vers l'extérieur.
- Le **degré d'un sommet** x est égal à d(x) = d−(x) + d+(x). (Pour un graphe non orienté, le degré d'un sommet est le nombre d'arêtes qui lui sont incidentes.)
- Le **degré d'un graphe** est le degré maximum de tous ses sommets.

Propriété. Soit G un graphe, S l'ensemble de ses sommets et A le nombre d'arêtes de G. Alors la somme des degrés vérifie :

$$\sum_{A \in \mathcal{S}} d(A) = 2Card(\mathcal{A})$$

3. Chemin, chaîne, circuit et cycle

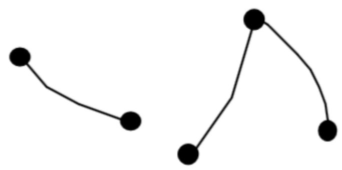
- Un **chemin** est une suite d'arcs dont l'extrémité finale de chacun est l'extrémité initiale du suivant (sauf pour le dernier). Exemple : a → b → c → d → g
- Si le graphe n'est pas orienté, on parle alors de **chaîne**.
- Un chemin **simple** est un chemin qui ne passe pas plus d'une fois par le même arc.
- Un chemin **élémentaire** est un chemin qui ne passe pas plus d'une fois par le même sommet.
- La **longueur** d'un chemin est le nombre d'arcs qui constituent le chemin.
- Un **circuit** est un chemin qui se ferme sur lui-même (son origine et son extrémité sont confondues).
- Si le graphe n'est pas orienté, on parle alors de **cycle**.
- Un chemin (resp. circuit) **hamiltonien** est un chemin (resp. circuit) qui passe par tous les sommets du graphe une fois et une seule.
- Un chemin (resp. circuit) **eulérien** est un chemin (resp. circuit) qui passe par tous les arcs du graphe une fois et une seule.
- On appelle **distance** entre deux sommets la longueur de la plus petite chaîne les reliant.
- On appelle **diamètre** d'un graphe la plus longue des distances entre deux sommets.
- Une chaîne dont les sommets de départ et de fin sont les mêmes est appelée chaîne **fermée**.

4. Connexité et forte connexité

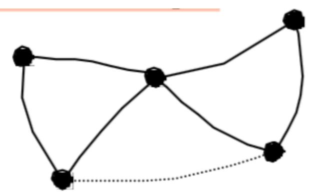
Définitions :

- Un graphe est dit **connexe** s'il existe une chaîne entre toutes paires de sommets du graphe. (Si le graphe est orienté, cette notion s'évalue sur le graphe non orienté qui s'en déduit naturellement.)

- Si le graphe n'est pas connexe on peut identifier plusieurs sous-graphes connexes maximaux (au sens de l'inclusion) appelés **composantes connexes**. Chaque composante connexe est une classe d'équivalence pour la relation xRy ⇔ "il existe une chaîne entre x et y".
- Un graphe orienté est dit **fortement connexe** si pour toute paire de sommets (x,y) il existe un chemin de x vers y ET de y vers x. Si le graphe n'est pas fortement connexe on peut identifier plusieurs sous-graphes fortement connexes maximaux appelés composantes fortement connexes. Chaque composante fortement connexe est une classe d'équivalence pour la relation xRy ⇔ "il existe un chemin de x vers y ET de y vers x".
- Un graphe est « ***k-connexe*** » si on peut le déconnecter en retirant k sommets et qu'on ne peut pas en en supprimant k – 1 ; son « degré de connexité » est alors k.



Graphe à deux composantes connexes



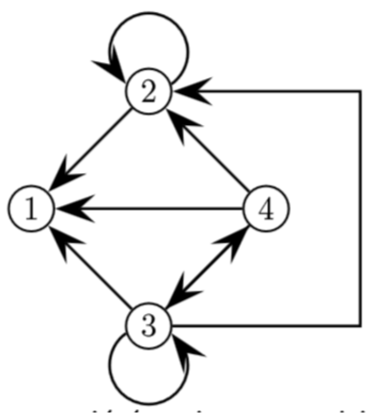
Graphe 1-connexe (traits pleins)
Avec les pointillés, il devient 2-connexe.

5. Matrice d'adjacence et listes d'adjacence :

Définition 1. (matrice d'adjacence) Soient $E = \{x_1;x_2;\ldots;x_p\}$, $F = \{y_1;y_2;\ldots;y_n\}$ et $\mathcal{R} = (E,F,U)$ une relation alors on appelle matrice d'adjacence de $\mathcal{R}$ la matrice booléenne $M_{\mathcal{R}}$ telle que $M_{\mathcal{R}} = (m_{i,j})$ avec $m_{i,j} = \begin{cases} 1 & \text{si } (x_i, y_j) \in U \\ 0 & \text{sinon} \end{cases}$

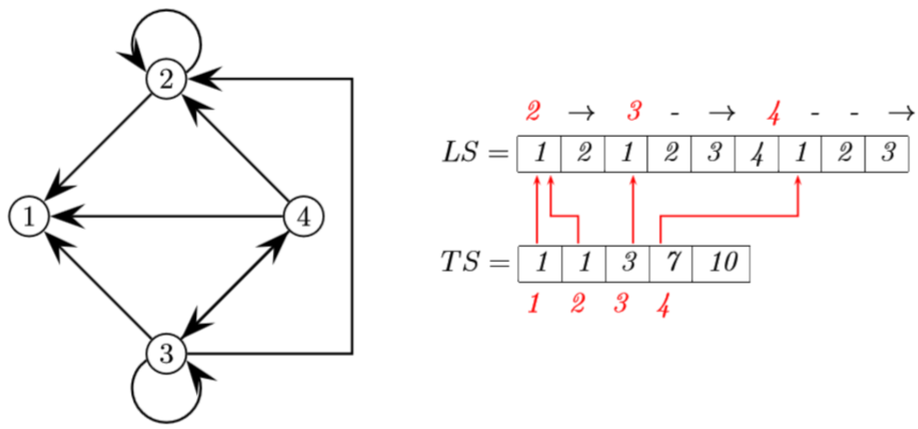
Exemple.

$$M_{\mathcal{R}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



Définition 2. (listes d'adjacence) Soit G = (S,A) un graphe orienté d'ordre n et de taille m dont les sommets $x_1;x_2;\dots x_n$ sont ordonnés. Le graphe G peut être représenté par des listes d'adjacence (LS,TS) qui sont définies par :

- LS = liste de longueur m appelée ≪ liste des successeurs ≫, elle contient les successeurs du sommet 1 (rangé dans l'ordre croissant) puis du sommet 2 …et si un sommet n'a pas de successeur on passe au sommet suivant.
- TS = liste de longueur n+1 appelée ≪ liste des têtes successeurs ≫ qui indique la position du premier successeur de chaque sommet dans LS la liste TS est définie comme suit :
- TS(1) = 1
- pour x ∈ S — si x à des successeurs alors TS(x) = numéro de la case de LS du premier successeur de x — sinon TS(x) = T



Les listes d'adjacences occupent une place mémoire de taille n + m + 1 c'est le minimum d'informations pour représenter un graphe comparé à la matrice d'adjacence occupe une place n^2 et la liste des arcs occupe une place 2m.

Définition3. Fermeture transitive

On appelle ***fermeture transitive d'un sommet x*** l'ensemble des sommets y (y compris x) reliés par un chemin à x.

On appelle ***fermeture transitive d'un graphe G*** le graphe Ĝ construit de la manière suivante :

- Les sommets de Ĝ sont ceux de G
- Chaque sommet de Ĝ porte une boucle
- Il y a dans Ĝ un arc de x vers y ssi il existe un chemin entre x et y dans G

Remarque La matrice M^ associée à Ĝ est donnée par : $M\hat{} = M\hat{}^{[k]} = M^{(0)} + M^{(1)} + \ldots + M^{(k)}$ (on s'arrête quand $M\hat{}^{[k]} = M\hat{}^{[k+1]}$, $M^{(0)} = Id$)

Définition4. Fermeture transitive et composante fortement connexe

Proposition Deux sommets x_i et x_j appartiennent à la même composante fortement connexe si et seulement si les lignes (ou les colonnes) i et j de M^ sont identiques.

Remarque Un graphe G est fortement connexe si et seulement si tous les éléments de la matrice M^ sont égaux à 1.

6. Types de graphe

Def1. (Graphe simple) Un graphe est dit simple si : il n'y a pas de boucle (u = (x,x)) il n'y a pas plus d'un arc (ou pas plus d'une arête) pour relier deux sommets.

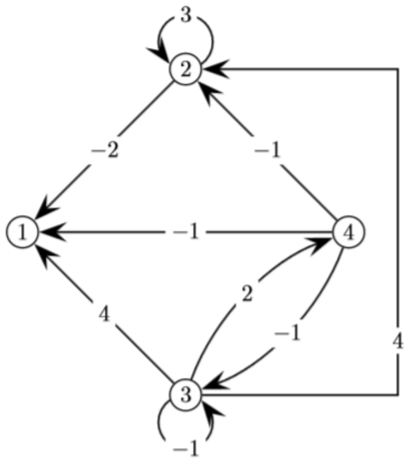
Un graphe simple G = (S,A) est dit **nul** si S et A sont vides.

un graphe simple G est dit **trivial** si il admet un seul sommet et aucune arête, on le note par **T1**. On appelle graphe **trivial d'ordre n** le graphe admettant **n** sommets et aucune arête, on le note par **On** ou **Tn**

Déf2. (graphe valué ou pondéré) Un graphe valué *G = (S,A,v)* est un graphe (S,A) (orienté ou non-orienté) muni d'une application $v : A \to \mathbb{R}$ Le graphe peut être représenté par la matrice des valuations : $W \in \mathcal{M}_n(\mathbb{R})$ telleque

$$W_{i,j} = \begin{cases} \infty & \text{si } (xi, yj) \notin A \\ v\big((xi, yj)\big) & \text{si } (xi, yj) \in A \end{cases}$$

Pour un graphe valué on ajoutera les valuations sur chaque arc ou arête.



graphe valué orienté

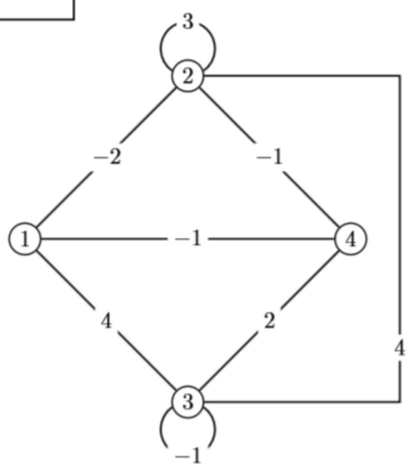
$$W = \begin{pmatrix} \infty & \infty & \infty & \infty \\ -2 & 3 & \infty & \infty \\ 4 & 4 & -1 & 2 \\ -1 & -1 & -1 & \infty \end{pmatrix}$$

$$et\ M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

graphe valué non-orienté

$$W = \begin{pmatrix} \infty & -2 & 3 & 4 \\ -2 & -1 & 4 & -1 \\ 3 & 4 & -1 & 2 \\ 4 & -1 & 2 & \infty \end{pmatrix}$$

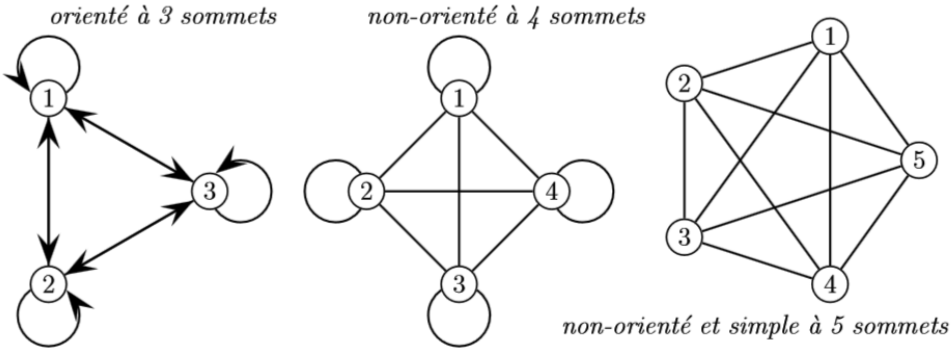
$$et\ M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



On rencontrera dans les applications de la théorie des graphes plusieurs types de valuations associées aux arcs d'un graphe : poids, longueur, coût, capacité,...

Déf3. (graphe complet) On appelle « graphe complet à *n* sommets », souvent noté K_n, le graphe d'ordre *n* ayant le plus d'arcs/arêtes possibles.

Exemple.



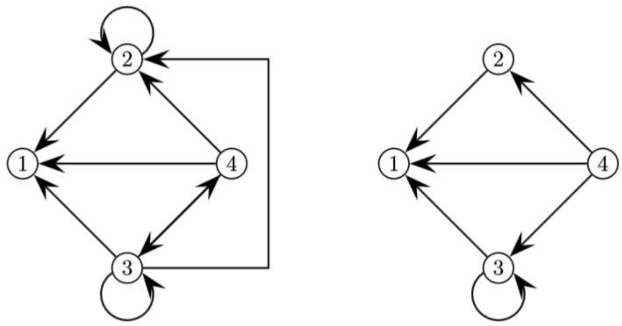
- On appelle « ***tournoi*** » un graphe complet orienté. Une flèche peut alors représenter la relation « a gagné sur ».

Quel est le nombre d'arêtes dans le cas d'un graphe complet K_n ?

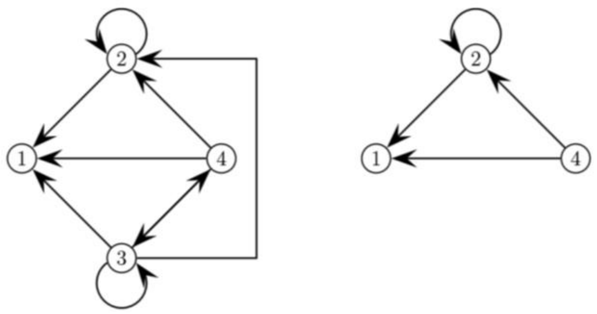
Déf4. (sous-graphe et graphe partiel) Soit G = (S,A) un graphe (orienté ou pas) alors

- un graphe partiel de G est un graphe G′ ayant pour sommets tous les sommets de G et pour arcs/arêtes seulement un sous-ensemble de A, ce qui s'écrit : $G' = (S,A')$ avec $A' \subset A$
- un sous-graphe de G est un graphe G′ ayant pour sommets un sous ensemble S′ des sommets de G et en ne conservant que les arcs/arêtes joignant les sommets de S′ ce qui s'écrit : $G' = (S',A')$ avec $S' \subset S$ et $A' = \{(x,y) \in A | x \in S' \text{ et } y \in S'\}$
- Un sous-graphe H de G est dit graphe induit de G si les sommets de H qui sont adjacents dans G ils le sont dans H.

Exemple. graphe partiel de G induit par $A' = A \setminus \{(2,2);(3,2);(4,3)\}$



sous-graphe de G induit par $S' = \{1;2;4\}$



Deux exemples importants de sous-graphe et de graphe partiel sont les cliques et les stables :

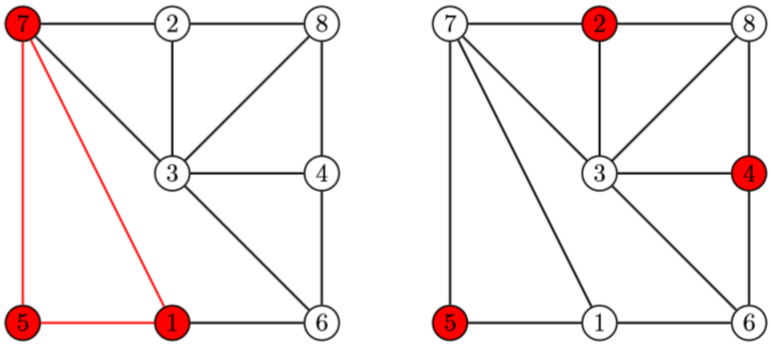
Déf5. (clique et stable) Soit $G = (S,A)$ un graphe (orienté ou pas) alors

• une clique est un sous-graphe complet de G

• un stable est un sous-graphe de G sans arcs/arêtes

La recherche du plus grand stable ou de la plus grande clique d'un graphe est un problème très important en théorie des graphes

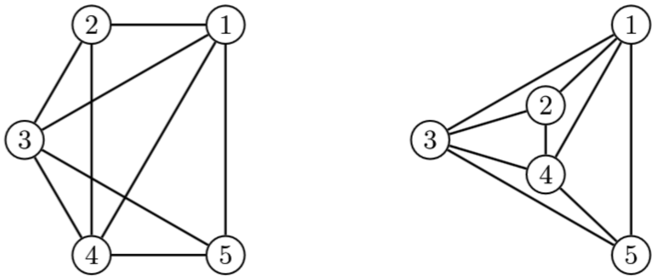
Exemple. Trouver le plus grand stable et la plus grande clique d'un graphe



Dans la graphe G l'ensemble de sommets {1;5;7} induit une clique maximale alors que {2;4;5} induit un stable maximal (il y en a d'autres).

Déf6. (graphe planaire) Un graphe G = (S,A) est dit planaire s'il existe un graphe équivalent à ce graphe où aucun arc/arêtes n'en coupe d'autre.

Exemple. Rendre le graphe suivant planaire il faut déplacer les sommets 2 et 4

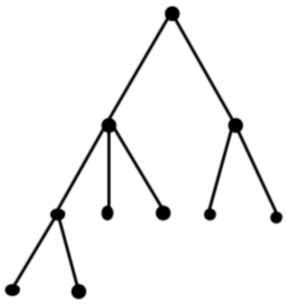


Est-ce encore possible si on ajoute l'arête {2;5} au graphe?

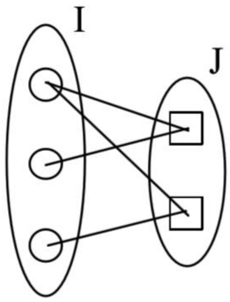
Un graphe ayant pour nombre chromatique γ(G) = 5 ne peut donc pas être planaire. Mais attention, la réciproque de ce théorème est fausse : un graphe avec γ(G) = 4 n'est pas forcément planaire!

Autres graphes :

Arbre. Ce sont des GNO connexes sans cycle (acyclique). Ou encore des GNO connexes dont le nombre de sommets est égal au nombre d'arêtes plus 1.



Graphe biparti. Un ensemble I de sommets d'un graphe est « indépendant » si aucun élément de I n'est connecté à un autre élément de I. Un graphe « biparti » est un graphe qui représente des relations entre deux ensembles indépendants I et J. On définit de même un graphe multiparti.



Hypergraphe. Quand les relations ne sont pas toutes binaires, on parle d'hypergraphe; on peut alors le transformer en graphe biparti.

Graphe régulier Un graphe est dit régulier si tous les sommets ont le même degré.
Si le degré commun est k, alors on dit que le graphe est k-régulier.
Si r = 3 le graphe est dit cubique. Si G est régulier d'ordre n et deg(G) = n−1 alors le graphe G est le graphe complet d'ordre n.

7. Etude de la connexité :

Trouver les composantes connexes d'un graphe est assez facile visuellement, mais trouver les composantes fortement connexes est plus difficile. Les algorithmes suivants permettent de construire les composantes connexes et fortement connexes dans G=(S,A).

a. Construction de la *composante simplement connexe* de x

i) Marquer le sommet x *

ii) Marquer tout sommet adjacent à un sommet déjà marqué.

Poursuivre (ii) jusqu'à ce que l'on ne puisse plus marquer aucun sommet. Alors les sommets marqués sont ceux de la composante connexe de x.

b. Construction de la *composante fortement connexe* de x

i) Marquer + et – le sommet x

ii) Marquer + tout successeur (non encore marqué +) d'un sommet déjà marqué +. Marquer - tout prédécesseur (non encore marqué -) d'un sommet déjà marqué -.

Lorsque plus aucun sommet ne peut être marqué ni + ni -, les sommets marqués à la fois + et – sont ceux de la composante fortement connexe de x.

Def1. Graphe k-connexe

Un graphe G est dit **k-connexe** si et seulement si G est connexe d'ordre $n > k + 1$ et il n'existe pas d'ensemble $X \subset S$ de cardinal $k - 1$ tel que le sous graphe engendré par $S - X$ n'est pas connexe. En d'autres termes, en supprimant moins de k sommets, le graphe sera toujours connexe.

Def2. Graphe réduit

A tout graphe orienté G = (S,A) on associe le graphe simple GR = (SR,AR) appelé graphe réduit de G défini comme suit : SR = {A chaque C.F.C C_i de G correspond un sommet C_i}

$AR = \{(C_i,C_j)/i \neq j \text{ et } \exists x \in C_i \text{ et } \exists y \in C_j \text{ et } (x,y) \in A\}$

Remarque. Un graphe fortement connexe possède une seule C.F.C. Le graphe réduit d'un graphe ne possède pas de circuits.

Exemple. Voir cours

8. Décomposition d'un graphe en niveaux :

Une décomposition en niveaux (***tri topologique***) d'un graphe orienté sans circuit est réalisée en organisant les sommets d'un graphe en k niveaux (couches ou layer) de la manière suivante :

• Les sommets sans prédécesseurs sont de niveau 0

• tout sommet x a un niveau supérieur aux niveaux de ses prédécesseurs :

niveau(x) = max $y \in \Gamma^{-}(x)$

niveau(y) + 1

On peut ensuite re-dessiner le graphe G en disposant les sommets de gauche à droite dans l'ordre croissant des niveaux.

Pour décomposer en niveau un graphe G on utilisera l'algorithme suivant :

```
procedure DecompositionNiveaux
    N ← 1
    S` ← S
    Tant que S` = ∅ do
        Pour tout sommet x de S` sans prédécesseur faire
        Niv(x) ← N
        Fin pour
        enlever de S` tous les sommets de niveau N
        N ←N + 1
    Fin tant que
fin procedure
```

Exemple. Voir cours et TD

9. Parcours de graphes orientés :

Le parcours d'un graphe est un problème type de théorie des graphes auquel peuvent se ramener de nombreux autres problèmes d'algorithmique.

Déf1. (parcours d'un graphe) Soit G = (S,A) un graphe et x ∊ S un sommet, un parcours du graphe G à partir de x est une visite de chaque sommet accessible depuis x. Le résultat d'un parcours est un ensemble de chemins partants de x allant vers les sommets accessibles depuis x. Un parcours peut être représenté par sous-graphe de G qui est un arbre de racine x (ou une forêt si certains sommets de G ne sont pas accessibles depuis x).

Déf2. (parcours en largeur) Soit G = (S,A) un graphe et x ∊ S un sommet, un parcours en largeur du graphe G à partir de x est un parcours dans lequel un sommet y est marqué avant le début de traitement de ses successeurs. Si on veut récupérer la liste des prédécesseurs P qui permet de retrouver l'arbre de parcours en largeur depuis le sommet x on utilisera l'algorithme suivant :

```
fonction P = parcours_largeur(G, x)
   L = file des sommets à traiter
   P = liste des prédécesseurs de l'arbre de parcours
   marquer le sommet x et le mettre dans L
   tant que L ≠ ∅ faire
            sortir le 1er sommet y de L
            V = successeurs non traités de y
            pour tout z ∈ V faire
                      marquer z; P(z) = y
                      mettre z à la fin de L
              fin faire
            fin faire
```

Le parcours en largeur est très utile pour trouver les distances depuis un sommet donné dans un graphe.

Théorème1. (parcours en largeur et distances) Soit G = (S,A) un graphe et x un sommet, alors les niveaux des sommets dans le parcours en largeur depuis le sommet x sont exactement les distances de x à ces sommets.

Déf3. (parcours en profondeur) Soit G = (S,A) un graphe et x ∈ S un sommet, un parcours en profondeur du graphe G à partir de x est un parcours dans lequel un sommet y n'est marqué qu'après le début du traitement de ses successeurs.

Comme pour le parcours en largeur on peut écrire précisément l'algorithme permettant de récupérer l'arbre de parcours en profondeur :

```
fonction P = parcours_profondeur(G, x)
   L = [x] (Pile des sommets à traiter)
   P = liste des prédécesseurs de l'arbre de parcours
   tant que L ≠ ∅ faire
            sortir le 1er sommet y de L
            V = successeurs non traités de y
            pour tout z ∈ V faire
                      P(z) = y
                      mettre z au début de L
              fin faire
              marquer le sommet y
            fin faire
```

Cet algorithme admet aussi une formulation récursive plus simple à programmer :

fonction $P =$ parcours_profondeur(G, x)
marquer x (début du traitement de x)
$V =$ successeurs non traités de x
pour tout $y \in V$ **faire**
$P(y) = x$
$P =$ parcours_profondeur(G, y)
fin faire
fin du traitement de x

Le parcours en profondeur ne permet pas de calculer les distances depuis un sommet! Au contraire le parcours en profondeur essaye de construire les chemins les plus long possibles depuis un sommet donné.

Le parcours en profondeur sert, entre autre, à classer les arcs en différentes catégories.

Déf4. (classification des arcs) le parcours en profondeur d'un graphe G depuis le sommet x permet de définir quatre types d'arc :

- ***arcs couvrants*** : les arcs retenus pour le parcours en profondeur
- ***arcs directs*** : les arcs n'appartenant pas au parcours en profondeur mais reliant un sommet à un descendant
- ***arcs rétrograde*** : les arcs n'appartenant pas au parcours en profondeur mais reliant un sommet à un ascendant (ou à lui-même)
- ***arcs traversiers*** : les arcs n'appartenant pas au parcours en profondeur mais reliant deux branches distinctes de l'arbre

Cette classification permet de détecter des circuits dans un graphe.

Théorème2. (détection des circuits) Dans un graphe G = (S,A) et x un sommet. Si dans le parcours en profondeur de G à partir de x il existe un arc rétrographe alors il existe au moins un circuit dans le graphe G.