

LES GRAPHES, UN OUTIL DE MODELISATION

Dans ce chapitre nous introduisons un concept, celui de graphe, qui permet de modéliser et de résoudre de nombreux problèmes.

L'utilisation des graphes est courante en tant qu'outil de représentation : par exemple, le plan schématisé de rues, un arbre généalogique, la représentation d'un réseau informatique sont des exemples de graphes et vous en avez utilisé comme Monsieur Jourdain faisait de la prose... sans le savoir.

Mais, au-delà de la représentation de données, les graphes servent aussi et surtout pour proposer des solutions à certains problèmes.

Bien que l'on puisse dater le premier problème utilisant les graphes à Euler (1736) lors de la résolution du problème des 7 ponts de Königsberg (voir plus loin), le nom de graphes n'est apparu qu'en 1822 sous la plume d'un anglais J.-J. Sylvester.

I Définitions

Définition d'un graphe orienté

Un **graphe orienté** $G=(X,U)$ est défini par la donnée de deux ensembles :

- un ensemble X dont les éléments sont appelés "**sommets**" ou "**nœuds**"
- un ensemble U dont les éléments sont des couples de sommets appelés "**arcs**".

Si $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ est l'ensemble des sommets un arc $u \in U$ s'écrit : $u = (x_i, x_j)$ où x_i est l'**origine** (ou extrémité initiale) et x_j l'**extrémité** (ou extrémité terminale) de l'arc.

Si $x_i = x_j$, l'arc est une "**boucle**".

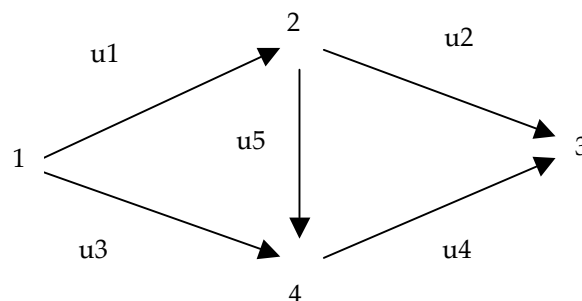
Le graphe est **fini** si $n = |X|$ et $m = |U|$ sont finis.

n , nombre de sommets, est l'**ordre** du graphe. Nous ne considérerons que des graphes finis.

Représentation sagittale d'un graphe orienté fini

Chaque sommet est représenté par un point du plan, chaque arc par une... flèche reliant le sommet origine au sommet extrémité.

Exemple 1



Graphe de 4 sommets et 5 arcs : $u_1 = (1,2)$ $u_2 = (2,3)$

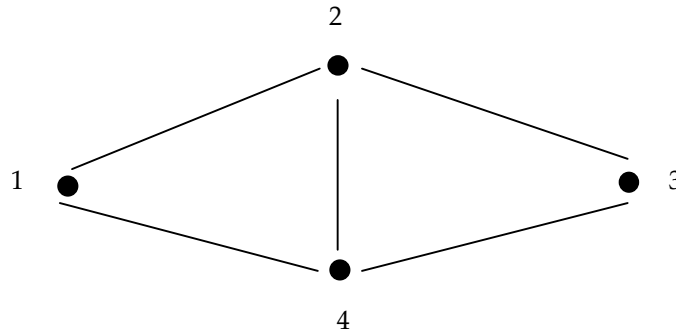
Définition d'un graphe non orienté

Dans certains problèmes, le sens des arcs n'a pas d'importance. Par exemple, si le problème associé au graphe concerne une personne se déplaçant à pied dans une ville, il n'est pas nécessaire de se préoccuper de l'orientation des rues qui correspond aux sens interdits. On se contente alors d'utiliser la notion de graphe non orienté. La notion d'arc est remplacée par celle d'arête.

Un graphe non orienté $G = (X, V)$ est défini par la donnée de deux ensembles :

- un ensemble de **sommets** X
- un ensemble V **d'arêtes** : une arête étant une paire non ordonnée de sommets.

Dans la représentation d'un graphe non orienté, chaque arête est représentée par un simple trait entre ses deux extrémités.



Chemin dans un graphe orienté

Les graphes sont souvent utilisés pour modéliser des problèmes associés à des parcours ou à des successions d'actions.

Pour cela, on introduit la notion de **chemin**.

Chemin, circuit, racine

On appelle "**chemin d'origine x et d'extrémité y** " une séquence d'arcs telle que :

- le premier arc a pour origine x
- l'origine de tous les autres coïncide avec l'extrémité de l'arc qui le précède dans la séquence
- le dernier arc a pour extrémité y .

Si les arcs de la séquence sont tous distincts le chemin est "**simple**".

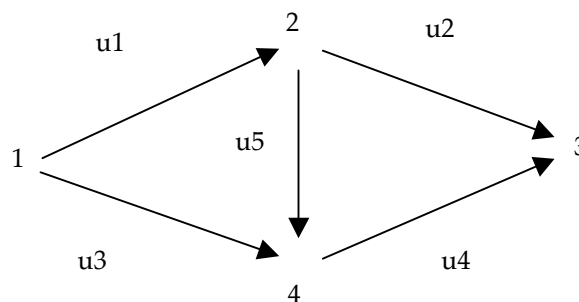
$y \in X$ est un "**descendant**" de $x \in X$ s'il existe un chemin d'origine x et d'extrémité y , x est alors un "**ascendant**" de y .

Un sommet x est **racine** si quel que soit le sommet y du graphe il existe un chemin de x à y .

Un "**circuit**" est un chemin simple dont les extrémités coïncident.

Un chemin est "**élémentaire**" s'il ne contient pas de circuit.

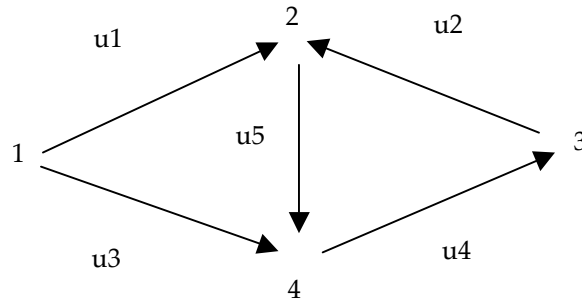
Exemple 2



La séquence u_1, u_5, u_4 représente un chemin (simple et élémentaire) de 1 à 3.

Ce chemin peut également être décrit par la suite des sommets 1, 2, 4, 3.

Exemple 3



Dans le graphe ci-dessus, la séquence d'arcs u_2, u_5, u_4 représente un circuit.

La séquence u_1, u_5, u_4, u_2 représente un chemin d'origine 1 et d'extrémité 2 qui n'est pas élémentaire. Ce chemin peut être décrit par la suite des sommets 1, 2, 4, 3, 2. On peut en extraire le chemin élémentaire de 1 à 2 constitué de l'arc u_1 .

Chaîne, cycle

On appelle "**Chaîne reliant x et y**" une séquence d'arcs telle que :

- le premier arc est adjacent à x par une de ses extrémités et au deuxième arc de la séquence par son autre extrémité.
- le dernier arc est adjacent à y par une de ses extrémités et à l'avant-dernier arc par son autre extrémité
- chaque arc intermédiaire de la séquence est adjacent au précédent par une de ses extrémités et au suivant par l'autre.

Si les arcs de la séquence sont tous distincts la chaîne est "**simple**".

Un "**cycle**" est une chaîne simple dont les extrémités coïncident.

Une chaîne est "**élémentaire**" si elle ne contient pas de cycle.

Exemple 4

Dans le graphe de l'exemple 3, la séquence u_1, u_2 constitue une chaîne simple de 1 à 3 mais n'est pas un chemin.

La séquence u_1, u_5, u_3 est un cycle mais n'est pas un circuit.

II Exemples de problèmes modélisables à partir d'un graphe

A) Exemple 1 : Le passeur, le loup, la chèvre et le chou

Cette histoire bien connue serait due à Alcuin d'York (735 - 804).

Sur la rive d'un fleuve se trouvent un loup, une chèvre, un chou et... un passeur.

Il s'agit de faire passer tout ce petit monde sur l'autre rive de telle manière que ne restent jamais sur une rive, seuls sans le passeur, la chèvre et le chou, la chèvre et le loup, sachant qu'en outre le passeur ne peut mettre qu'un des trois protagonistes dans sa barque.

Ce problème peut se modéliser au moyen d'un graphe.

On représente le passeur par P, le loup par L, la chèvre par C, et le chou par X.

Les **sommets** du graphe sont de 2 types :

- ceux qui correspondent à une situation possible sur la rive de départ avec le passeur

Tous les états sont possibles sauf PL et PX : PL (passeur + loup) n'est pas possible car sur l'autre rive il y aurait la chèvre et le chou, PX (passeur + chou) laisse seuls sur l'autre rive le loup et la chèvre.

- ceux qui correspondent à une situation possible sur la rive de départ sans le passeur

Les états possibles sur la rive de départ, lorsque le passeur n'est pas là, doivent exclure le loup et la chèvre (état LC) ou la chèvre et le chou (état CX).

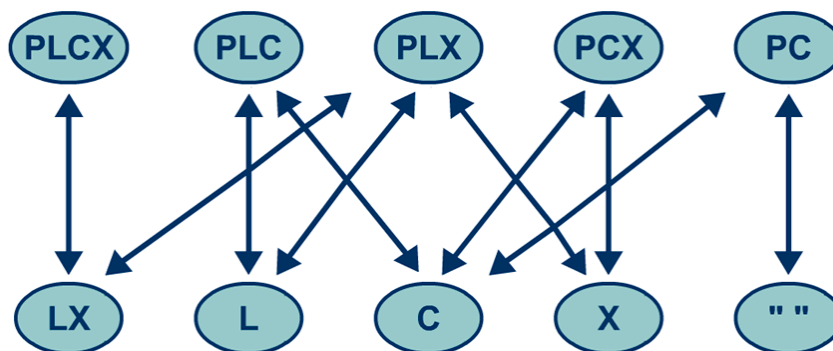
Il reste donc : LX (loup + chou) et les situations avec un seul des trois et bien sûr la situation sans personne, situation que l'on veut atteindre.

Les **arcs** de ce graphe correspondent au passage d'un état à un autre : lorsque le passeur quitte la rive dans un sens et lorsqu'il y revient dans l'autre.

Lorsqu'il part, il part seul ou emmène avec lui un de ses acolytes, l'état suivant est donc obtenu en retirant au plus un des protagonistes.

Quand il revient, il revient seul ou avec un des protagonistes.

D'où le graphe :



Le problème consiste, en partant de l'état initial PLCX, à arriver, par une succession de passages, à l'état où il n'y a plus personne sur la rive de départ.

Il s'agit donc de **trouver dans ce graphe un chemin** du sommet PLCX au sommet "".

Si on souhaite effectuer le moins possible de traversées, il s'agit de trouver un chemin du sommet PLCX au sommet "" comportant le plus petit nombre d'arcs possibles.

Une solution est PLCX puis LX, PLX, X, PCX, C, PC et enfin le vide sur la berge de départ qui se traduit par :

- le passeur part avec la chèvre (il reste le loup et le chou, état LX)
- il revient seul : état PLX
- il repart avec le loup : état X
- il revient avec la chèvre : état PCX
- il repart avec le chou : état C
- il revient seul : état PC
- il repart avec la chèvre.

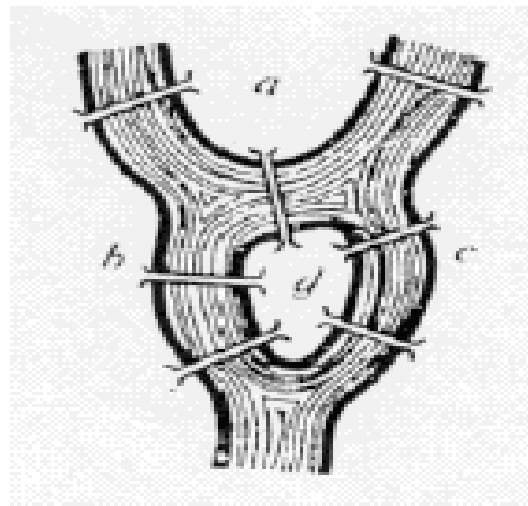
Une autre solution est : PCLX, LX, PLX, L, PLC, C, PC, "".

Elle comporte le même nombre de passages.

Un problème analogue est celui des missionnaires et des cannibales : trois missionnaires et trois cannibales veulent traverser un fleuve avec un bateau qui ne peut contenir que trois personnes à la fois. On ne peut évidemment pas laisser un missionnaire seul avec un cannibale.

B) Exemple 2 : Les 7 ponts de Königsberg

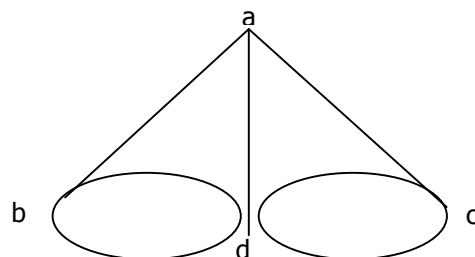
La ville de Königsberg, devenue depuis Kaliningrad, est traversée par la rivière la Pregel. 7 ponts relient 4 quartiers différents. Les habitants de cette ville se demandaient s'il était possible d'effectuer un circuit, passant par chacun des 7 ponts une et une seule fois.



Modélisation du problème des 7 ponts par un graphe :

Le graphe est non orienté puisqu'on suppose être à pied et que les ponts sont franchissables dans chaque sens.

A chaque quartier on associe un sommet et à chaque pont on associe une arête reliant les quartiers situés aux 2 extrémités du pont.



Le problème d'Euler est alors traduit par :

En partant d'un sommet quelconque, peut-on y revenir en ayant parcouru les arêtes une fois et une seule ?

Le problème est maintenant modélisé, il reste à le résoudre !

Résolution du problème des 7 ponts

Dans le cas général, le problème est le suivant :

Etant donné un graphe non orienté, peut-on trouver un cycle passant par toutes les arêtes une fois et une seule ?

Un tel cycle s'appelle un cycle... eulérien.

Euler a établi le résultat suivant :

Proposition

Une condition nécessaire et suffisante pour qu'un graphe possède un cycle eulérien est qu'il soit connexe(*) et que tous ses sommets soient de degré(**) pair.

(*) Un graphe est connexe si deux sommets quelconques sont reliés par une chaîne.

(**) Le degré d'un sommet est le nombre de sommets voisins (voir § suivant).

Pour le graphe des 7 ponts, tous les sommets sont de degré impair : par exemple, le sommet "a" possède 3 voisins "b", "c" et "d". Donc, pour le problème posé par les habitants de Königsberg, la réponse était "non", il n'est pas possible de trouver un cycle eulérien dans ce graphe donc il n'est pas possible de parcourir tous les ponts une fois et une seule et revenir au point de départ.

C) Exemple 3 : Un problème d'ordonnancement de tâches

On doit réaliser 4 pièces qui nécessite chacune 2 opérations sur 2 machines différentes.

Les pièces A et C passent sur la machine 1 puis sur la machine 2, alors que les pièces B et D passent d'abord sur la machine 2 puis sur la machine 1.

Les opérateurs de chaque machine ont reçu les consignes suivantes pour réaliser les différentes opérations : sur la machine 1 l'ordre doit être A, B, D et C alors que sur la machine 2 l'ordre doit être A, B, C et D.

On souhaite planifier les différentes opérations.

Représentation des données du problème par un graphe

On représente la succession des différentes opérations par un graphe dont les sommets correspondent aux différentes opérations : par exemple, le sommet A1 correspond à la pièce A passant sur la machine 1.

On introduit ainsi 8 sommets associés aux 8 opérations.

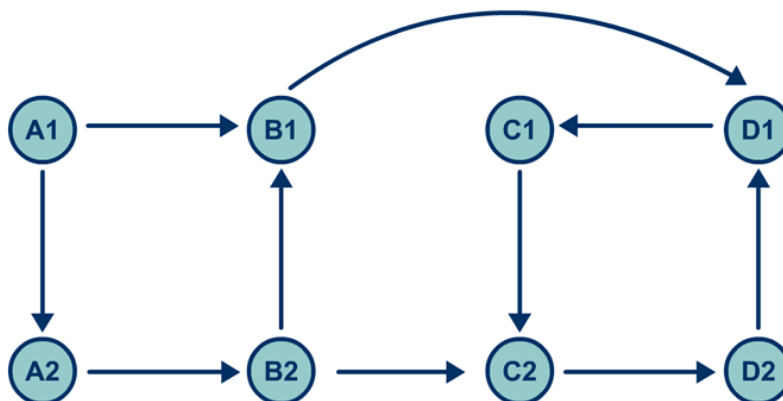
Représentations des différentes contraintes de succession par des arcs :

Contraintes sur la fabrication des pièces

On relie par un arc les sommets correspondant à deux opérations qui doivent se succéder pour une pièce donnée : par exemple, la contrainte "A passe d'abord sur la machine 1 puis sur la machine 2" est représenté par un arc entre les 2 sommets A1 et A2.

Contraintes sur les ordres de fabrication sur les 2 machines

On relie par un arc les sommets correspondant à deux opérations qui doivent se succéder sur une machine donnée : par exemple, sur la machine 1 l'ordre imposé est A, B, D puis C. L'opération B1 doit être précédée de l'opération A1, ce qui se représente par un arc de A1 vers B1.



On constate sur ce graphe qu'il est impossible de planifier l'ensemble des opérations :
on peut exécuter A1 puis A2 puis B2 et B1, mais C2 doit attendre C1 qui doit attendre D1 qui doit attendre D2 et D2 doit attendre C2 !
Ce graphe possède un circuit qui traduit l'impossibilité de réaliser les tâches dans l'ordre imposé !

III Représentations d'un graphe

La représentation sagittale d'un graphe ne permet pas d'effectuer de traitements sur les données associées à ce graphe. Il faut pour cela en avoir une représentation manipulable par un programme informatique.

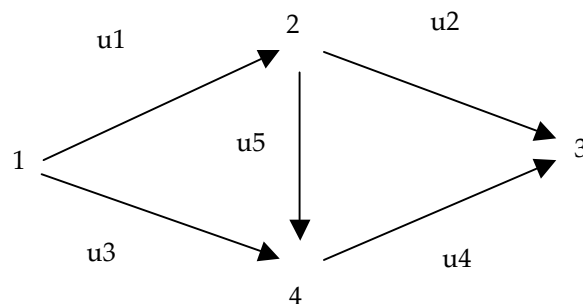
Plusieurs représentations sont possibles.

A - Représentation par la description des arcs

On peut représenter un graphe par la donnée de $G = (X,U,I,T)$ avec :

- X ensemble de sommets (ou nœuds)
- U ensemble d'arcs
- I et T deux applications de U dans X où I(u) est l'origine de l'arc u et T(u) l'extrémité de u.

Exemple



Pour le graphe précédent, on a :

$X = \{1,2,3,4\}$ $U = \{u1,u2,u3,u4,u5\}$

u	u1	u2	u3	u4	u5
I(u)	1	2	1	4	2
T(u)	2	3	4	3	4

B - Représentation par listes d'adjacence

Soit un arc $u = (x,y)$:

- x est un "**prédécesseur**" de y et y un "**successeur**" de x
- x et y sont des sommets "**adjacents**" (ou voisins).

On note **Succ(x)** l'ensemble des successeurs de x.

On peut définir un graphe en décrivant pour chaque sommet la liste Succ(x) de ses successeurs.

On note **Pred(x)** l'ensemble des prédécesseurs de x.

On peut définir un graphe en décrivant pour chaque sommet la liste Pred(x) de ses prédécesseurs.

Exemple

Le graphe précédent est parfaitement défini par la donnée de :

$X = \{1,2,3,4\}$ et $\text{Succ}(1) = \{2,4\}$ $\text{Succ}(2) = \{3,4\}$ $\text{Succ}(3) = \emptyset$ $\text{Succ}(4) = \{3\}$
ou par la donnée de :

$X = \{1,2,3,4\}$ et $\text{Pred}(1) = \emptyset$ $\text{Pred}(2) = \{1\}$ $\text{Pred}(3) = \{2,4\}$ $\text{Pred}(4) = \{1,2\}$

Le nombre de successeurs de x , noté $d^+(x)$ est le **demi-degré extérieur** de x .

Le nombre de prédécesseurs de x , noté $d^-(x)$ est le **demi-degré intérieur** de x .

Le nombre de sommets adjacents à un sommet est le **degré** $d(x)$ du sommet x .

On a $d(x) = d^+(x) + d^-(x)$

Proposition

La somme des degrés de tous les sommets est égal à deux fois le nombre d'arcs.

Démonstration

Soit m le nombre d'arcs.

On a :

$m = \sum_{x \in X} d^+(x)$: il suffit de dénombrer les arcs à partir de leur origine

$m = \sum_{x \in X} d^-(x)$: il suffit de dénombrer les arcs à partir de leur extrémité.

Comme $d(x) = d^+(x) + d^-(x)$, on a bien $\sum_{x \in X} d(x) = 2m$.

Ce résultat est connu sous le nom de **lemme des poignées de mains**.

Ceci est du au corollaire suivant et à son application.

Proposition

Dans un graphe, le nombre de sommets de degré impair est pair.

Démonstration

La somme des degrés de tous les sommets est égale à $2m$, donc paire. Il ne peut donc y avoir qu'un nombre pair de sommets de degré impair.

Application

Dans une assemblée, on prétend que le nombre de personnes ayant serré la main à un nombre impair de personnes est pair ! Comment le prouver ?

Ce problème est modélisé par un graphe non orienté dont les sommets sont associés aux personnes, et les arêtes associées aux couples de personne ayant échangé une poignée de mains.

Si on considère le sommet associé à un individu donné, le degré de ce sommet correspond aux nombres de personnes auxquelles l'individu a serré la main. D'après le lemme des poignées de mains, il ne peut y avoir qu'un nombre pair de personnes (sommets) qui ont serré la main à un nombre impair de personnes (ses voisins dans le graphe).

C - Représentation d'un graphe par une matrice

Dans certains problèmes, il peut être utile de représenter le graphe par une matrice.

Soit un graphe de n sommets numérotés de 1 à n .

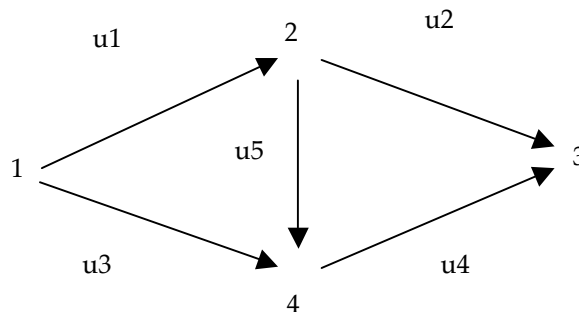
On appelle **matrice d'adjacence** du graphe la matrice carrée A , de dimension n , dont l'élément générique a_{ij} est défini par :

$$a_{ij} = \begin{cases} 1 & \text{si } \exists u \in U \text{ } u = (i,j) \\ 0 & \text{sinon} \end{cases}$$

Les éléments de la matrice sont 1 ou 0 : l'élément de la i ème ligne et la j ème colonne vaut 1 si l'arc (i,j) existe, 0 sinon.

Exemple

Le graphe



a pour matrice d'adjacence

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Selon le problème que l'on aura à traiter, on retiendra une représentation plutôt qu'une autre de manière à limiter le nombre de calculs.

IV Détermination d'un circuit dans un graphe et tri topologique

Comme nous l'avons vu dans le problème d'ordonnement de tâches, il peut être nécessaire de déterminer si un graphe possède ou non un circuit. S'il n'en possède pas, on pourra numéroter les sommets dans un ordre particulier.

L'algorithme suivant permet de répondre à la question : existe-t-il ou non un circuit dans un graphe ?

Pour le mettre en oeuvre de manière efficace, il faut que le graphe soit représenté de telle manière que les prédécesseurs de chaque sommet soient aisément accessibles.

L'algorithme est le suivant :

- 1 - Chercher un sommet sans prédécesseur
SI il n'en existe pas ALORS Stop "le graphe contient un circuit"
SINON
 en choisir un
 lui donner 1 comme numéro
 Poser $k = 1$
- 2 - SI tous les sommets sont numérotés ALORS Stop " le graphe ne possède pas de circuit "
SINON
 Poser $k = k+1$
 Rechercher un sommet sans prédécesseur ou dont tous les prédécesseurs sont déjà numérotés
 SI il n'en existe pas ALORS Stop "le graphe possède un circuit"
 SINON
 en choisir un
 lui donner k comme numéro
 Retour en 2

Proposition

L'algorithme précédent se termine avec tous les sommets numérotés si et seulement si le graphe est sans circuit.

Démonstration

Si l'algorithme s'arrête alors que tous les sommets ne sont pas encore numérotés, c'est qu'il existe un sous-ensemble S de sommets tel que chacun de ces sommets possède au moins un prédécesseur non numéroté.

En partant d'un des sommets de S , on peut, en passant à son prédécesseur non numéroté et au prédécesseur non numéroté de ce nouveau sommet ..., construire une suite de sommets qui constitue un chemin dans S .

Dans ce parcours, on repassera nécessairement par un sommet déjà visité puisqu'il n'y a qu'un nombre fini de sommets, mettant ainsi en évidence un circuit.

Réciproquement, si tous les sommets ont pu être numérotés c'est qu'il n'existe pas de circuit. Démontrons le par l'absurde. Supposons qu'il existe un circuit. Pour numéroté un sommet de ce circuit il faut que son prédécesseur sur le circuit soit numéroté donc que le prédécesseur du prédécesseur soit numéroté... ainsi de suite. Pour numéroté un sommet, il faudrait donc qu'il soit lui-même numéroté.

Commentaires

1 - On notera que si cet algorithme permet de dire si le graphe possède un circuit, il ne permet pas de savoir où il se trouve.

2 - Ordre de grandeur du nombre d'opérations : pour mettre en oeuvre cet algorithme la représentation du graphe la mieux adaptée est celle qui donne l'accès pour chacun à ses prédécesseurs ; le temps de calcul croît alors comme le nombre d'arcs du graphe.

L'algorithme précédent permet de détecter si le graphe possède ou non un circuit.

S'il n'en possède pas, il permet en outre de numéroté les sommets et donc le cas échéant de les parcourir dans un ordre tel que chaque sommet a un numéro plus grand que celui de chacun de ses prédécesseurs.

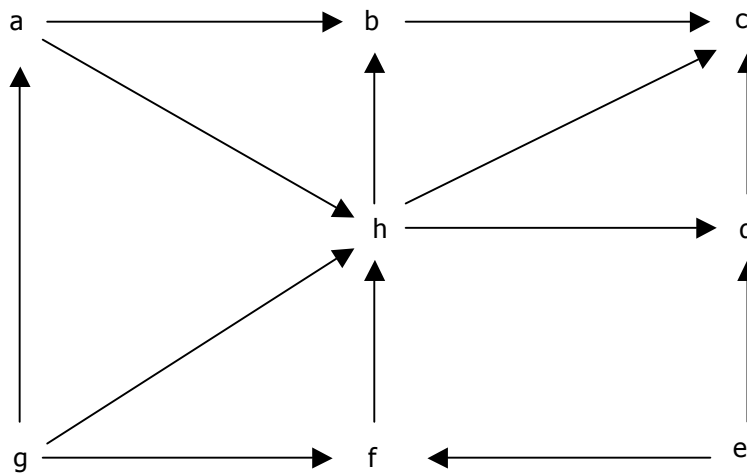
Définition du tri topologique

Les sommets sont numérotés dans l'ordre d'un **tri topologique** si chaque sommet a un numéro plus grand que celui de chacun de ses prédécesseurs.

D'après la proposition précédente, ceci n'est possible que si le graphe est sans circuit.

Application

Considérons le graphe



L'application de l'algorithme de détection de circuit à ce graphe donne :

Les sommets e et g sont sans prédécesseurs : on peut commencer par numéroter e par 1.

On peut alors numéroter g par 2.

Dès que g est numéroté, on peut numéroter a puis f dont les 2 prédécesseurs (e et g) sont maintenant numérotés. On numérotera ensuite le sommet h puis b et d, puis enfin c.

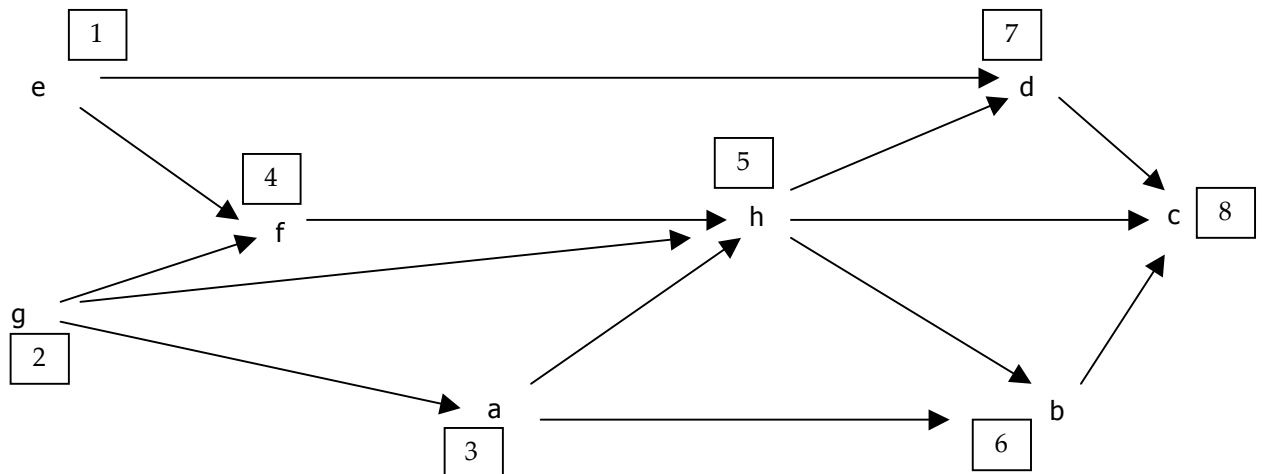
Un ordre de numérotation possible est le suivant :

$\text{num}(e) = 1, \text{num}(g) = 2, \text{num}(a) = 3, \text{num}(h) = 4, \text{num}(d) = 5, \text{num}(b) = 6, \text{num}(d) = 7, \text{num}(c) = 8$

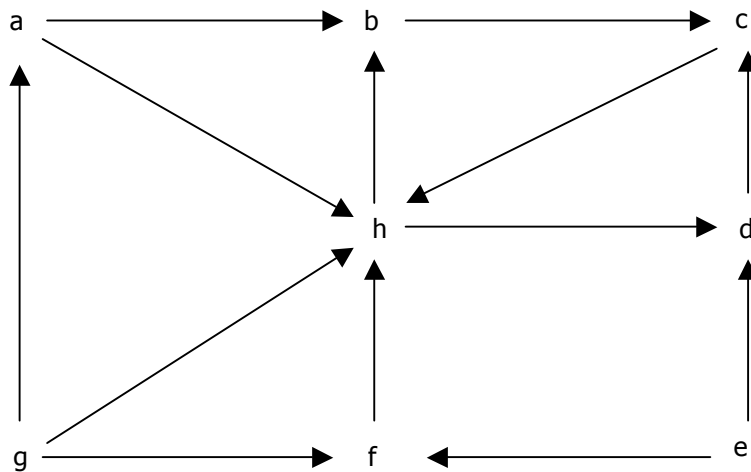
Tous les sommets sont numérotables, donc ce graphe ne possède pas de circuit.

Le graphe de la page précédente peut aussi être représenté de la manière suivante.

Cette représentation permet de mieux visualiser la numérotation et le fait que ce graphe soit sans circuit.



Si on change le sens de l'arc (h,c) on obtient le graphe suivant :



On peut commencer l'algorithme en numérotant les sommets e, g, a et f mais aucun des 4 autres sommets n'est numéroté : ils ont tous un prédécesseur non numéroté. Ce graphe possède donc un circuit que l'on peut trouver sur ce graphe : b c h b est un circuit.