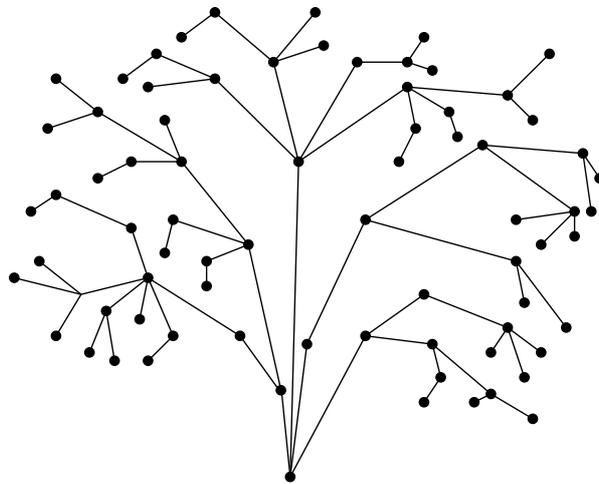




Faculté des sciences
Département de mathématiques

Théorie des graphes



Deuxièmes bacheliers en sciences mathématiques
Année académique 2009–2010
Michel Rigo

Table des matières

Introduction	1
Chapitre I. Premier contact avec les graphes	5
1. Graphes orientés	5
2. Graphes non orientés	8
3. Quelques exemples	12
4. Chemins et circuits	22
4.1. Recherche du plus court chemin	25
4.2. Graphes et chemins eulériens	29
4.3. Connexité des graphes non orientés	32
4.4. Décomposition en composantes fortement connexes	33
5. Sous-graphes	37
6. Coupes, points d'articulation, k -connexité	38
7. Théorème(s) de Menger	44
8. Graphes orientés sans circuit et tri topologique	46
9. Arbres	49
9.1. Parcours d'arbres	51
10. Isomorphismes de graphes	53
11. Graphes hamiltoniens	56
11.1. Fermeture d'un graphe et théorème de Chvátal	62
Chapitre II. Un peu de théorie algébrique des graphes	69
1. Matrice d'adjacence	69
2. Théorie de Perron-Frobenius	73
2.1. Période d'une matrice irréductible	79
2.2. Estimation du nombre de chemins de longueur n	83
2.3. Cas d'un graphe ayant plusieurs composantes f. connexes	85
3. Une application : l'algorithme de PageRank	87
4. Algèbre d'adjacence	92
5. Arbres couvrants	95
5.1. Une formule de Cayley	96
5.2. Une preuve bijective	98
5.3. Nombre de sous-arbres couvrants	102
6. Arbres couvrants de poids minimal	108
Chapitre III. Graphes planaires	111
1. Introduction	111

2. Formule d'Euler	113
3. Graphes homéomorphes	115
4. Théorème de Kuratowski	116
Chapitre IV. Coloriage	123
1. Nombre chromatique	123
2. Le théorème des cinq couleurs	124
3. Polynôme chromatique	132
4. Coloriage d'arêtes et théorème de Ramsey	135
Chapitre V. Annexe : implémentation en C	141
1. Pointeurs, allocation mémoire et listes chaînées	141
1.1. Adresses	141
1.2. Allocation dynamique	144
1.3. Structures	145
1.4. Listes chaînées	146
2. Liste d'adjacence	148
2.1. Une manipulation simplifiée	150
2.2. Utilisation des fonctions	152
2.3. Détail de l'implémentation	153
Bibliographie	161
Liste des figures	163
Index	169

Introduction

La théorie des graphes est, avec la combinatoire, une des pierres angulaires de ce qu'il est commun de désigner par *mathématiques discrètes*. Cependant, elle n'a reçu qu'assez tardivement une attention soutenue de la part de la communauté mathématique. En effet, bien que les graphes eulériens soient l'émanation du célèbre problème des sept ponts de Königsberg étudié par Euler en 1736, on peut dire que les premiers développements majeurs de la théorie des graphes datent du milieu du vingtième siècle (N. Biggs, C. Berge, W.T. Tutte, ...). Ainsi, un des premiers ouvrages, si pas le premier, traitant de théorie des graphes "*Theorie der endlichen und unendlichen Graphen*" écrit par König remonte à 1936. Depuis cette époque, la théorie des graphes s'est largement développée et fait à présent partie du cursus standard en mathématiques de bon nombre d'universités. Ces notes de cours constituent le support écrit du cours dispensé aux *deuxièmes bacheliers en sciences mathématiques* de l'Université de Liège.

Un graphe $G = (V, E)$ est essentiellement défini par une relation binaire $E \subseteq V \times V$ sur un ensemble V le plus souvent fini (nous ne ferons que de brèves incursions dans le monde des graphes infinis, ce qui sera d'ailleurs l'occasion de donner un avant-goût de la théorie des langages formels). Les graphes sont utilisés pour modéliser de nombreuses situations et leurs applications sont par conséquent aussi nombreuses que variées : dans d'autres branches des mathématiques (algèbre, combinatoire, ...), en informatique, en recherche opérationnelle (tournées de distribution, ordonnancement de tâches, construction de circuits imprimés, ...), en cartographie (coloriage de cartes), en chimie, etc ...

Selon nous, il est impensable de vouloir traiter *efficacement* un problème réel (i.e., de taille non négligeable) sans posséder de solides bases. Ainsi, nos développements seront principalement théoriques. Néanmoins, le caractère appliqué de la théorie sera mis en exergue par de nombreux exemples (en particulier, au premier chapitre, nous détaillons une série de problèmes issus du "monde réel".)

Par essence, en mathématiques discrètes et *a fortiori* en théorie des graphes, de nombreux raisonnements ont une composante combinatoire importante. Ainsi, des preuves pouvant être jugées "difficiles" consistent souvent en une succession, parfois longue, de raisonnements simples (mais pas toujours aisés à saisir). Dès lors, certaines parties paraîtront peut-être complexes sans pour autant être compliquées ! Nous espérons que des exemples

bien choisis pourront alors aider le lecteur dans sa compréhension. Bien évidemment, les méthodes mises ici en évidence sont souvent tout autre que celles des “*mathématiques du continu*” comme l’analyse. Nous espérons que cette présentation permettra ainsi à l’étudiant d’enrichir sa palette de techniques et de raisonnements mathématiques. (Les mathématiques ne sont bien sûr pas cloisonnées et les interactions entre les diverses branches sont nombreuses et souvent fort riches: “*It is unquestionable that interplay between ideas from different sources, and elaborate techniques successfully applied, are among the features that make much of mathematics fascinating. Moreover, mathematics does often display a tendency to unify itself and to build up a body of technique. Therefore one may well guess that graph theory, as it matures, will continue to develop its own characteristic techniques and that many of its results will become increasingly unified, both among themselves and with the rest of mathematics.*”¹.)

Dans le premier chapitre, nous présentons la notion de graphe et ses variantes (graphe non orienté, arbre, multi-graphe, hypergraphe). Les principaux résultats du chapitre concernent les graphes eulériens et les graphes hamiltoniens (théorème de Dirac, d’Ore et de Chvátal).

Le deuxième chapitre donne quelques résultats de la théorie algébrique des graphes. En effet, à un graphe G , on associe de manière classique une matrice, appelée matrice d’adjacence. Dès lors, l’application de résultats standards d’algèbre linéaire permet de déduire des renseignements non triviaux sur G . Par exemple, on dénombre le nombre d’arbres couvrants d’un graphe donné par un simple calcul de mineur. Nous présentons également quelques éléments concernant la théorie de Perron-Frobenius qui permet une estimation assez fine du nombre de chemins de longueur n dans un graphe.

Les trois chapitres suivants sont assez classiques. Les thèmes évoqués sont présentés dans bon nombre d’ouvrages d’introduction à la théorie des graphes. On y étudie les graphes planaires, i.e., les graphes pouvant être représentés dans le plan euclidien par une figure dont les arcs ne se coupent pas. Ensuite, on s’intéresse à quelques problèmes de coloriage (théorème des quatre/cinq couleurs) et en particulier, à l’important théorème de Ramsey qui permet d’introduire sommairement la théorie extrémale des graphes. Enfin, suivant le temps disponible, quelques heures du cours seront consacrées à présenter l’un ou l’autre problèmes “d’actualité” de recherche en mathématiques discrètes.

Vu le caractère appliqué indubitable du sujet, le dernier chapitre est une annexe présentant une implémentation en langage \mathbf{C} des graphes finis (orientés) par liste d’adjacence. Cette structure de données est particulièrement bien adaptée aux graphes peu denses. En effet, tout au long du présent texte, divers algorithmes sont présentés sous forme de pseudo-code. Il est donc indispensable de disposer d’une interface raisonnable pour implémenter

¹C. ST. J. A. Nash-Williams, dans la préface de [3].

et tester ces algorithmes sur des jeux de données conséquents. Signalons que de nombreux problèmes rencontrés en théorie des graphes sont, du point de vue de la complexité, difficiles. Cependant, nous avons volontairement omis les notions de complexité algorithmique et de calculabilité : problèmes NP, NP-difficiles ou NP-complets. (Cette problématique est présentée en troisième année de bacheliers en sciences mathématiques.)

Une fois n'est pas coutume, je voudrais aussi remercier (et les futurs étudiants ayant ces notes de cours entre les mains s'associeront certainement à moi) l'ensemble des étudiants de deuxième bachelier de l'année académique 2005–2006 et en particulier : Cédric H., Damien G., Damien K., Damien L., Kastriot, Marie, Mehdi, Mélissa, Naïm, Nicolas, Rukiye, Leur intérêt marqué, leurs questions pertinentes, leurs interrogations m'ont permis d'améliorer (et parfois de corriger) la présentation de ce cours. Je n'oublierai pas non plus les retours et les suggestions de Laurent Waxweiler.

CHAPITRE I

Premier contact avec les graphes

*Un petit dessin vaut mieux
qu'un grand discours,*

Napoléon.

Introduire une nouvelle matière n'est pas toujours chose plaisante car il s'agit souvent d'une accumulation de définitions ! Et c'est hélas la situation rencontrée ici. Nous allons donc agrémenter cette présentation, autant que faire se peut, d'exemples mettant en lumière l'intérêt pratique de la théorie des graphes. Comme l'indique le titre de ce chapitre, nous nous intéressons aux *graphes* et comme le lecteur va très vite s'en apercevoir, il en existe de plusieurs types: simple, multi-graphe, digraphe, hyper-graphe,...

1. Graphes orientés

Définition I.1.1. Soient V un ensemble (fini ou infini) et E une partie de $V \times V$ (i.e., une *relation* sur V). Le *graphe* $G = (V, E)$ est la donnée du couple (V, E) . Les éléments de V sont appelés les *sommets*¹ ou *noeuds* de G . Les éléments de E sont appelés les *arcs*² ou *arêtes* de G . Si V est fini, on parlera de *graphe fini* (en particulier, E est alors fini et contient au plus $(\#V)^2$ arcs).

Remarque I.1.2. Observons que l'ordre au sein des couples appartenant à E est intrinsèquement présent³. On parlera donc parfois de *graphe orienté* ou de *graphe dirigé*. Soit I , un ensemble d'indices. Si $V = \{v_i \mid i \in I\}$ et si $a = (v_i, v_j)$, $i, j \in I$, on pourra alors parler de l'*origine* v_i et de la *destination* v_j de l'arc a . On dit que v_i et v_j sont les *extrémités* de l'arc a et que a *relie* v_i à v_j . Si $b = (v_i, v_i)$, on parle généralement de la *boucle* b . Il est souvent commode de donner une représentation sagittale d'un graphe. Les sommets

Cette distinction va devenir rapidement indispensable, lorsqu'on introduira les graphes non orientés.

¹En anglais, cela se dit "*vertex*" (au pluriel, "*vertices*"). D'où l'initiale V pour désigner l'ensemble des sommets. Dans ces notes, nous ne dérogerons pas à la coutume anglo-saxonne de noter un graphe $G = (V, E)$.

²En anglais, cela se dit "*edge*". D'où l'initiale usuelle E .

³On parle de *couple* et non de *paire*. Un couple est une *paire ordonnée*. On distingue d'ailleurs les notations (x, y) et $\{x, y\}$.

sont représentés par des points et si (v_i, v_j) est un arc, alors on trace une flèche de v_i vers v_j (cf. figure I.1). Deux arcs sont *adjacents* s'ils ont au moins une extrémité en commun.

arcs adjacents

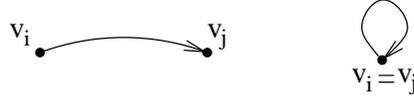


FIGURE I.1. Un arc reliant deux sommets, une boucle.

Définition I.1.3. Soit $a = (v_i, v_j) \in E$. On dit que a est un arc *sortant* de v_i ou encore que a est un *arc incident* à v_i vers l'extérieur (resp. un arc *entrant* dans v_j ou encore que a est un *arc incident* à v_j vers l'intérieur). L'ensemble des arcs sortant de v_i est noté $\omega^+(v_i)$ et l'ensemble des arcs entrant dans v_j est noté $\omega^-(v_j)$. L'ensemble des arcs incidents à un sommet v est $\omega(v) := \omega^+(v) \cup \omega^-(v)$. On définit le *demi-degré sortant* (resp. *demi-degré entrant*) d'un sommet v par

$$d^+(v) = \#(\omega^+(v)) \quad (\text{resp. } d^-(v) = \#(\omega^-(v))).$$

arc incident à un sommet

Handshaking formula.

Si $G = (V, E)$ est un graphe fini, il est clair que

$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v).$$

Enfin, le *degré* de v est $\deg(v) = d^+(v) + d^-(v)$. L'ensemble des *successeurs* d'un sommet v est l'ensemble $\text{succ}(v) = \{s_1, \dots, s_k\}$ des sommets s_i tels que $(v, s_i) \in \omega^+(v)$, i.e., $(v, s_i) \in E$. De manière analogue, l'ensemble des *prédécesseurs* d'un sommet v est l'ensemble $\text{pred}(v) = \{s_1, \dots, s_k\}$ des sommets s_i tels que $(s_i, v) \in \omega^-(v)$, i.e., $(s_i, v) \in E$. Enfin, l'ensemble des voisins de v est simplement

$$\nu(v) = \text{pred}(v) \cup \text{succ}(v).$$

sommets adjacents

Si u appartient à $\nu(v)$, on dit que u et v sont des sommets *voisins* ou *adjacents*.

Exemple I.1.4. Soit le graphe $G = (V, E)$ où $V = \{a, b, c, d, e\}$ et

$$E = \{(a, b), (a, e), (b, b), (b, c), (c, c), (c, d), (c, e), (d, a), (e, a), (e, d)\}.$$

Celui-ci est représenté à la figure I.2. Par exemple, $\omega^+(a) = \{(a, b), (a, e)\}$

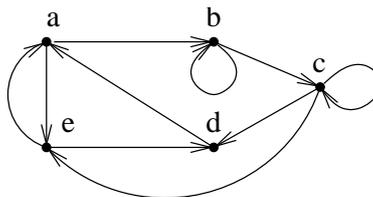


FIGURE I.2. Un exemple de graphe.

et $\omega^-(d) = \{(c, d), (e, d)\}$. On a aussi $\text{succ}(a) = \{b, e\}$, $\text{succ}(b) = \{b, c\}$, $\text{pred}(d) = \{c, e\}$ et $\nu(a) = \{b, d, e\}$. On voit aussi que les arcs (e, a) et (d, a) sont adjacents. Enfin, le demi-degré sortant de c est $d^+(c) = 3$.

Définition I.1.5. Un *multi-ensemble*⁴ est un ensemble au sein duquel un même élément peut être répété plus d'une fois. Ainsi, on s'intéresse non seulement à savoir si un élément appartient ou non à un multi-ensemble donné, mais également à sa multiplicité. Par exemple, $\{1, 1, 2, 3\}$, $\{1, 2, 3\}$ et $\{1, 2, 2, 3\}$ sont des multi-ensembles distincts. Pour distinguer les copies d'un même élément x , il est commode de les indiquer. Par exemple, on considère le multi-ensemble $\{1_1, 1_2, 1_3, 2_1, 2_2, 3\}$. Cette manière de procéder nous permettra de définir facilement des fonctions définies sur un multi-ensemble.

On suppose qu'un élément est répété au plus un nombre dénombrable de fois.

Un *multi-graphe* $G = (V, E)$ est un graphe pour lequel l'ensemble E des arcs est un multi-ensemble. Autrement dit, il peut exister plus d'un arc reliant deux sommets donnés. Un exemple de représentation d'un multi-graphe est donné à la figure I.3. Un multi-graphe $G = (V, E)$ est *fini* si V

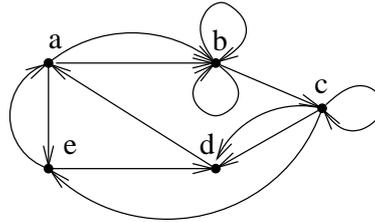


FIGURE I.3. Un exemple de multi-graphe.

et E sont finis. (En effet, dans le cas des multi-graphes, supposer V fini n'implique pas que E soit fini.)

Soit $p \geq 1$. Un *p-graphe* est un multi-graphe $G = (V, E)$ pour lequel tout arc de E est répété au plus p fois. En particulier, un *1-graphe* est un graphe.

Remarque I.1.6. On peut observer que la remarque I.1.2, la définition I.1.3 et la “handshaking formula” s'appliquent également au cas des multi-graphes. Il est laissé au lecteur le soin d'adapter les définitions de $\omega^+(v)$, $d^+(v)$, $\text{succ}(v)$ et $\omega^-(v)$, $d^-(v)$, $\text{pred}(v)$. En particulier, $\omega^+(v)$ et $\omega^-(v)$ sont en général des multi-ensembles.

Définition I.1.7. Un graphe $G = (V, E)$ est dit *simple* (ou *strict*) s'il ne s'agit pas d'un multi-graphe et si E est irréflexif, c'est-à-dire que quel que soit $v \in V$, (v, v) n'appartient pas à E (i.e., G ne contient pas de boucle). Un exemple de graphe simple est donné à la figure I.4.

⁴Cette définition n'est pas très rigoureuse. En effet, le concept même d'ensemble ne vous a, jusqu'à présent, été introduit que de manière naïve.

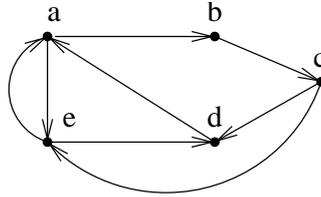


FIGURE I.4. Un exemple de graphe simple.

2. Graphes non orientés

Les graphes non orientés sont en fait un cas particulier de graphes (orientés).

Définition I.2.1. Soit $G = (V, E)$ un graphe (resp. un multi-graphe). Si E est une relation symétrique sur V , on dira que G est un graphe (resp. un multi-graphe) *non dirigé* ou *non orienté*. Autrement dit, G est non dirigé si

$$\forall v_1, v_2 \in V : (v_1, v_2) \in E \Rightarrow (v_2, v_1) \in E.$$

Dans ce cas, on simplifie la représentation sagittale de G en traçant simplement un segment entre v_1 et v_2 . Pour alléger l'écriture, on identifiera les arcs (v_i, v_j) et (v_j, v_i) avec une unique "arête non orientée" donnée par la paire $\{v_i, v_j\}$. Dans le cas dirigé (resp. non dirigé), nous nous efforcerons de parler d'arcs (resp. d'arêtes).

Si par contre, on désire insister sur le caractère non symétrique de E , on parlera de graphe *dirigé* ou, par abus de langage, *digraphe*⁵.

Les définitions rencontrées précédemment s'adaptent aisément au cas non orienté.

Définition I.2.2. Soient $G = (V, E)$, un multi-graphe non orienté et $a = \{v_i, v_j\}$ une de ses arêtes. On dit que a est *incidente* aux sommets v_i et v_j . Le nombre d'arêtes incidentes à v_i est le *degré* de v_i , noté $\deg(v_i)$. On suppose en outre que les **boucles** apportent une double contribution au degré d'un sommet. L'ensemble des arêtes incidentes à v_i se note $\omega(v_i)$. Il est clair que, dans un graphe simple, $\deg(v_i) = \#\omega(v_i)$. Ces notations sont bien évidemment compatibles avec celles données dans le cas orienté. Deux arêtes sont *adjacentes* si elles ont au moins une extrémité en commun.

Deux sommets $v_i, v_j \in V$ sont *adjacents* si l'arête $\{v_i, v_j\}$ appartient à E . On dit aussi qu'ils sont *voisins*. L'ensemble des voisins de v se note $\nu(v)$. Enfin, la définition d'un p -graphe est analogue à celle donnée dans le cas orienté.

Remarque I.2.3 (Handshaking lemma). Si $G = (V, E)$ est un multi-graphe non orienté, alors

$$\sum_{v \in V} \deg(v) = 2 \#E.$$

⁵En anglais "directed graph" se contracte en "digraph".

arête incidente
à un sommet

arêtes adjacentes
sommets adjacents

C'est immédiat. (Et on comprend mieux la double contribution des boucles pour le degré d'un sommet...)

L'exemple suivant illustre les différentes classes de graphes rencontrées jusqu'à présent. Bien sûr, tout graphe simple est un graphe et tout graphe est un multi-graphe.

Exemple I.2.4. A la figure I.5, on a représenté, dans le cas dirigé, un graphe simple, un graphe et enfin, un multi-graphe. La figure I.6 reprend

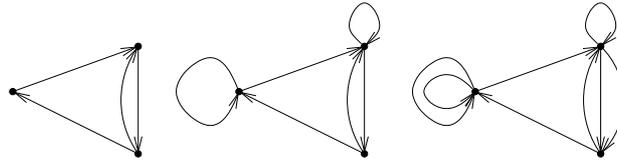


FIGURE I.5. Un graphe (dirigé) simple, un graphe et un multi-graphe.

les mêmes éléments dans le cas non orienté.

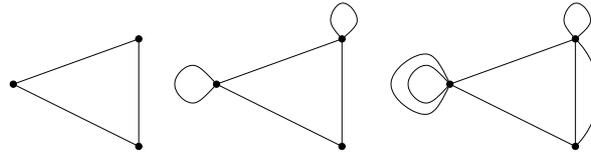


FIGURE I.6. Un graphe (non dirigé) simple, un graphe et un multi-graphe.

Définition I.2.5. Soit $k \geq 1$. Un multi-graphe orienté (resp. non orienté) $G = (V, E)$ est k -régulier si pour tout $v \in V$, $d^+(v) = k$ (resp. $\deg(v) = k$). Le graphe de gauche (resp. de droite) de la figure I.7 est 3-régulier (resp. 4-régulier). Le graphe de droite de la figure I.7 est en particulier simple et

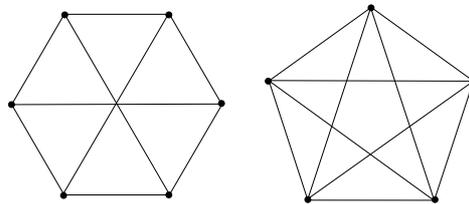


FIGURE I.7. Des graphes non orientés 3 et 4-réguliers.

complet. Un graphe $G = (V, E)$ est *complet* si $E = V \times V$, plus exactement, on suppose souvent que

$$E = V \times V \setminus \{(v, v) \mid v \in V\}$$

(autrement dit, on ne tient pas compte des boucles). En particulier, un graphe complet est symétrique. On note K_n le graphe simple non orienté

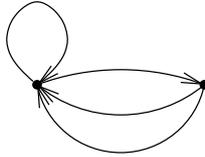


FIGURE I.8. Un multi-graphe orienté 2-régulier.

complet à n sommets. Ainsi, la figure I.7 représente le graphe K_5 . Dans ce cours, lorsqu'on parlera de graphes complets, il sera sous-entendu qu'il s'agit de graphes simples et non orientés.

Définition I.2.6. Un graphe $G = (V, E)$ est dit *biparti* si V peut être partitionné en deux ensembles V_1 et V_2 de manière telle que $E \subseteq V_1 \times V_2$. Si $\#V_1 = m$, $\#V_2 = n$ et $E = V_1 \times V_2$, alors on parle du graphe biparti

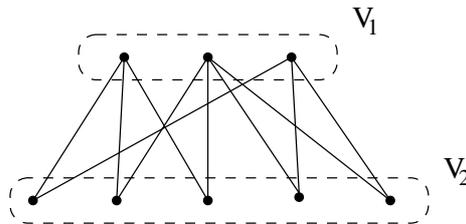


FIGURE I.9. Un graphe biparti (non complet).

complet et il est noté $K_{m,n}$. On peut généraliser cette notion et définir des graphes n -partis, pour $n \geq 2$. Pour ce faire, V doit être partitionné en n sous-ensembles V_1, \dots, V_n de manière telle que

$$E \subseteq \bigcup_{i \neq j} V_i \times V_j.$$

Définition I.2.7. Un multi-graphe $G = (V, E)$ (orienté ou non) est *étiqueté* (par f) s'il existe une fonction

$$f : E \rightarrow \Sigma$$

où Σ est un ensemble quelconque. Si $\Sigma \subseteq \mathbb{R}^+ = [0, +\infty[$, on parle souvent de multi-graphe *pondéré* et on dit que f est une *fonction de poids*. Un étiquetage peut par exemple servir à préciser des coûts (coût de transport, des distances, des couleurs, etc...). Si a est un arc, $f(a)$ est l'*étiquette*, le *label* ou encore le *poids* de a . On peut de la même manière définir un étiquetage des sommets au moyen d'une fonction $g : V \rightarrow \Sigma$.

Exemple I.2.8. Le graphe de la figure I.10 représente quelques villes belges connectées par un réseau autoroutier. L'étiquette de chaque arête représente la distance, par autoroute, entre les deux extrémités de celle-ci. Nous avons choisi un graphe non orienté car les autoroutes belges vont toujours dans les deux sens.

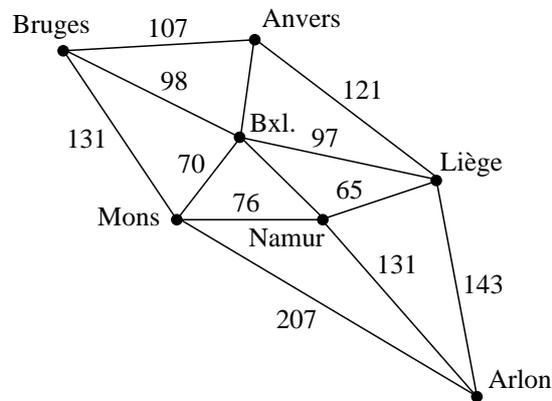


FIGURE I.10. Graphe étiqueté par les distances entre villes (itinéraire “express” calculé par www.mappy.fr).

Pour terminer cette section, nous généralisons le concept de graphe en autorisant non plus des relations binaires entre sommets (autrement dit, des arcs) mais des relations d’arité quelconque (des hyper-arêtes). Ensuite, nous donnons une brève introduction aux matroïdes.

Définition I.2.9. Un *hyper-graphe* $H = (V, E)$ est la donnée de deux ensembles V et E . L’ensemble V , comme dans le cas d’un graphe, est l’ensemble des sommets de H . Par contre, E est une partie de $\mathcal{P}(V)$ (l’ensemble des parties de V). Un élément de E est appelé *hyper-arête*. Soient $V = \{a, b, c, d, e\}$ et

$$E = \{\{a, b\}, \{a, c, d\}, \{b, d, e\}\}.$$

On dispose de l’hyper-graphe $H = (V, E)$ représenté à la figure I.11. Un

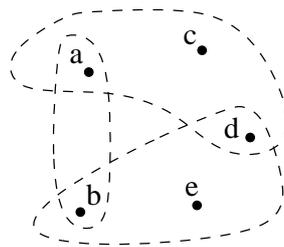


FIGURE I.11. Un exemple d’hyper-graphe.

hyper-graphe $H = (V, E)$ est *fini* si V est fini.

La notion de matroïde est due à H. Whitney (1935) et a été développée par W. Tutte. Elle permet notamment l’étude axiomatique des propriétés de l’indépendance linéaire ou aussi l’étude des cycles et des arbres.

Définition I.2.10. Un *matroïde* (M, \mathcal{I}) est la donnée d’un ensemble fini M et d’une partie \mathcal{I} de $\mathcal{P}(M)$, i.e., d’une collection de sous-ensembles de M , vérifiant les trois propriétés suivantes

- ▶ $\emptyset \in \mathcal{I}$ (de manière équivalente, \mathcal{I} est non vide),
- ▶ si $X \in \mathcal{I}$ et $Y \subseteq X$, alors $Y \in \mathcal{I}$,
- ▶ si $X, Y \in \mathcal{I}$ et $\#X > \#Y$, alors il existe $x \in X \setminus Y$ tel que $Y \cup \{x\} \in \mathcal{I}$.

Les ensembles de \mathcal{I} sont dits *indépendants*. Enfin, un matroïde est qualifié de *normal* si tous les singletons sont indépendants.

Exemple I.2.11. Soient E un espace vectoriel, M une partie finie de E et \mathcal{I} la collection des parties libres de M . Le couple (M, \mathcal{I}) est un matroïde.

Exemple I.2.12. Soit $G = (V, E)$ un graphe non orienté simple. On considère le matroïde (E, \mathcal{I}) où une partie X de E appartient à \mathcal{I} si et seulement si X ne contient aucun cycle (i.e., X est une forêt).

Remarque I.2.13. Tout matroïde (M, \mathcal{I}) est bien évidemment un hypergraphe. Les sommets de ce dernier sont les éléments de M et les hyper-arêtes sont les éléments de \mathcal{I} .

3. Quelques exemples

Nous allons présenter dans cette courte section quelques exemples de graphes permettant la modélisation de divers problèmes. Puisqu'il ne s'agit que d'exemples, certaines définitions sont volontairement omises. Elles seront précisées en temps utiles. Nous espérons que la variété des exemples présentés servira de motivation profonde à l'étude théorique réalisée dans les sections et chapitres suivants. Pour des raisons évidentes de présentation, nous ne donnons que des exemples de "petite taille" qui peuvent souvent être résolus sans véritable méthode. Dans des problèmes réels, il faut imaginer des graphes pouvant avoir plus de 10^6 sommets. Dans ce cas, la solution paraît nettement moins évidente !

Exemple I.3.1 (Circuit eulérien). Les ouvrages de théorie des graphes reprennent toujours le célèbre exemple des *ponts de Königsberg*. Nous ne dérogerons pas à cette règle. Au XVII-ième siècle, les habitants de Königsberg (actuel Kaliningrad, ville de Russie proche de la Lituanie et de la Pologne où coule la rivière Pregel) désiraient se promener le dimanche en passant une et une seule fois par chacun des sept ponts de la ville. Les ponts étaient disposés comme à la figure I.12. Une modélisation du problème revient à considérer un graphe ayant comme sommets, les deux rives et les deux îles et comme arêtes, les sept ponts. Puisqu'il n'y a aucune contrainte sur le sens de parcours des ponts, nous avons choisi un multi-graphe non orienté.

La question générale sous-jacente est donc de *déterminer pour un multi-graphe donné (éventuellement orienté) s'il existe un circuit, i.e., un chemin fermé, passant une et une seule fois par chaque arête.*

Exemple I.3.2 (Circuit hamiltonien — TSP). Le *problème du voyageur de commerce* (Travel Salesman Problem) est en quelque sorte un "problème

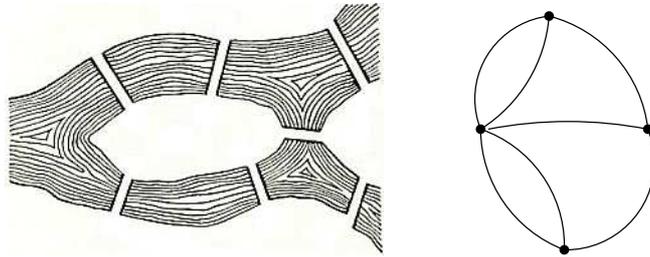


FIGURE I.12. Les sept ponts de Königsberg.

dual” de l'exemple précédent (on s'intéresse ici à trouver un *circuit passant une et une seule fois par chaque sommet* et non par chaque arête). On cherche un circuit permettant non seulement de relier n villes en passant une et une seule fois par chacune d'entre elles, mais de plus, ce circuit doit minimiser la distance totale parcourue. On pourrait par exemple considérer les villes belges et les données de la figure I.10 et en déterminer un circuit optimal. Dans certains cas, on a recours à un graphe orienté plutôt qu'à un graphe non orienté comme à la figure I.10. Cela a pour avantage de permettre la modélisation de coûts différents pour aller d'un sommet A à un sommet B plutôt que de B à A (ceci permet, par exemple, de prendre en compte des sens uniques, des payages, des temps de transports différents suivant la direction choisie, etc. . .).

Il s'agit donc de problèmes typiques rencontrés dans la détermination de tournées, de circuits de distribution, etc. . .

Exemple I.3.3 (Coloriage). Considérons un cube. Si chaque sommet (resp. chaque arête) d'un graphe représente un sommet (resp. une arête) du cube, on obtient le graphe de gauche de la figure I.13, on parle du *squelette* du cube. Par contre, si on représente les faces du cube par les sommets d'un graphe et si les sommets du graphe correspondant à des faces du cube ayant une arête commune sont adjacents, on obtient celui de droite. On peut alors

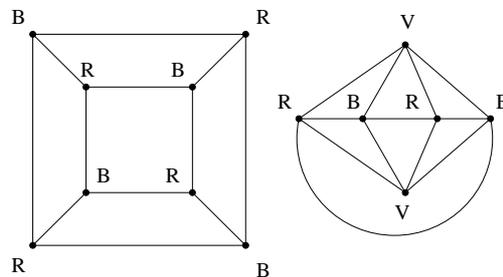


FIGURE I.13. Deux représentations d'un cube.

poser la question générale de *déterminer le nombre de couleurs nécessaire et suffisant pour colorier les sommets d'un multi-graphe donné, de manière telle que deux sommets adjacents ne reçoivent pas la même couleur*. Si on

répond à cette question dans le cas de notre exemple initial, on s'aperçoit que pour colorier les sommets (resp. les faces d'un cube), deux (resp. trois) couleurs sont suffisantes.

Exemple I.3.4 (Cartographie). Quel est le nombre minimum de couleurs nécessaires pour colorier les pays d'une carte⁶ géographique de manière telle que des couleurs distinctes soient attribuées à des pays voisins ? Ce problème

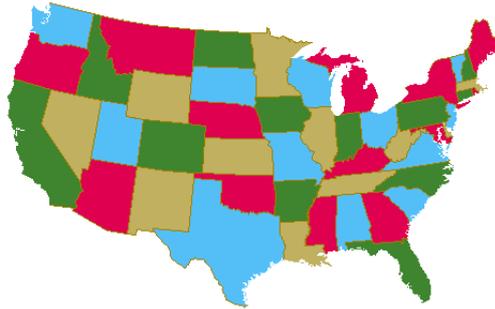


FIGURE I.14. La carte des USA.

se ramène au précédent. On considère un graphe ayant pour sommet les différents pays de la carte. Deux sommets sont adjacents si et seulement si ils partagent une frontière.

Exemple I.3.5 (Graphe d'incompatibilité). Pour le transport de produits chimiques par rail, certains de ces produits ne peuvent être transportés dans un même wagon (des produits co-combustibles sont placés dans des wagons distincts). La figure I.15 représente le graphe d'incompatibilité de transport (deux sommets adjacents ne peuvent être placés dans le même wagon). On

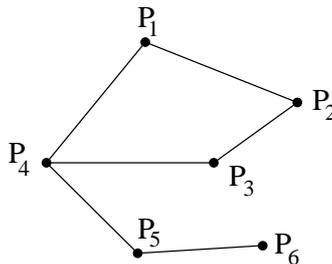


FIGURE I.15. Graphe d'incompatibilité.

demande de minimiser le nombre de wagons nécessaires au transport. Ce problème se ramène donc à un problème de coloriage. Deux sommets adjacents doivent avoir des couleurs distinctes, une couleur correspondant à un wagon.

Par rapport à l'exemple précédent, le graphe n'est pas nécessairement planaire.

⁶La figure I.14 est tirée de <http://www.math.gatech.edu/~thomas/FC/fourcolor.html>

Exemple I.3.6 (Coloriage d'arêtes). Au lieu de vouloir colorier les sommets d'un multi-graphe, on peut aussi s'intéresser au problème suivant. *Déterminer le nombre de couleurs nécessaires et suffisantes pour colorier les arêtes d'un multi-graphe donné, de manière telle que deux arêtes adjacentes ne reçoivent pas la même couleur.*

Exemple I.3.7 (Graphe de Cayley). Soit G un groupe et S un ensemble de générateurs de G (tout élément de G s'obtient comme produit d'un nombre fini d'éléments de S ou d'inverses d'éléments de S). Le *graphe de Cayley* du groupe G (par rapport à S) est un graphe orienté $\mathcal{C}_S(G)$ ayant pour sommets les éléments de G . Ses arcs sont définis comme suit. Pour tous $g \in G$, $s \in S$, l'arc (g, gs) est un arc du graphe ayant s pour label. Soit le groupe symétrique \mathcal{S}_3 des permutations à 3 éléments ayant pour ensemble générateur⁷

$$S = \{x = (1\ 2), y = (1\ 3), z = (2\ 3)\}.$$

Il est clair que

$$\begin{aligned} xx = yy = zz = id, \quad xy = (1\ 2)(1\ 3) = (1\ 3\ 2), \\ xz = (1\ 2)(2\ 3) = (1\ 2\ 3), \quad yz = (1\ 3)(2\ 3) = (1\ 3\ 2) \end{aligned}$$

et

$$yx = (1\ 2\ 3), \quad zx = (1\ 3\ 2), \quad zy = (1\ 2\ 3).$$

Le graphe de Cayley correspondant est représenté à la figure I.16.

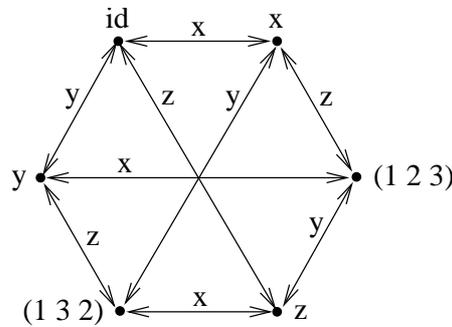
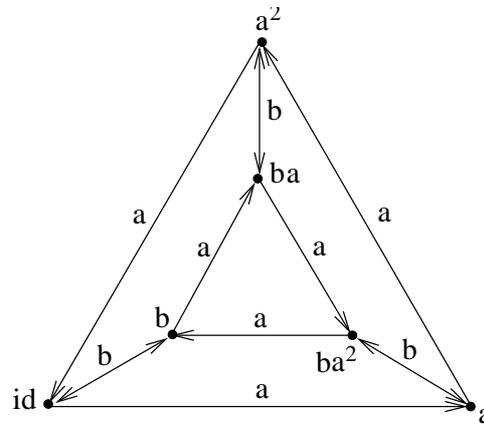


FIGURE I.16. Graphe de Cayley de \mathcal{S}_3 .

On peut obtenir un autre graphe de Cayley de \mathcal{S}_3 en considérant comme ensemble de générateurs $\{a = (1\ 2\ 3), b = (1\ 2)\}$. On obtient alors le graphe de la figure I.17 (les détails de calcul sont laissés au lecteur). Bien évidemment, le graphe de Cayley d'un groupe ne donne aucun renseignement qui ne saurait être obtenu sous une forme "algébrique" classique. Néanmoins, un tel diagramme a l'avantage de fournir presque immédiatement certaines informations qui seraient plus pénibles à obtenir par d'autres moyens (imaginez un groupe défini par certaines relations, par exemple, $cd = dc$, $c^2dc = d$, etc...). Ainsi, il est ici immédiat de vérifier sur la figure I.17 que les éléments

⁷Rappelez-vous, toute permutation est un produit de transpositions.

FIGURE I.17. Un autre graphe de Cayley de S_3 .

$baaba$ et a^2 sont identiques (il suffit de suivre les labels de chemins depuis id). L'utilisation des graphes de Cayley se révèle être un outil souvent bien utile en *théorie des représentations* ou dans la résolution de problèmes précis à l'aide de l'ordinateur.

Exemple I.3.8 (Arbre couvrant). Une société de téléphonie souhaite câbler entièrement, au moyen de nouvelles fibres optiques, une ville en minimisant le nombre de connexions à réaliser. Le nouveau câblage s'appuie sur le réseau électrique déjà existant et bien évidemment, tous les points de la ville doivent être desservis. La figure I.20 représente le réseau actuel de la ville et ses connexions. A droite, se trouve les sélections envisagées. La question générale

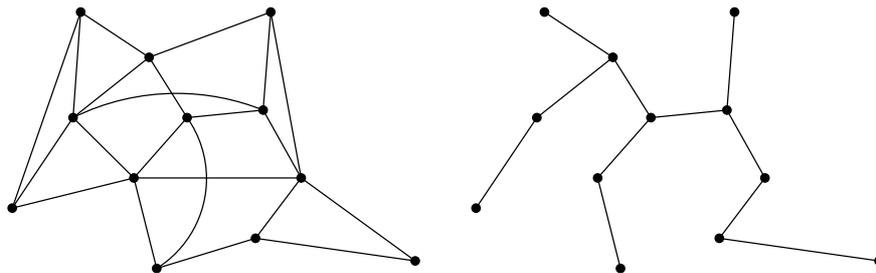


FIGURE I.18. Sous-graphe couvrant.

qui est posée est de *rechercher un sous-graphe (ou un sous-arbre) couvrant dans un graphe donné*. On peut aussi envisager une version pondérée dans laquelle chaque arc aurait un coût et on *rechercherait un sous-graphe (ou un sous-arbre) couvrant de poids minimal*.

Exemple I.3.9 (Forte connexité). Suite à divers problèmes de circulation, des responsables communaux désirent placer les rues d'un quartier à sens unique. Si un graphe modélise les rues et leurs croisements, la question qui se pose est donc d'*orienter les arcs d'un graphe non orienté de manière telle*

qu'il existe un chemin orienté entre toute paire de sommets. Un tel exemple est donné à la figure I.19.

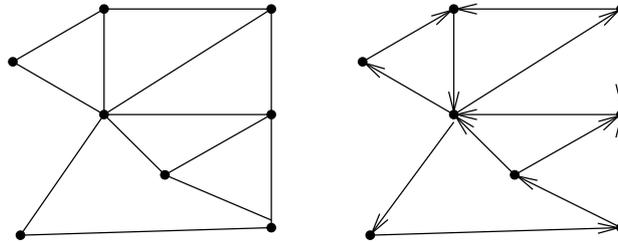


FIGURE I.19. Orientation d'un graphe.

Exemple I.3.10 (Distance). Considérons le problème de routage de paquets de données devant transiter sur un réseau (du type internet). Si un utilisateur désire accéder au contenu d'une page web présente sur un serveur, une connexion entre ce serveur et la machine doit être établie. Cette connexion n'est en général pas directe mais doit passer par une série de machines relais. Pour préserver la bande passante, pour accélérer le transfert ou encore pour minimiser les coûts, on essaie de minimiser le nombre de "hops" (i.e., le nombre de machines relais utilisées). Si chaque ordinateur, routeur, passerelle ou serveur présents sur l'internet représente un sommet d'un graphe dont les arcs représentent les connexions entre ceux-ci, il s'agit d'un problème délicat! Voici un exemple de "route" suivie par un paquet de données :

```
> traceroute www.google.be
traceroute to www.google.be (66.249.85.99), 30 hops max, 40 byte packets
 1 mont3-0014.gw.ulg.ac.be (139.165.159.1) 0.177 ms 0.177 ms 0.163 ms
 2 segi3-0813-mont3.gw.ulg.ac.be (193.190.228.125) 0.221 ms 0.179 ms 0.184 ms
 3 inet3-3031.gw.ulg.ac.be (139.165.192.49) 0.734 ms 0.612 ms 0.597 ms
 4 fe.m20.access.liege.belnet.net (193.191.10.17) 0.791 ms 0.707 ms 0.713 ms
 5 oc48.m160.core.science.belnet.net (193.191.1.185) 2.266 ms 2.239 ms 16.679 ms
 6 oc192.m160.ext.science.belnet.net (193.191.1.2) 2.252 ms 2.260 ms 2.235 ms
 7 216.239.43.88 7.954 ms 7.922 ms 7.731 ms
 8 64.233.175.249 14.932 ms 14.906 ms 14.911 ms
 9 216.239.46.49 14.628 ms 14.431 ms 14.407 ms
10 66.249.85.99 14.409 ms 14.722 ms 14.624 ms
```

Le problème général sous-jacent est donc de déterminer, dans un graphe donné, le chemin le plus court permettant de relier deux sommets quelconques d'un graphe. On pourra envisager des variantes pondérées ou orientées. Dans la version pondérée, on recherchera le chemin de poids minimal.

Exemple I.3.11 (Planarité). Lors de l'élaboration de circuits électriques, les connexions reliant les différents composants ne peuvent se toucher sous peine de court-circuit. La question de théorie des graphes qui est posée est donc de déterminer si, dans le plan, il existe une configuration géométrique des sommets et des arcs du graphe assurant qu'aucune paire d'arcs ne se

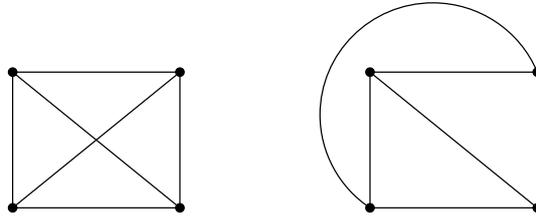


FIGURE I.20. Un graphe planaire.

croise. Si tel est le cas, on parle de graphe *planaire*. Plutôt que de représenter un graphe dans un plan, on pourrait aussi imaginer des représentations sur une sphère ou un tore et se poser la même question. Ainsi, le graphe $K_{3,3}$ de la figure I.21 n'est pas planaire (cf. Lemme III.4.2), mais si on le représente sur un tore, on peut obtenir une représentation dont les arcs ne se coupent pas (cf. figure I.22). Notons enfin que, dans l'élaboration des circuits

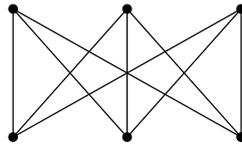
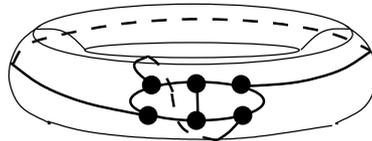


FIGURE I.21. Un graphe non planaire.

FIGURE I.22. $K_{3,3}$ sur un tore.

imprimés (VLSI⁸ par exemple), il est possible de réaliser de tels circuits sur plusieurs couches. Ainsi, un autre problème serait de déterminer le nombre minimum de couches à envisager pour la conception du circuit imprimé.

Remarque I.3.12. Le graphe de la figure I.21 est classiquement illustré par le *problème des trois villas et des trois usines*. Imaginez que trois villas doivent être reliées au gaz, à l'eau et à l'électricité. Est-il possible de construire des canalisations alimentant chaque villa depuis chaque usine de manière telle que ces canalisations ne se chevauchent pas ?

Exemple I.3.13 (Graphe d'intervalles). Considérons les intervalles ouverts $]0, 2[$, $]1, 4[$, $]2, 5[$, $]3, 4[$, $]3, 8[$ et $]6, 8[$. A chaque intervalle correspond un sommet d'un graphe non orienté. Si $]a, b[\cap]c, d[\neq \emptyset$, alors une arête relie dans le graphe les sommets correspondant à ces deux intervalles. Le graphe

⁸Very Large Scale Integration

d'intervalles correspondant aux intervalles donnés ci-dessus est représenté à la figure I.23. Ces graphes sont parfois utilisés en archéologie ou encore en

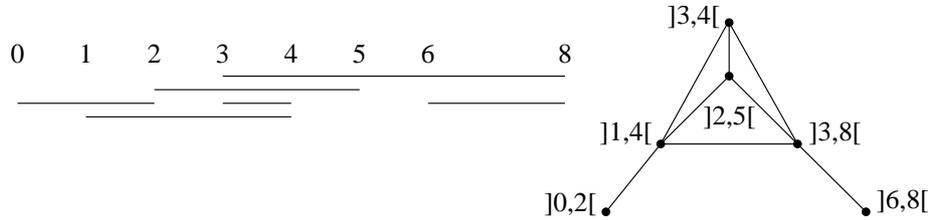


FIGURE I.23. Graphe d'intervalles.

génétique pour exprimer ou mettre en évidence des éléments communs se produisant à travers le temps (par exemple, des caractéristiques communes d'une période de l'histoire ou des mutations génétiques au sein du génome).

Exemple I.3.14 (Problème d'affectation). Soient O_1, \dots, O_k des ouvriers et T_1, \dots, T_t des postes de travail. Chaque ouvrier O_i possède certaines qualifications lui permettant de travailler sur certains postes $T_{i,1}, \dots, T_{i,d_i}$. Comment répartir les ouvriers pour que chaque poste de travail soit occupé par au moins un ouvrier ? Pour modéliser ce problème, on utilise un graphe biparti dont les sommets représentent les ouvriers et les postes. On trace un arc entre O_i et T_j si O_i possède la qualification pour travailler au poste T_j .

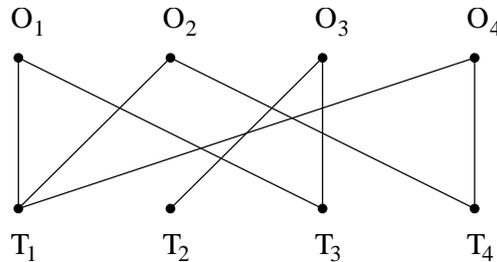


FIGURE I.24. Problème d'affectation.

Exemple I.3.15 (Tri topologique). Dans la majorité des exemples décrits précédemment, nous avons rencontré des graphes non orientés. Un exemple de graphe orienté est le suivant. On rencontre parfois des problèmes pour lesquels on recherche un ordre acceptable dans l'ordonnancement de tâches dépendant les unes des autres. On peut par exemple penser à la fabrication d'une voiture sur une chaîne de montage : on peut monter de façon indépendante le moteur et la carrosserie, avant de les assembler. Un autre exemple provient des choix de cours réalisés par des étudiants. En effet, certains prérequis sont parfois nécessaires. On considérera un graphe dont les sommets sont les tâches (resp. les cours) à réaliser (resp. à suivre) et si une tâche (resp. un cours) doit être réalisée (resp. suivi) avant une autre (resp. un autre), on tracera un arc orienté entre les deux sommets

correspondant. La question sous-jacente est de *déterminer une indexation des sommets d'un graphe orienté sans cycle de manière telle que s'il existe un arc de v_i à v_j , alors $i < j$.*

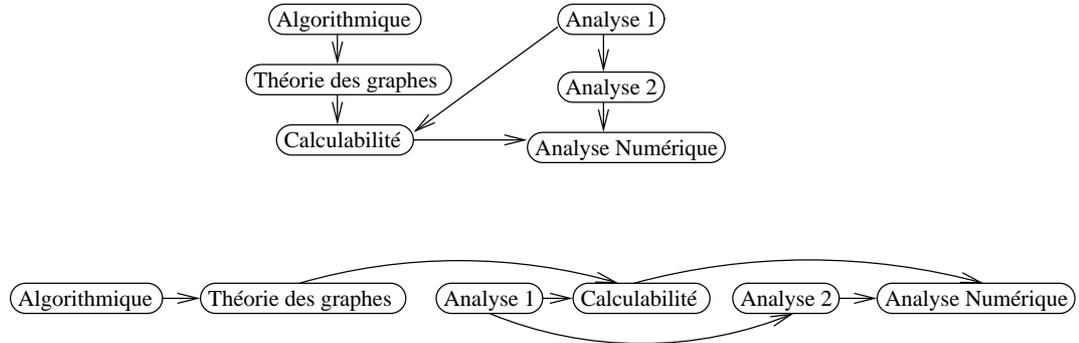


FIGURE I.25. Une application du tri topologique.

Exemple I.3.16 (Tournoi). On imagine un ensemble d'équipes ou de joueurs et une compétition où chaque joueur affronte tout autre joueur exactement une fois. Le seul résultat possible est la victoire ou la défaite. On peut alors considérer un graphe dont les sommets sont les joueurs et un arc relie le joueur i au joueur j si i a battu j lors de leur confrontation directe. La question naturelle qui se pose est alors d'essayer de déterminer un vainqueur pour la compétition.

Exemple I.3.17 (Graphe de De Bruijn). En combinatoire des mots, on étudie entre autres, les mots infinis (i.e., les suites infinies $w : \mathbb{N} \rightarrow \Sigma$ dont les éléments sont à valeurs dans un ensemble fini Σ). Le *graphe de De Bruijn* d'ordre k du mot w a pour sommets les facteurs de longueur k de w (i.e., les mots finis de la forme $w_i \cdots w_{i+k-1}$) et on dispose d'un arc de label $\sigma \in \Sigma$ entre les sommets τx et $x\sigma$ si et seulement si $\tau x \sigma$ est un facteur de w de longueur $k + 1$. La figure I.26 représente le graphe de De Bruijn d'ordre 3 du mot périodique $ababab \cdots$. Ces graphes sont en relation directe avec

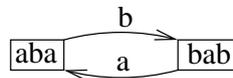


FIGURE I.26. Le graphe de De Bruijn d'ordre 3 de $ababab \cdots$.

certaines propriétés combinatoires des mots correspondants. Par exemple, un mot infini est ultimement périodique (i.e., il existe des mots finis u et v tels que $w = uvvv \cdots$) si et seulement si, pour tout k suffisamment grand, son graphe de De Bruijn d'ordre k contient un unique cycle dont tous les sommets ont un demi-degré sortant égal à 1.

Exemple I.3.18 (Flot). Une société hydro-électrique dispose d'un ensemble de canalisations interconnectées de divers diamètres et désire acheminer

une quantité d'eau maximale d'un point A à un point B . Quelle quantité d'eau peut effectivement être acheminée et comment l'eau est-elle distribuée le long du réseau de canalisations ? On peut imaginer que des contraintes techniques imposent en outre que chaque canalisation ait un débit maximum et minimum. Un tel exemple est représenté à la figure I.27 (les bornes de chaque canalisation sont représentées par un intervalle, une solution optimale est représentée par un nombre associé à chaque arc). On peut également

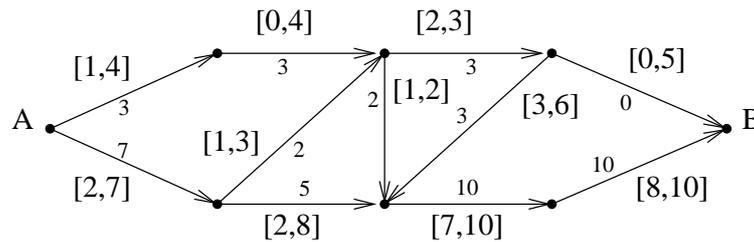


FIGURE I.27. Un problème de flot maximum.

prendre en compte une version pondérée où les coûts d'entretien des canalisations pourraient varier. On voudrait alors disposer d'un flot maximum pour un coût minimum.

Exemple I.3.19 (Chemin critique). Un entrepreneur chargé de construire une maison doit planifier l'accomplissement de plusieurs tâches :

- A: Creusage des fondations
- B: Construction du gros-oeuvre
- C: Installation électrique
- D: Installation du chauffage central
- E: Réalisation des peintures extérieures
- F: Réalisation des peintures intérieures

Ces diverses opérations sont soumises à des contraintes temporelles. Certaines tâches ne peuvent débuter avant que d'autres ne soient achevées; par contre, certaines peuvent aussi être réalisées en parallèle. Par exemple, les fondations doivent être creusées avant de débuter le gros oeuvre. De même, l'installation électrique doit être achevée avant de débuter les peintures intérieures. Par contre, on peut effectuer simultanément les peintures extérieures et l'installation du chauffage. On considère un graphe dont les sommets correspondent à des instants marquant la fin de certaines étapes et le début d'autres. Les arcs sont pondérés par le temps nécessaire à la réalisation d'une tâche. On obtient pour notre exemple, le graphe représenté à la figure I.28 où 0 représente le début des travaux, 1 le début du gros-oeuvre, 2 le début de l'installation électrique, 3 celui du chauffage, 4 celui de la peinture extérieure, 5 celui de la peinture intérieure et enfin, 6 représente la fin des travaux. Sur notre exemple, on voit des pondérations différentes au sortir du sommet 1. Cela peut par exemple signifier que la tâche 2 peut débuter sans que le gros-oeuvre ne soit totalement achevé (par contre, pour

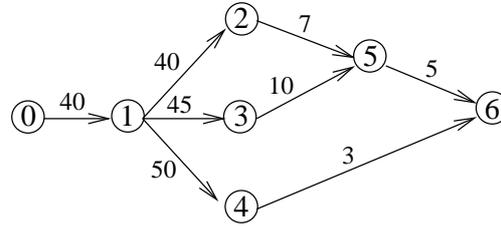


FIGURE I.28. Chemin critique dans la planification de tâches.

débuter 4, le gros oeuvre doit être entièrement terminé, ce qui explique une pondération plus importante). Se dégage alors la notion de *chemin critique*, chemin de poids maximal entre le début et la fin des travaux. En effet, pour achever la maison, toutes les tâches auront été entièrement exécutées et par conséquent, il faudra également emprunter le chemin correspondant à la réalisation des tâches les plus lentes. Ce chemin critique permet donc de déterminer le temps minimum nécessaire à l'achèvement des travaux.

4. Chemins et circuits

Les définitions suivantes sont somme toute assez naturelles et intuitives, mais il faut bien les préciser au moins une fois de manière rigoureuse pour savoir de quoi on parle exactement.

Définition I.4.1. Soit $G = (V, E)$ un multi-graphe non orienté⁹. Un *chemin* de longueur $k \geq 1$ est une suite ordonnée (e_1, \dots, e_k) de k arêtes adjacentes $e_i = \{e_{i,1}, e_{i,2}\}$, i.e., pour tous $i \in \{1, \dots, k-1\}$, $e_{i,2} = e_{i+1,1}$. Ce chemin de longueur k *joint* les sommets $e_{1,1}$ et $e_{k,2}$. On dit que le chemin (e_1, \dots, e_k) *pass*e par les arêtes e_1, \dots, e_k (resp. par les sommets $e_{1,1}, e_{2,1}, \dots, e_{k,1}$ et $e_{k,2}$). On supposera qu'un chemin de longueur 0 (correspondant à la suite vide) joint toujours un sommet à lui-même.

Si les extrémités du chemin sont égales, i.e., si $e_{1,1} = e_{k,2}$, on parle plutôt de *cycle*, de *circuit* ou encore de *chemin fermé*. Si on désire préciser que le chemin considéré n'est pas un cycle, on parlera de *chemin ouvert*.

Il se peut que les arêtes d'un chemin soient toutes distinctes (cela n'implique pas que les sommets du chemin soient tous distincts). On parle alors de *piste* ou de chemin *élémentaire* (voir par exemple, la figure I.29).

Si les arêtes d'un chemin sont toutes distinctes **et** si de plus, les sommets sont tous distincts¹⁰, on parle alors de chemin *simple*.

Bien sûr, les circuits étant des chemins particuliers, on parle aussi de circuit élémentaire ou simple (voir par exemple, la figure I.30). Evidemment, dans la définition d'un circuit simple, on admet que les sommets $e_{1,1}$ de départ et $e_{k,2}$ d'arrivée puissent être égaux, mais seulement eux. Il est clair

⁹Ce graphe peut avoir ou non des boucles, peu importe.

¹⁰Sauf dans le cas pathologique du circuit $(\{a, b\}, \{b, a\})$, demander que les sommets d'un chemin soient tous distincts (mis à part les extrémités du chemin dans le cas d'un circuit), implique que les arêtes sont aussi toutes distinctes.

Une arête peut être répétée plus d'une fois.

On a toujours $a \sim a$.

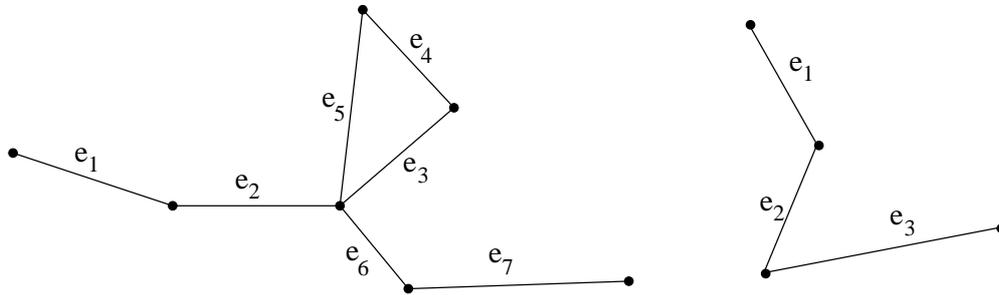


FIGURE I.29. Une piste (ou chemin élémentaire) et un chemin simple

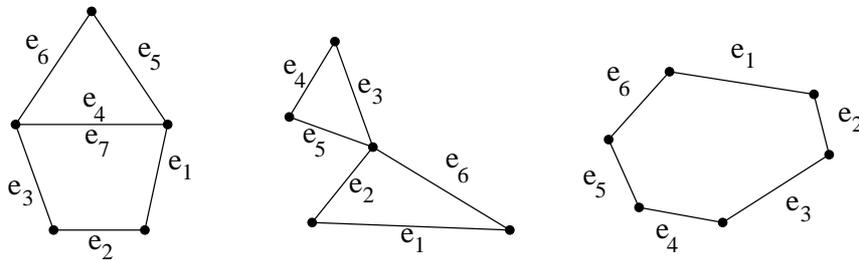


FIGURE I.30. Un circuit, un circuit élémentaire (ou piste fermée) et un circuit simple.

que tout circuit peut se décomposer en circuits simples.

Remarque I.4.2. Suivant les auteurs, la terminologie peut changer énormément. Ainsi, il n'est pas rare (par exemple, pour C. Berge), d'inverser les définitions des adjectifs *simple* et *élémentaire*. Pour d'autres, la notion de cycle contient de plus le caractère *élémentaire*, voire *simple*.

Remarque I.4.3. Dans le cas d'un graphe (qui n'est pas un multi-graphe), c'est-à-dire pour un 1-graphe, un chemin est aussi univoquement déterminé par une suite de sommets (v_1, \dots, v_k) de manière telle que $\{v_i, v_{i+1}\}$ est une arête du graphe.

Définition I.4.4. Deux sommets a et b sont *connectés* s'il existe un chemin les joignant, ce que l'on notera $a \sim b$. La relation \sim "être connecté" est une relation d'équivalence sur V . Une classe d'équivalence pour \sim est une *composante connexe* de G . Un multi-graphe non orienté est *connexe* si V/\sim contient une seule classe d'équivalence, i.e., G possède une seule composante connexe. Autrement dit, un multi-graphe non orienté est connexe si, pour toute paire de sommets, il existe un chemin les joignant. On supposera de plus que $G = (\{v\}, \emptyset)$ est connexe (ce qui revient à supposer qu'un chemin de longueur 0 joint toujours un sommet à lui-même).

Définition I.4.5. Soit $G = (V, E)$ un multi-graphe non orienté connexe¹¹. La *distance*¹² entre deux sommets a et b est la longueur du plus court chemin joignant a et b . On la note $d(a, b)$. Le *diamètre* de G est défini par

$$\text{diam}(G) = \max_{a, b \in V} d(a, b).$$

Si G est en outre pondéré par la fonction de poids $f : E \rightarrow \mathbb{R}^+$, la distance entre les sommets a et b est égale au poids minimal des chemins joignant a et b , i.e.,

$$d(a, b) = \min_{\substack{\text{chemin } (e_1, \dots, e_t) \\ \text{joignant } a \text{ et } b}} \sum_{i=1}^t f(e_i).$$

Les définitions données précédemment s'adaptent aisément au cas d'un multi-graphe orienté. Il suffit de respecter en plus le sens de parcours imposé par les arcs. Donnons quelques précisions.

Définition I.4.6. Soit $G = (V, E)$ un multi-graphe *orienté*¹³. Un *chemin* de longueur $k \geq 1$ est une suite ordonnée (v_1, \dots, v_k) de k arcs $v_i = (v_{i,1}, v_{i,2})$ tels que pour tous $i \in \{1, \dots, k-1\}$, $v_{i,2} = v_{i+1,1}$. Ce chemin de longueur k *joint* les sommets $v_{1,1}$ et $v_{k,2}$.

S'il existe un chemin joignant deux sommets a et b , on notera $a \rightarrow b$. Si $a \rightarrow b$ et $b \rightarrow a$, on dira que a et b sont *fortement connectés* et on notera $a \leftrightarrow b$. Si on impose à \leftrightarrow d'être réflexive (i.e., on suppose que $a \leftrightarrow a$), on vérifie aisément que la relation \leftrightarrow "être fortement connecté" est une relation d'équivalence sur V . Une classe d'équivalence pour \leftrightarrow est une *composante fortement connexe* (ou, plus court, *f. connexe*) de G . Si V/\leftrightarrow contient une seule classe, on dira que G est *fortement connexe* (ou *f. connexe*).

Les sommets appartenant à un cycle maximal, i.e., un cycle auquel on ne peut adjoindre de nouveaux sommets, constituent une composante *f. connexe*. Autrement dit, un multi-graphe orienté G est *f. connexe* si et seulement si il existe un cycle passant par chaque sommet de celui-ci.

Si on supprime l'orientation des arcs de G et si le multi-graphe non orienté obtenu de cette manière est connexe, alors on dira que G est *simplement connexe* (ou *s. connexe*). On pourra bien entendu définir, de manière évidente, les *composantes simplement connexes* (ou *s. connexe* de G).

Remarque I.4.7. Les notions de distance et de diamètre données dans le cas non orienté s'adaptent facilement au cas d'un multi-graphe orienté fortement connexe. On remarquera cependant qu'ici, la fonction $d(\cdot, \cdot)$ n'est en général pas symétrique¹⁴.

¹¹Si G n'était pas connexe, la fonction distance ne serait pas une fonction totale définie sur $V \times V$ tout entier.

¹²Vérifier qu'il s'agit effectivement d'une distance.

¹³Ce graphe peut avoir ou non des boucles, peu importe.

¹⁴Il ne peut donc pas s'agir à proprement parler d'une distance.

Exemple I.4.8. Considérons le multi-graphe orienté de la figure I.31 dont l'ensemble des sommets est $\{a, \dots, e\}$ et l'ensemble des arcs est $\{1, \dots, 7\}$. Ce graphe est simplement connexe mais il n'est pas fortement connexe. En

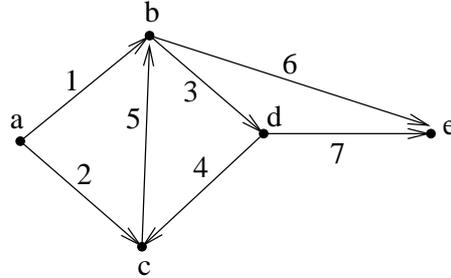


FIGURE I.31. Un graphe orienté.

effet, il n'existe pas de chemin joignant les sommets d à a . L'ensemble $\{b, c, d\}$ est une composante fortement connexe du graphe (les deux autres composantes sont $\{a\}$ et $\{e\}$). Un chemin est par exemple donné par $(1, 3, 7)$, un cycle par $(3, 4, 5)$ et une piste par $(1, 3, 4, 5, 6)$. La distance entre d et c vaut 1. Par contre, la distance entre c et d vaut 2.

4.1. Recherche du plus court chemin. Soit $G = (V, E)$ un digraphe pondéré par la fonction $p : E \rightarrow \mathbb{R}^+$. Nous allons présenter l'algorithme de Dijkstra de recherche d'un plus court chemin (i.e., un chemin de poids minimal) d'un sommet u fixé à un sommet quelconque de G . Il est clair que l'on peut restreindre ce problème au cas d'un digraphe simple¹⁵. Pour rappel, si u et v sont deux sommets tels que $u \rightarrow v$, le poids d'un chemin (e_1, \dots, e_k) les joignant est $\sum_i p(e_i)$. La distance de u à v est alors le poids minimal de tels chemins. Si $u \not\rightarrow v$, on n'a pas à proprement parler de distance¹⁶ et par convention, on posera que la "distance" de u à v est alors $+\infty$.

L'algorithme de Dijkstra s'applique également à un graphe non orienté. Il suffit de remplacer l'arête $\{u, v\}$ de poids α par deux arcs (u, v) et (v, u) ayant chacun un poids α et ainsi obtenir un digraphe.

Remarque I.4.9. Quitte à supposer p à valeurs dans $\mathbb{R}^+ \cup \{+\infty\}$, on étend p de E à $V \times V$ tout entier en posant $p(x, x) = 0$, pour tout $x \in V$ et $p(x, y) = +\infty$, si $(x, y) \notin E$. C'est ce que nous supposerons par la suite.

Intuitivement, l'algorithme fonctionne de la manière suivante. A chaque sommet v de G , on associe une valeur $T(v)$, initialisée à $p(u, v)$ et une liste de sommets $C(v)$ censée correspondre à un chemin de u à v . Lorsque

¹⁵En effet, passer par une boucle de poids $\alpha > 0$ ne ferait qu'augmenter inutilement le poids de ce chemin. De même, si plusieurs arcs joignent deux sommets, il suffit de conserver l'arc de poids minimal.

¹⁶Comme nous l'avons définie, la distance n'a de sens que sur des digraphes fortement connexes.

l'algorithme s'achève, $T(v)$ contient le poids minimal des chemins joignant u à v et $C(v)$ réalise un tel chemin (ou alors, $T(v) = +\infty$ si $u \not\rightarrow v$). L'idée est de construire de proche en proche un ensemble $X \subseteq V$ de manière telle qu'un chemin de poids minimal de u à $v \in X$ passe uniquement par des sommets de X . L'ensemble X est initialisé à $\{u\}$ et à chaque étape, on ajoute un sommet à l'ensemble.

Algorithme I.4.10 (Algorithme de Dijkstra). Les données sont un digraphe simple $G = (V, E)$ pondéré par une fonction $p : V \times V \rightarrow \mathbb{R}^+ \cup \{+\infty\}$ (cf. remarque I.4.9) et un sommet u .

```

Pour tout sommet  $v \in V$ ,  $T(v) := p(u, v)$ ,  $C(v) := (u, v)$ 
 $X := \{u\}$ 
Tant que  $X \neq V$ , répéter
  Choisir  $v \in V \setminus X$  tel que, pour tout  $y \in V \setminus X$ ,  $T(v) \leq T(y)$ 17
   $X := X \cup \{v\}$ 
  Pour tout  $y \in V \setminus X$ 
    Si  $T(y) > T(v) + p(v, y)$ , alors  $T(y) := T(v) + p(v, y)$  et  $C(y) := [C(v), y]$ 

```

Dans cet algorithme, la notation $[C(v), y]$ représente la liste $C(v)$ à laquelle on ajoute un élément y . Intuitivement, lorsqu'on ajoute un sommet v à X , on regarde s'il est avantageux pour les sommets y ne se trouvant pas dans X de passer par ce sommet v nouvellement ajouté à X . Si tel est le cas, on met à jour les informations concernant y .

Avant de démontrer l'exactitude de cet algorithme, donnons un exemple d'application de ce dernier.

Exemple I.4.11. Voici une application de l'algorithme de Dijkstra au graphe représenté à la figure I.32. Pour l'initialisation, prenons $X = \{a\}$

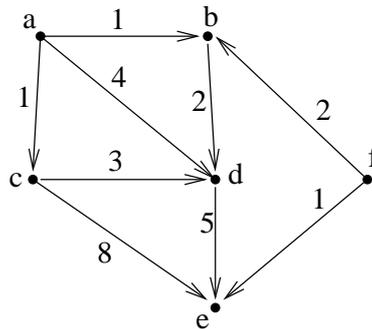


FIGURE I.32. Un digraphe simple pondéré.

et on a

v	a	b	c	d	e	f
$T(v)$	0	1	1	4	$+\infty$	$+\infty$
$C(v)$	(a, a)	(a, b)	(a, c)	(a, d)	(a, e)	(a, f)

¹⁷On suppose que $r < +\infty$, si $r \in \mathbb{R}$ et que $+\infty \leq +\infty$.

Si le sommet b est choisi (on a le choix entre b et c), on a $X = \{a, b\}$ et le tableau est mis à jour :

v	a	b	c	d	e	f
$T(v)$	0	1	1	3	$+\infty$	$+\infty$
$C(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, e)	(a, f)

A l'étape suivante, on est forcé de choisir c . Ainsi, $X = \{a, b, c\}$ et le tableau devient :

v	a	b	c	d	e	f
$T(v)$	0	1	1	3	9	$+\infty$
$C(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, c, e)	(a, f)

A présent, d est sélectionné, $X = \{a, b, c, d\}$ et la valeur de $C(e)$ peut être améliorée,

v	a	b	c	d	e	f
$T(v)$	0	1	1	3	8	$+\infty$
$C(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, b, d, e)	(a, f)

Ensuite, e et f sont choisis successivement mais le tableau ne change plus. Par exemple, on en conclut que le plus court chemin joignant a à e passe par b et d et son poids vaut 8.

Démonstration. Il est clair que l'algorithme s'achève toujours puisqu'à chaque itération de la boucle, un nouvel état est ajouté à X . Il nous suffit donc de vérifier qu'il s'achève avec le résultat attendu. Pour ce faire, on procède par récurrence sur $\#X$ et on vérifie les deux assertions suivantes pour toutes les valeurs de $\#X$, d'où le résultat annoncé quand $X = V$.

- i) Pour tout $v \in X$, $T(v)$ est le poids minimal de tous les chemins joignant u à v .
- ii) Pour tout $v \notin X$, $T(v)$ est le poids minimal des chemins joignant u à v qui, à l'exception de v , passent uniquement par des sommets de X .

Pour $\#X = 1$, c'est immédiat car $X = \{u\}$ et l'initialisation effectuée dans l'algorithme correspond aux assertions. Supposons le résultat vérifié pour $\#X = n$ et vérifions-le pour $\#X = n + 1$, $1 \leq n < \#V$.

Lorsque X contient n sommets, l'algorithme stipule de lui ajouter un sommet v ayant une valeur $T(v)$ minimale parmi les sommets n'appartenant pas à X . Par l'hypothèse de récurrence ii), puisqu'à ce stade $\#X = n$ et $v \notin X$, $T(v)$ est le poids minimal des chemins joignant u à v qui, à l'exception de v , passent uniquement par des sommets de X . Nous ajoutons à présent v à X pour obtenir un ensemble de taille $n + 1$. Procédons par l'absurde et supposons que $T(v)$ n'est pas le poids minimal des chemins joignant u à v (c'est-à-dire que l'assertion i) n'est pas vérifiée). Par conséquent, il existe un sommet $y \notin X$ et un chemin de u à v passant par y dont le poids est strictement inférieur à $T(v)$. De là, on en conclut que $T(y) < T(v)$, ce qui est impossible au vu du choix du sommet v au sein de l'algorithme.

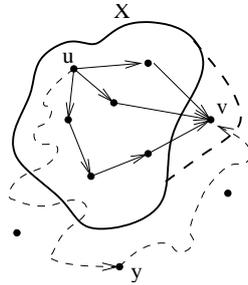


FIGURE I.33. Illustration de l'algorithme de Dijkstra.

Pour la seconde assertion, on procède de manière analogue en utilisant cette fois la dernière ligne de l'algorithme pour obtenir une contradiction. Plus précisément, passons d'un ensemble X à n sommets à un ensemble X' à $n + 1$ sommets en lui ajoutant un sommet v . Supposons à présent que ii) n'est plus satisfait et qu'il existe donc un sommet $y \notin X'$ tel que $T(y)$ est strictement supérieur au poids minimal des chemins joignant u à y qui, à l'exception de y , passent uniquement par des sommets de X' . Or, par hypothèse de récurrence, nous savons que l'assertion ii) était satisfaite pour $\#X = n$. Autrement dit, avant d'ajouter le sommet v , $T(y)$ était minimal

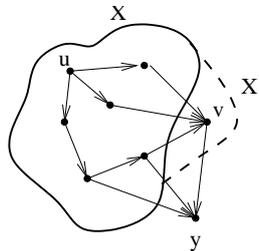


FIGURE I.34. Illustration de l'algorithme de Dijkstra.

pour les chemins joignant u à y qui, à l'exception de y , passent uniquement par des sommets de X . Ainsi, en ajoutant le sommet v à X , on aurait remplacé $T(y)$ par une valeur supérieure, ce qui est en contradiction avec les prescriptions de l'algorithme (à la dernière ligne, on spécifie de remplacer $T(y)$ uniquement lorsqu'il est préférable de passer par v).

■

Remarque I.4.12. On peut facilement voir que l'algorithme de Dijkstra a une complexité temporelle en $\mathcal{O}((\#V)^2)$. Avec une implémentation minutieuse, utilisant les listes d'adjacence et les files de priorité, on obtient même une complexité en $\mathcal{O}((\#E + \#V) \log \#V)$.

Remarque I.4.13. [16] Le routage des données entre les réseaux d'un internet est l'une des applications où les plus courts chemins jouent un rôle important. Le routage est le processus consistant à prendre des décisions sur la façon de déplacer

des données d'un point à un autre. Dans un internet, il est effectué en propageant de petites portions de données, les *paquets*, le long de points interconnectés, les *passerelles*. Lorsque chaque paquet passe par une passerelle, un *routeur* étudie où le paquet doit se rendre et décide alors de la passerelle suivante où l'envoyer. Le but de chaque routeur est de propager un paquet vers un point de plus en plus près de sa destination finale.

Afin de rapprocher un paquet de sa destination, chaque routeur gère des informations sur la structure, ou topologie, de l'internet. Ces informations sont stockées dans une *table de routage*, contenant une entrée pour chaque passerelle que le routeur sait atteindre. Chaque entrée indique la passerelle suivante à laquelle envoyer les paquets destinés à une autre passerelle que le routeur lui-même.

Pour que les paquets soient continuellement envoyés par la meilleure route possible, les routeurs mettent régulièrement à jour leurs tables de routage afin de tenir compte des changements dans l'internet. Dans l'un des types de routage, appelé *routage SPF (Shortest Path First)*, chaque routeur gère sa propre carte de l'internet afin de mettre à jour sa table de routage en calculant les plus courts chemins entre lui et les autres destinations. Sa carte est un graphe orienté et pondéré, dont les sommets sont les passerelles et les arcs les connexions entre celles-ci. Chaque arc est pondéré par les dernières performances constatées sur la connexion.

4.2. Graphes et chemins eulériens. La définition suivante est valable aussi bien dans le cas de graphes orientés que non orientés. Nous supposons à chaque fois qu'il sera question de problèmes eulériens être en présence de graphes connexes¹⁸.

Définition I.4.14. Un chemin (resp. un circuit) d'un multi-graphe G est *eulérien* s'il passe une et une seule fois par chaque arête/arc de G . (Un tel chemin (resp. circuit) peut bien évidemment passer plus d'une fois par un même sommet.) Autrement dit, un chemin (resp. un circuit) eulérien est une piste (resp. une piste fermée) passant par chaque arête/arc de G . Un *multi-graphe eulérien* est un graphe qui possède un circuit eulérien.

Remarque I.4.15. Dans le problème consistant à déterminer si un graphe G possède ou non un chemin eulérien, l'existence de boucles au sein de G n'a aucune importance. En effet, soient G un graphe et G' le graphe obtenu en supprimant les boucles de G . Il est évident que G possède un chemin ou un circuit eulérien si et seulement si G' en possède un.

Considérons le cas des multi-graphes finis non orientés. Il est évident que pour que G possède un circuit eulérien, i.e., pour que G soit eulérien, il est nécessaire que G soit connexe et que le degré de chaque sommet soit pair. Comme le montre le résultat suivant, ces conditions sont également suffisantes. On peut donc constater que le fait d'être eulérien est une propriété "locale" (elle ne fait intervenir que le degré de chaque sommet pris "isolément").

¹⁸On évite comme cela les pathologies de graphes possédant des points isolés.

Théorème I.4.16. *Un multi-graphe fini non orienté connexe $G = (V, E)$ possède un circuit eulérien si et seulement si le degré de chaque sommet est pair.*

Démonstration. Supposons donc que chaque sommet de G est de degré pair. Débutons la construction d'une piste avec un sommet a_1 de G . A chaque étape $i \geq 1$ de cette construction, on choisit un sommet a_{i+1} de manière telle qu'une arête $\{a_i, a_{i+1}\} \in E$ est sélectionnée parmi les $\#E - i + 1$ arêtes non déjà sélectionnées. Puisque chaque sommet est de degré pair, cette sélection est toujours possible ("lorsqu'on aboutit dans un sommet, on peut toujours en repartir"). Puisque le graphe est fini, cette procédure s'achève toujours.

On dispose alors d'une piste P joignant a_1 à un certain sommet a_ℓ . En fait, on peut supposer que cette piste est fermée, i.e., $a_\ell = a_1$. En effet, si a_ℓ diffère de a_1 , puisque le degré de chaque sommet est pair, on peut étendre la piste en ajoutant une arête $\{a_\ell, a_{\ell+1}\}$. En continuant de la sorte¹⁹, on épuise les sommets jusqu'à revenir en a_1 .

Si la piste fermée P est un circuit eulérien, le théorème est démontré. Sinon, il existe un sommet b de P qui est l'extrémité d'un nombre pair d'arêtes n'apparaissant pas dans P . (Une illustration est donnée à la figure I.35. Depuis b , il est possible de construire une piste fermée Q formée

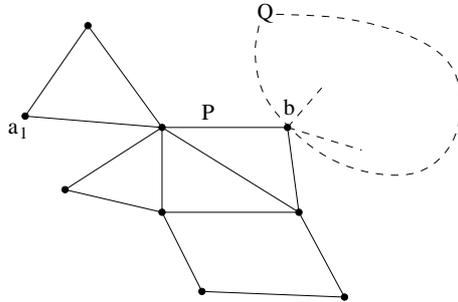


FIGURE I.35. Construction d'un circuit eulérien.

uniquement d'arêtes n'apparaissant pas dans P . (On utilise la même procédure que précédemment; il est clair que le degré de chaque sommet est encore pair.) De cette façon, nous avons étendu la piste P en une piste plus longue $P \cup Q$ (couvrant un plus grand nombre d'arêtes). On obtient alors un circuit eulérien en répétant cette étape un nombre fini de fois. ■

Corollaire I.4.17. *Le problème des sept ponts de Königsberg donné dans l'exemple I.3.1 n'admet pas de solution.*

¹⁹Procédons par l'absurde et imaginons un instant ne jamais revenir en a_1 . A chaque sélection d'une nouvelle arête depuis a_ℓ , le sommet atteint diffère donc de a_1 et une des arêtes restantes est consommée. Ainsi, le nombre d'arêtes disponibles décroît strictement. Pourtant, chaque sommet étant de degré pair, on peut continuer de la sorte jusqu'à ce qu'il ne reste, dans le pire cas, qu'une seule arête disponible, celle retournant à a_1 .

Démonstration. C'est immédiat, le graphe de la figure I.12 possède un sommet de degré 5 et 3 sommets de degré 3. ■

Corollaire I.4.18. *Un multi-graphe non orienté connexe possède un chemin eulérien joignant deux sommets a et b si et seulement si a et b sont les deux seuls sommets de degré impair.*

Démonstration. Pour se ramener au théorème précédent, il suffit de considérer le graphe G auquel on ajoute une arête supplémentaire joignant les sommets a et b . ■

Exemple I.4.19. Fort des résultats précédents, on peut par exemple répondre à la question : “le dessin suivant peut-il ou non être tracé d'un seul trait, sans lever le crayon et sans repasser deux fois par un trait déjà tracé?”. C'est une simple application du corollaire précédent²⁰.

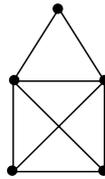


FIGURE I.36. Une maison à tracer d'un seul trait.

Nous présentons succinctement l'algorithme de Fleury permettant de construire un circuit eulérien (à supposer qu'un tel circuit existe). Cet algorithme repose sur la notion d'arête de coupure présentée à la section 6. Nous conseillons donc au lecteur de passer cet algorithme en première lecture.

Algorithme I.4.20 (Fleury). La donnée fournie à cet algorithme est un graphe simple eulérien.

Choisir un sommet $v_0 \in V$

$i := 1$

Répéter tant que possible

Choisir une arête $e_i = \{v_{i-1}, v_i\} \in V$ telle que

- ▶ e_i diffère des arêtes déjà choisies e_1, \dots, e_{i-1}
- ▶ autant que possible, e_i ne doit pas être une arête de coupure de $G_i = G - \{e_1, \dots, e_{i-1}\}$

$i := i + 1$

Cet algorithme fournit une suite d'arêtes e_1, e_2, \dots qui constituent un circuit eulérien.

²⁰Il est clair que pour trouver la solution à ce problème, il faut débiter son dessin par un des deux coins inférieurs. Ce sont les deux seuls sommets de degré impair.

Le cas des multi-graphes orientés se traite de manière analogue. Les preuves sont laissées à titre d'exercices.

Théorème I.4.21. *Un multi-graphe fini orienté simplement connexe $G = (V, E)$ possède un circuit eulérien si et seulement si le demi-degré entrant de chaque sommet est égal à son demi-degré sortant.*

Démonstration. Laisée à titre d'exercices. On adaptera facilement le raisonnement de la preuve du théorème I.4.16. ■

Corollaire I.4.22. *Un multi-graphe fini orienté connexe $G = (V, E)$ possède un chemin eulérien si et seulement si il existe deux sommets v_0 et v_1 tel que*

- ▶ pour tout $v \in V \setminus \{v_0, v_1\}$, $d^-(v) = d^+(v)$,
- ▶ $d^+(v_0) = d^-(v_0) + 1$,
- ▶ $d^-(v_1) = d^+(v_1) + 1$.

Démonstration. Laisser à titre d'exercices. ■

4.3. Connexité des graphes non orientés. Nous venons de voir qu'une condition nécessaire pour qu'un graphe soit eulérien est qu'il soit connexe. Nous allons donc fournir un algorithme naïf permettant de décider si un graphe non orienté est connexe. Il est clair que lorsqu'on s'intéresse à cette question, il suffit de considérer des graphes simples. (En effet, l'existence de boucles ou d'arêtes multiples n'a pas d'incidence sur la connexité du graphe.)

Algorithme I.4.23. La donnée fournie à cet algorithme est un graphe non orienté $G = (V, E)$, l'ensemble des sorties possibles est $\{\text{oui}, \text{non}\}$.

```

Choisir au hasard un sommet  $v_0 \in V$ 
Composante :=  $\{v_0\}$ , New :=  $\{v_0\}$ 
Tant que New  $\neq \emptyset$ 
  Voisins :=  $\emptyset$ 
  pour tout sommet  $v$  appartenant à New
    Voisins := Voisins  $\cup \nu(v)$ 
  New := Voisins  $\setminus$  Composante
  Composante := Composante  $\cup$  New
Si Composante =  $V$ 
  alors sortie : "oui,  $G$  connexe"
  sinon sortie : "non,  $G$  non connexe"

```

Il s'agit d'un algorithme "tache d'huile" ou "glouton" (on détermine de proche en proche, les sommets accessibles depuis le sommet v_0). La variable **Composante** est destinée à contenir la composante connexe du sommet v_0 . La variable **New** contient les sommets qui, à l'étape en cours d'exécution, sont

nouvellement déterminés comme appartenant à la composante connexe. Il s'agit donc de deux variables de type ensembliste. Lorsqu'il n'y a plus de nouveaux sommets à ajouter à la liste des sommets appartenant à la composante connexe, l'algorithme s'achève. Il reste alors à déterminer si cette composante connexe contient ou non tous les sommets de G . En particulier, à la fin de cet algorithme, la variable `Composante` est égale à la composante connexe de v_0 .

L'emploi de cet algorithme est facilité par la construction préalable d'un dictionnaire des voisins. Il s'agit d'une liste associant à chaque sommet v , l'ensemble $\nu(v)$ de ses voisins.

Exemple I.4.24. Considérons le graphe simple représenté à la figure I.37 et appliquons-lui l'algorithme I.4.23. Les résultats de l'algorithme se trouvent dans le tableau I.2. Le tableau I.1 donne le dictionnaire des voisins

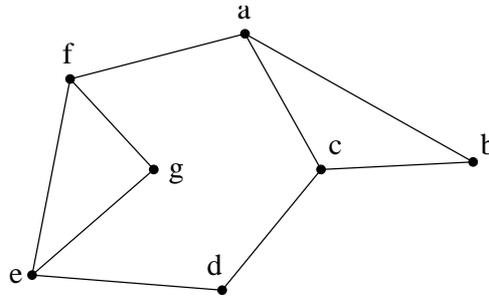


FIGURE I.37. Un graphe simple (connexe).

construit une fois pour toutes. Le tableau I.2 montre l'évolution des vari-

v	$\nu(v)$
a	$\{b, c, f\}$
b	$\{a, c\}$
c	$\{a, b, d\}$
d	$\{c, e\}$
e	$\{d, f, g\}$
f	$\{a, e, g\}$
g	$\{e, f\}$

TABLE I.1. Dictionnaire des voisins.

ables apparaissant dans l'algorithme avec comme choix initial, $v_0 = c$.

4.4. Décomposition en composantes fortement connexes. La détermination de la composante fortement connexe contenant un sommet donné v_0 s'inspire de l'algorithme I.4.23. Il est clair que, pour ce problème, il suffit de considérer le cas de graphes orientés simples. Déterminer si un graphe est connexe, autrement dit s'il possède une seule composante connexe, est l'une des deux étapes permettant de décider si un graphe donné possède

Composante	New	Voisins
$\{c\}$	$\{c\}$	\emptyset $\nu(c) = \{a, b, d\}$
$\{c\} \cup \{a, b, d\}$ $= \{a, b, c, d\}$	$\{a, b, d\} \setminus \{c\}$ $= \{a, b, d\}$	\emptyset $\nu(a) \cup \nu(b) \cup \nu(d)$ $= \{a, b, c, e, f\}$
$\{a, b, c, d\} \cup \{e, f\}$ $= \{a, b, c, d, e, f\}$	$\{a, b, c, e, f\} \setminus \{a, b, c, d\}$ $= \{e, f\}$	\emptyset $\nu(e) \cup \nu(f)$ $= \{a, d, e, f, g\}$
$\{a, b, c, d, e, f\} \cup \{g\}$ $= \{a, b, c, d, e, f, g\}$	$\{a, d, e, f, g\} \setminus \{a, b, c, d, e, f\}$ $= \{g\}$	\emptyset $\nu(g) = \{e, f\}$
$\{a, b, c, d, e, f, g\} \cup \{e, f\}$ $= \{a, b, c, d, e, f, g\}$	$\{e, f\} \setminus \{a, b, c, d, e, f, g\}$ $= \emptyset$	

TABLE I.2. Application de l'algorithme de test de connexité.

ou non une structure d'arbre (comme nous le verrons, les arbres sont particulièrement importants à bien des égards).

Définition I.4.25. La clôture réflexive et transitive de l'application succ est définie comme suit,

$$\text{succ}^*(v) := \bigcup_{j=0}^{\infty} \text{succ}^j(v) \quad \text{où} \quad \begin{cases} \text{succ}^0(v) = v \\ \text{succ}^{j+1}(v) = \text{succ}(\text{succ}^j(v)) \end{cases} .$$

Si le graphe G est fini, il est clair que

$$\text{succ}^*(v) = \bigcup_{j=0}^{\#E} \text{succ}^j(v).$$

De même, s'il existe $k < \#E$ tel que $\text{succ}^k(v) = \text{succ}^{k+1}(v)$, alors

$$\text{succ}^*(v) = \bigcup_{j=0}^k \text{succ}^j(v).$$

Avec les notations précédentes, pour tous $a, b \in V$,

$$a \rightarrow b \Leftrightarrow b \in \text{succ}^*(a).$$

Remarque I.4.26. On définit de manière semblable la clôture réflexive et transitive de l'application pred. On a bien évidemment, pour tous sommets a et b ,

$$b \rightarrow a \Leftrightarrow b \in \text{pred}^*(a).$$

Ainsi,

$$a \leftrightarrow b \Leftrightarrow b \in \text{succ}^*(a) \cap \text{pred}^*(a).$$

L'algorithme I.4.23 peut facilement être adapté pour calculer $\text{succ}^*(a)$ (resp. $\text{pred}^*(a)$). Il suffit principalement d'initialiser les variables **Composante** et **New** à $\{a\}$ et de remplacer $\nu(v)$ par $\text{succ}(v)$ (resp. $\text{pred}(v)$). En recherchant ainsi l'intersection des deux ensembles, on détermine la composante f. connexe du sommet a . Si cette composante est égale à l'ensemble des sommets, alors le graphe est f. connexe.

Définition I.4.27. Soit $G = (V, E)$ un graphe orienté. On construit un nouveau graphe orienté, appelé *graphe acyclique des composantes*²¹ ou encore *condensé* de G , dont les sommets sont les composantes f. connexes de G . Un arc joint deux composantes f. connexes A et B , s'il existe $a \in A$ et $b \in B$ tels que $a \rightarrow b$.

Remarque I.4.28. Bien évidemment, s'il existe $a \in A$ et $b \in B$ tels que $a \rightarrow b$, alors il n'existe aucun $a' \in A$ ni aucun $b' \in B$ tels que $b' \rightarrow a'$. En effet, si tel était le cas, $A \cup B$ serait alors une composante f. connexe de G . Ceci est impossible vu la maximalité des composantes connexes. D'une manière générale, le graphe des composantes ne contient pas de cycle.

Exemple I.4.29. La figure I.38 représente un digraphe et ses composantes f. connexes ainsi que le graphe des composantes correspondant.

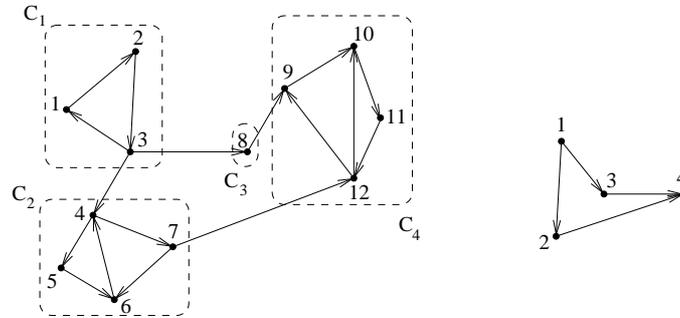


FIGURE I.38. digraphe et graphe des composantes.

Le résultat suivant donne un lien entre la simple connexité et la forte connexité.

Lemme I.4.30. Soit G un digraphe (simple) tel que pour tout $v \in V$, $d^+(v) = d^-(v)$. Alors, G est f. connexe si et seulement si il est s. connexe.

Démonstration. Il est clair que la f. connexité implique la s. connexité. Supposons que G est s. connexe, que pour tout v , $d^+(v) = d^-(v)$, mais que G n'est pas f. connexe. Notre but est d'obtenir une contradiction. Soient C_1, \dots, C_r les composantes f. connexes de G (où $r \geq 2$). Comme nous

²¹en anglais, (*directed acyclic collapse* ou, *acyclic component graph*).

l'avons déjà remarqué plus haut (remarque I.4.28), le condensé de G (ayant les C_i pour sommets) ne possède pas de cycle. On en conclut²² qu'il existe (au moins) une composante C_j telle que tout arc issu d'un sommet de C_j possède sa destination également dans C_j (dans l'exemple correspondant à la figure I.38, c'est le cas de la composante C_4). Par hypothèse, G est s. connexe. Il existe donc une autre composante C_i et un arc joignant un sommet x de C_i à un sommet y de C_j .

Dans le graphe $G' = (V', E')$ restreint à la composante C_j , i.e., dans le graphe où l'on considère uniquement les sommets de C_j et les arcs ayant leurs deux extrémités dans C_j , la "handshaking formula" est satisfaite, i.e.,

$$\sum_{v \in V'} d^-(v) = \sum_{v \in V'} d^+(v).$$

Il est sous-entendu que si le symbole de sommation fait référence à V' , alors on considère le graphe G' correspondant. Revenons au graphe G de départ. Dans celui-ci, tout arc ayant son origine dans C_j a aussi son extrémité dans C_j . Ainsi, considérer le graphe G dans son entièreté ou sa restriction G' à C_j ne modifie pas le demi-degré sortant des sommets de C_j et on a

$$\sum_{v \in C_j} d^+(v) = \sum_{v \in V'} d^+(v).$$

Par contre, y est un sommet de C_j et nous savons qu'au moins un arc n'ayant pas son origine dans C_j aboutit en y . Autrement dit, nous avons cette fois,

$$\sum_{v \in C_j} d^-(v) > \sum_{v \in V'} d^-(v).$$

De là, on en conclut que

$$\sum_{v \in C_j} d^-(v) > \sum_{v \in C_j} d^+(v)$$

ce qui contredit l'hypothèse selon laquelle $d^+(v) = d^-(v)$ pour tout $v \in V$ et donc que $\sum_{v \in C_j} d^-(v) = \sum_{v \in C_j} d^+(v)$. ■

Une preuve alternative bien plus courte mais exploitant un résultat précédent. En fait, la preuve ci-dessus reprend les mêmes raisonnements que ceux développés dans la preuve du théorème I.4.16.

Démonstration. Supposons le graphe s. connexe. Puisqu'il est connexe et que pour tout $v \in V$, $d^+(v) = d^-(v)$, on peut appliquer le théorème I.4.21. On conclut directement que G possède un circuit eulérien. Il est donc f. connexe. ■

²²Pour une démonstration complète, cf. lemme I.8.1.

5. Sous-graphes

Dans la preuve du lemme I.4.30, nous avons considéré la restriction d'un graphe à une de ses composantes. Dans cette section, nous formalisons ce type de construction. Grosso-modo, un sous-graphe d'un graphe donné est un graphe obtenu en supprimant certains sommets et/ou arêtes.

Définition I.5.1. Soient $G = (V, E)$ et $G' = (V', E')$ deux graphes (orientés ou non). Le graphe G' est un *sous-graphe* de G si

- ▶ $V' \subseteq V$,
- ▶ $E' \subseteq E \cap (V' \times V')$.

Ainsi, G' est un sous-graphe de G s'il est obtenu en enlevant à G certains sommets et/ou certains arcs ou arêtes. En particulier, si on enlève un sommet v de G , il faut nécessairement enlever tous les arcs (ou arêtes) incidents à v . Par contre, pour construire G' , on peut très bien enlever un arc (ou une arête) de G sans pour autant enlever le moindre sommet de G . Le graphe G' est un *sous-graphe propre* de G si $E' \subsetneq E$ ou $V' \subsetneq V$. Dans le premier

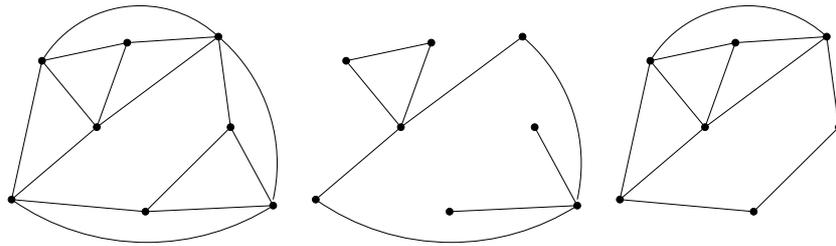


FIGURE I.39. Un graphe et deux sous-graphes.

sous-graphe de la figure I.39, on a enlevé uniquement certaines arêtes. Dans le second, on a enlevé un sommet et les arêtes adjacentes.

Soient $v \in V$ et $e \in E$. On note $G - e$ (resp. $G - v$) le sous-graphe G' de G obtenu en supprimant l'arc (ou l'arête) e (resp. le sous-graphe G' obtenu en supprimant le sommet v et les arcs (ou les arêtes) adjacents). Par analogie, on notera $G = G' + e$ (resp. $G = G' + v$), le graphe obtenu par adjonction à G' d'une arête ou d'un sommet.

On peut bien évidemment étendre ces notations à un ensemble fini de sommets. Ainsi, si $W = \{v_1, \dots, v_k\} \subseteq V$, alors $G - W$ est le sous-graphe

$$(\dots((G - v_1) - v_2) \dots - v_{k-1}) - v_k := G - v_1 - \dots - v_k.$$

On procède de même pour un ensemble fini d'arcs (ou d'arêtes) et on introduit une notation $G - F$ pour un sous-ensemble F de E .

Soit $W \subseteq V$. Le sous-graphe de G induit par W est le sous-graphe $G' = (W, E')$ avec $E' = E \cap (W \times W)$.

Si $W \subseteq V$ est tel que le sous-graphe induit par W ne contient aucune arête, alors les sommets de W sont dits *indépendants*. Le nombre maximal de sommets indépendants de G est noté $\alpha(G)$.

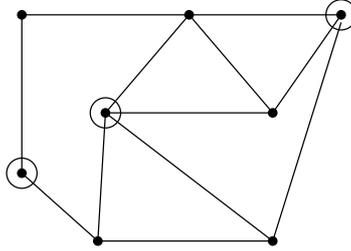


FIGURE I.40. Des sommets indépendants.

Définition I.5.2. Soient $G = (V, E)$ un graphe et $G' = (V', E')$ un de ses sous-graphes. On dit que G' est un *sous-graphe couvrant* G , si $V' = V$ et si

$$\forall v \in V, \exists z \in V : \{z, v\} \in E'.$$

On dira que E' est une couverture (des sommets) de G . Autrement dit, tout sommet de G est une extrémité d'une arête de E' .

Comme nous le verrons bientôt, on s'intéressera en particulier aux sous-graphes couvrants qui sont des arbres. Dans ce cas, on parlera naturellement de *sous-arbre couvrant*.

Exemple I.5.3. Le premier sous-graphe de la figure I.39 est couvrant. Un exemple de sous-arbre couvrant a été donné à l'exemple I.3.8.

6. Coupes, points d'articulation, k -connexité

Certains sommets ou arêtes d'un graphe (ou d'une composante) connexe jouent un rôle particulier : les enlever rendrait le graphe non connexe. D'un point de vue pratique (par exemple, pour un réseau électrique ou de distribution d'eau, ou encore pour l'Internet), il s'agit de composants cruciaux du réseau. En effet, une panne localisée en un tel endroit priverait par exemple toute une région de ressources peut-être vitales.

Définition I.6.1. Soit $H = (V, E)$ un multi-graphe²³ non orienté connexe (ou une composante connexe d'un multi-graphe non orienté). Le sommet v est un *point d'articulation*, si $H - v$ n'est plus connexe, ou réduit à un point. D'une manière générale, on dit aussi que v est un point d'articulation d'un multi-graphe H si $H - v$ contient plus de composantes connexes que H (ou est réduit à un point).

Si H est connexe et ne contient aucun point d'articulation, alors on dira que H est *au moins 2-connexe*.

Définition I.6.2. On peut étendre la notion de point d'articulation de la manière suivante. Un *ensemble d'articulation* est un ensemble W de sommets du multi-graphe connexe $H = (V, E)$ qui est tel que $H - W$ n'est plus

²³Lorsqu'on s'intéresse à des questions de connexité, il est souvent inutile de considérer autre chose que des graphes simples. Cependant, les notions définies ici restent valables dans le cadre général des multi-graphes.

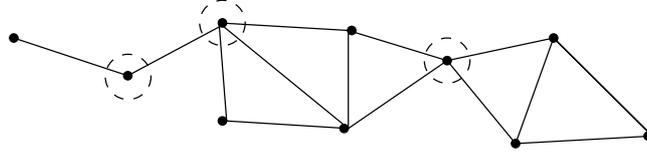


FIGURE I.41. Un graphe et ses points d'articulation.

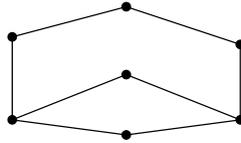


FIGURE I.42. Un graphe au moins 2-connexe.

connexe (ou réduit à un sommet). D'une manière générale, pour un multigraphe H quelconque, on dit aussi que W est un ensemble d'articulation si $H - W$ contient plus de composantes connexes que H .

Définition I.6.3. Pour un graphe connexe H , on note $\kappa(H)$ la taille minimale d'un ensemble d'articulation de H ,

$$\kappa(H) = \min\{\#W \mid W \subseteq V : H - W \text{ disconnecté ou réduit à un sommet}\}.$$

En particulier, on a que $\kappa(K_n) = n - 1$. Pour un graphe G non connexe, on pose $\kappa(G) = 0$. Si $\kappa(H) = k \geq 1$, alors on dit que H est k -connexe. Ainsi, dans un graphe connexe G , $\kappa(G) = k$ signifie que quels que soient les $k - 1$ sommets supprimés, G reste connexe mais il est possible de supprimer k sommets bien choisis pour disconnecter G (ou le rendre trivial). Pour un graphe G restreint à une unique arête, on a $\kappa(G) = 1$ et pour le graphe vide, on pose $\kappa(\emptyset) = 0$.

Dans la définition I.6.1, lorsque nous avons parlé de graphe "au moins 2-connexe", cela signifie simplement que $\kappa(G) \geq 2$. Par exemple, le graphe de la figure I.42 qualifié d'au moins 2-connexe est en fait 2-connexe, comme on peut l'observer à la figure I.43; il suffit d'enlever deux sommets bien choisis pour le disconnecter. D'une manière générale, on dira qu'un graphe

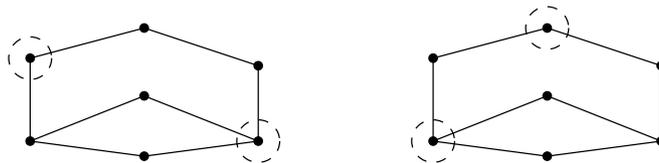


FIGURE I.43. Un graphe 2-connexe.

G est *au moins k -connexe* si $\kappa(G) \geq k$. En particulier, il est trivial²⁴ que si G est *au moins $(k + 1)$ -connexe*, alors il est *au moins k -connexe* (i.e., $\kappa(G) \geq k + 1 \Rightarrow \kappa(G) \geq k$).

Remarque I.6.4. Si un graphe $G = (V, E)$ contient au moins un point d'articulation, il est souvent commode de *séparer* G en ce qu'il est coutume d'appeler ses *composantes 2-connexes*. Il s'agit des composantes connexes obtenues après suppression des points d'articulation. Cette terminologie

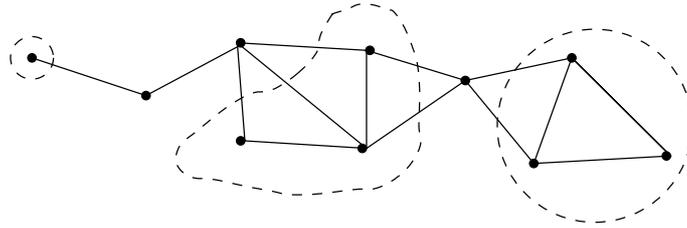


FIGURE I.44. Un graphe et ses composantes 2-connexes.

consacrée n'est pas nécessairement la meilleure car, comme on peut le noter sur la figure I.44, une composante 2-connexe n'est pas nécessairement un graphe 2-connexe ! En fait, les composantes 2-connexes permettent simplement de mettre en évidence les zones du graphe robustes vis-à-vis de la connexité et de la suppression éventuelle d'un sommet.

Après s'être intéressé aux sommets qui, lorsqu'on les supprime, disconnectent un graphe, nous considérons à présent les arêtes ayant une propriété analogue.

Définition I.6.5. Soit $H = (V, E)$ un multi-graphe non orienté connexe (ou une composante connexe d'un multi-graphe non orienté). L'arête e est une *arête de coupure* si $H - e$ n'est plus connexe. Au moins une extrémité d'une arête de coupure est bien évidemment un point d'articulation de H . Comme le montre le graphe de droite à la figure I.45, il y a des situations où seule une extrémité de l'arête de coupure est un point d'articulation.

arête de coupure

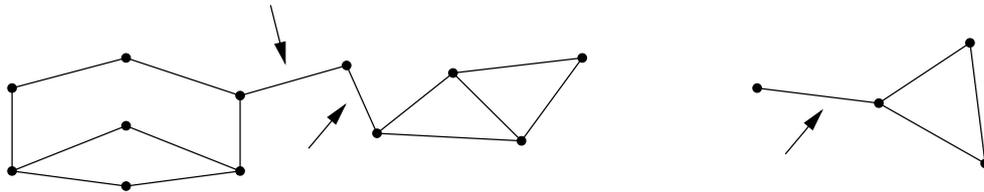


FIGURE I.45. Un graphe et ses arêtes de coupure.

²⁴Attirons l'attention du lecteur sur la différence entre "être k -connexe" et "être au moins k -connexe". En effet, si G est $(k + 1)$ -connexe, il n'est pas k -connexe (i.e., $\kappa(G) = k + 1 \Rightarrow \kappa(G) \neq k$).

Le résultat suivant illustre le concept d'arête de coupure et permet aussi de répondre à la question de la mise à sens unique de routes évoqué dans l'exemple I.3.9.

Proposition I.6.6. *Une arête e est une arête de coupure du graphe $H = (V, E)$ si et seulement si e n'appartient à aucune piste fermée de H .*

Démonstration. Si e est une arête de coupure, il existe des sommets u et v qui sont connectés dans H mais qui ne sont plus connectés dans $H - e$. Il existe donc un chemin joignant u et v qui passe par e . Dans $H - e$, une partie de ce chemin joint u à une extrémité de e , appelons-la e_1 et l'autre partie du chemin joint v à l'autre extrémité de e , appelons-la e_2 . Si e appartient à une piste fermée, il existe un chemin joignant e_1 à e_2 et ne passant pas par e . On peut donc en conclure que u et v sont encore connectés dans $H - e$, ce qui est impossible.

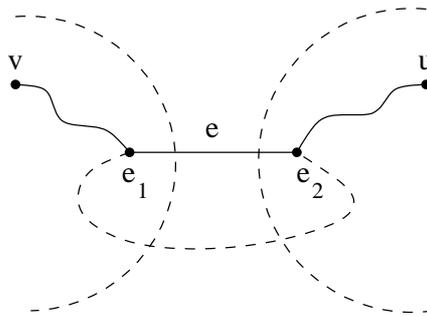


FIGURE I.46. Illustration de la proposition I.6.6

Passons à la réciproque et supposons que $e = \{e_1, e_2\}$ n'est pas une arête de coupure. Si H est connexe, $H - e$ est encore. Ainsi, il existe dans $H - e$ une piste joignant e_1 et e_2 . De là, on conclut que dans H , e appartient à une piste fermée. ■

Remarque I.6.7. Nous pouvons donner une solution au problème posé dans l'exemple I.3.9. En effet, si on dispose d'un graphe non orienté connexe et que l'on désire orienter ses arêtes de manière telle que le graphe résultant soit f. connexe, alors les arêtes de coupure doivent nécessairement être remplacées par deux arcs (pas de sens unique). Par contre, les autres arêtes appartiennent toutes à une piste fermée qu'il est aisé d'orienter (création de sens uniques).

Définition I.6.8. Soit $H = (V, E)$ un multi-graphe non orienté connexe (ou une composante connexe d'un multi-graphe non orienté). L'ensemble $F \subset E$ est un *ensemble de coupure* ou plus simplement une *coupe* ou *coupure* si F est un ensemble **minimal** (pour l'inclusion) tel que $H - F$ n'est pas connexe. A la figure I.48, on a représenté en traits discontinus une des coupures du graphe.

coupe, coupure

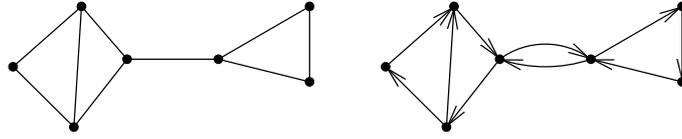


FIGURE I.47. Mise à sens unique

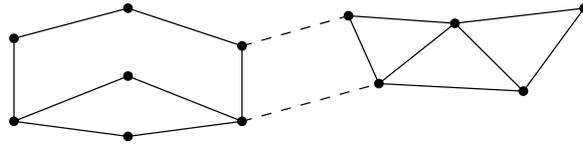


FIGURE I.48. Un graphe et une coupure.

Définition I.6.9. La taille minimale d'une coupe de H se note $\lambda(H)$,

$$\lambda(H) = \min\{\#F \mid F \subseteq E : H - F \text{ disconnecté}\}.$$

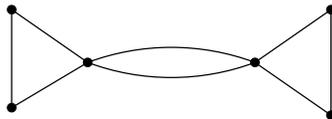
Si H n'est pas connexe, on pose $\lambda(H) = 0$. Notons encore que dans la littérature, on rencontre souvent la notation $\kappa'(H)$ plutôt que $\lambda(H)$. Si H est connexe et si $\lambda(H) = k$, on dit que H est k -connexe (pour les arêtes). Autrement dit, si on enlève $k-1$ arêtes à un graphe k -connexe (pour les arêtes), il reste connexe; par contre, il est possible d'enlever k arêtes pour le disconnecter. On veillera à ne pas confondre les notions de k -connexité et de k -connexité pour les arêtes.

Enfin, on dira qu'un graphe G est *au moins k -connexe (pour les arêtes)*, si $\lambda(G) \geq k$.

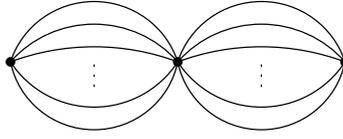
Au vu de la définition ci-dessus, de la proposition I.6.6 et de la remarque I.6.7, on a immédiatement le résultat suivant.

Corollaire I.6.10 (Théorème de H.Robbins (1939)). *On peut orienter un graphe connexe pour le rendre f -connexe si et seulement si ce graphe est au moins 2-connexe pour les arêtes.*

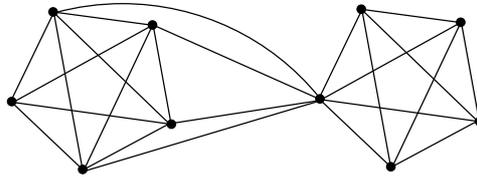
Remarque I.6.11. Sur la figure I.49, le multi-graphe est 2-connexe pour les arêtes et pourtant, il suffit d'enlever un seul sommet pour le disconnecter. Avec nos notations, $\kappa(G) = 1$ et $\lambda(G) = 2$.

FIGURE I.49. Un multi-graphe tel que $\lambda(G) = 2$ et $\kappa(G) = 1$.

De plus, les nombres $\lambda(G)$ et $\kappa(G)$ peuvent être très différents. En effet, pour le graphe de la figure I.50, on a $\kappa(G) = 1$ et $\lambda(G) = k$.

FIGURE I.50. Un multi-graphe tel que $\lambda(G) = k$ et $\kappa(G) = 1$.

Même dans le cas d'un graphe simple, il n'existe pas de lien direct entre $\lambda(G)$ et $\kappa(G)$. On s'en convainc en considérant deux copies de K_n comme illustré à la figure I.51.

FIGURE I.51. Un graphe simple tel que $\lambda(G) = 4$ et $\kappa(G) = 1$.

Enfin, il est clair que nous avons toujours

$$\lambda(G) \leq \min_{v \in V} \deg(v).$$

En effet, si v est un sommet de degré k , il suffit de supprimer les k arêtes incidentes à v pour isoler v des autres sommets du graphe. Signalons aussi un théorème de Whitney (1932),

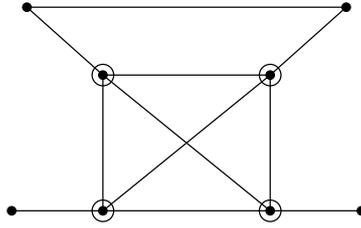
$$\kappa(G) \leq \lambda(G).$$

Il semble évident que plutôt que de supprimer une arête, il suffirait d'en supprimer au plus une extrémité.

Remarque I.6.12. Si F est une coupure d'un graphe G connexe, alors, de par la minimalité de F , $G - F$ possède exactement deux composantes connexes et l'ensemble des sommets de G est donc partitionné en deux sous-ensembles correspondant à ces deux composantes.

Terminons cette section par une dernière définition.

Définition I.6.13. Une *clique* d'un graphe non orienté et simple $G = (V, E)$ est un sous-graphe complet de G . La *taille* d'une clique est le nombre de sommets qui la composent. La taille maximale d'une clique de G est notée $\omega(G)$. Comme nous le verrons plus loin, les nombres $\alpha(G)$ défini en page 37 et $\omega(G)$ sont deux paramètres importants d'un graphe.

FIGURE I.52. Une clique de taille 4, $\omega(G) = 4$.

7. Théorème(s) de Menger

Nous nous plaçons ici dans le cas de graphes simples non orientés. Comme le lecteur pourra s'en convaincre, il n'y a aucune différence à considérer le cas de multi-graphes. La notion définie ci-dessous est proche de la notion d'ensemble d'articulation.

Définition I.7.1. Soient un graphe $G = (V, E)$ et u, v deux sommets distincts de G . Un sous-ensemble $S \subseteq V \setminus \{u, v\}$ sépare u et v s'il n'existe aucun chemin joignant u et v dans le sous-graphe de G induit par $V \setminus S$.

Définition I.7.2. Deux chemins joignant u et v sont *indépendants* si les seuls sommets qu'ils ont en commun sont u et v .

Théorème I.7.3 (Menger (1927)). Soient u, v deux sommets non adjacents d'un graphe connexe $G = (V, E)$. La taille minimum d'un sous-ensemble de sommets séparant u et v est égale au nombre maximum de chemins deux à deux indépendants joignant u et v .

Démonstration. La preuve de ce résultat est assez longue. Dans ce cours introductif, nous ne faisons que l'esquisser sommairement. Si un sous-ensemble $S \subset V$ sépare u et v , alors tout chemin joignant u et v passe nécessairement par un sommet de S . On en conclut que $\#S$ majore le nombre de chemins indépendants joignant u et v .

La seconde partie de la preuve consiste à montrer par récurrence sur $\#E + \#V$ que si un ensemble $S \subset V$ de taille minimum sépare u et v , alors le nombre de chemins indépendants joignant u et v vaut au moins $\#S$.

■

Corollaire I.7.4 (Menger (1927)). Soit $k \geq 2$. Un graphe $G = (V, E)$ est au moins k -connexe (pour les sommets) si et seulement si toute paire de sommets distincts de G est connectée par au moins k chemins indépendants.

Démonstration. Il est clair²⁵ que si toute paire de sommets est connectée par au moins k chemins indépendants, alors $\kappa(G) \geq k$.

La condition est nécessaire. Procédons par l'absurde. Supposons que $\kappa(G) \geq k$ mais qu'il existe deux sommets u et v joints par au plus $k - 1$

²⁵En effet, quand on enlève $k - 1$ sommets du graphe, toute paire de sommets est encore connectée par au moins un chemin.

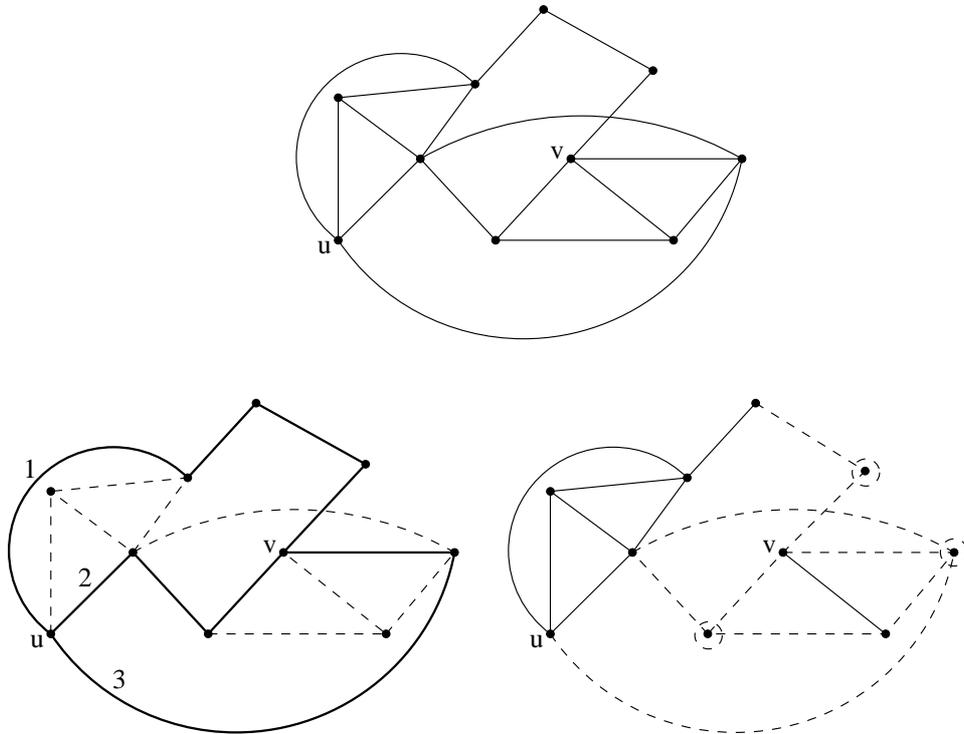


FIGURE I.53. Une illustration du théorème de Menger : 3 chemins indépendants joignent u et v , 3 sommets séparent u et v .

chemins indépendants. Au vu du théorème précédent, on en conclut que u et v sont adjacents, i.e., il existe une arête $e = \{u, v\} \in E$. Dans le graphe $G - e$, u et v sont joints par au plus $k - 2$ chemins indépendants. Puisque u et v ne sont pas adjacents dans $G - e$, on tire du théorème précédent qu'ils peuvent être séparés, dans $G - e$, par un ensemble S de taille minimale ne dépassant pas $k - 2$ sommets, i.e., $\#S \leq k - 2$.

Puisque $\kappa(G) \geq k$, cela implique en particulier que $\#V > k$. Il existe donc un sommet w n'appartenant pas à $S \cup \{u, v\}$. Dans $(G - e) - S$, il ne peut y avoir simultanément deux chemins joignant w respectivement à u et à v car sinon on disposerait dans $(G - e) - S$ de chemins joignant u à w et w à v et on pourrait en conclure que u et v ne sont pas séparés par S . Supposons dès lors qu'aucun chemin ne joint w et u dans $(G - e) - S$. L'ensemble $S \cup \{v\}$ possède (au plus) $k - 1$ éléments et sépare w et u dans G . Ceci contredit le fait que $\kappa(G) \geq k$.

■

Remarque I.7.5. Les résultats énoncés dans cette section concernent des propriétés de connexité relatives aux sommets d'un graphe. Il existe l'analogue de ces résultats en termes d'arêtes : *Un graphe est au moins k -connexe*

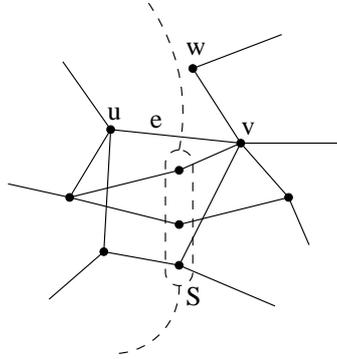


FIGURE I.54. Une illustration de la preuve corollaire de Menger.

pour les arêtes si et seulement si toute paire de sommets est connectée par au moins k chemins ne partageant aucune arête.

8. Graphes orientés sans circuit et tri topologique

Dans cette courte section, on considère un graphe simple orienté et on désire tester algorithmiquement si celui-ci possède ou non un cycle. Comme nous le verrons bientôt, pouvoir répondre à cette question est une des étapes permettant de décider si un graphe donné possède une structure d'arbre. Lorsqu'on prend en compte cette question, il est inutile de considérer un graphe possédant des arcs multiples (cela ne change rien à l'existence d'un cycle) ou des boucles aux sommets. C'est pour cette raison que nous nous restreignons au cas de graphes simples. Comme application, nous allons, en fin de section, reconsidérer le problème général du tri topologique sommairement présenté à l'exemple I.3.15.

Voici tout d'abord une condition nécessaire pour qu'un graphe soit sans circuit.

Lemme I.8.1. *Si un graphe simple orienté $G = (V, E)$ est sans cycle, alors il existe un sommet v tel que $d^-(v) = 0$ (resp. $d^+(v) = 0$).*

Démonstration. Considérons un chemin simple (x_1, \dots, x_k) de G de longueur maximale déterminé par des sommets de G (autrement dit, ce chemin passe par des sommets distincts et il n'est pas possible d'avoir un chemin passant par plus de sommets). Si $d^-(x_1) > 0$, alors il existe $y \in \text{pred}(x_1)$. Si y était égal à un des x_j , on aurait alors un cycle (y, x_1, \dots, x_j) , ce qui est impossible par hypothèse. Or par maximalité du chemin (x_1, \dots, x_k) , il n'est pas possible d'avoir un sommet distinct des x_j et tel que $(y, x_1) \in E$. ■

On peut en déduire une condition nécessaire et suffisante pour qu'un graphe soit sans circuit.

Proposition I.8.2. *Soit $G = (V, E)$ un graphe simple orienté. Ce graphe est sans cycle si et seulement si il existe un sommet v tel que $d^-(v) = 0$ et pour tout sommet v tel que $d^-(v) = 0$, le graphe $G - v$ est sans cycle.*

Démonstration. La condition est nécessaire. Si G est sans cycle, on peut appliquer le lemme précédent. De plus, tout sous-graphe $G - v$ d'un graphe sans cycle G est bien sûr sans cycle.

La condition est suffisante. Soit v un sommet tel que $d^-(v) = 0$. Par hypothèse, $G - v$ est sans cycle. Par conséquent, si le graphe G possède un cycle, ce dernier doit nécessairement passer par v . Si un cycle passe par v , on en conclut que $d^-(v) \geq 1$. C'est impossible, donc G est dépourvu de cycle. ■

Ce résultat nous fournit un premier algorithme permettant de décider si un graphe est sans cycle.

Algorithme I.8.3. La donnée fournie à cet algorithme est un graphe simple orienté $G = (V, E)$.

```
Tant qu'il existe  $v \in V$  tel que  $d^-(v)=0$ ,
   $G := G - v$ 
Si  $G = \emptyset$ 
  alors sortie : "oui,  $G$  sans cycle"
  sinon sortie : "non,  $G$  possède un cycle"
```

Remarque I.8.4. Si on implémente cet algorithme à l'aide de listes d'adjacence (cf. chapitre V), la détection d'un sommet v pour lequel $d^-(v) = 0$ nécessite de parcourir l'ensemble du graphe. Un tel parcours est effectué à chaque étape de la boucle. Cela a pour conséquence de fournir un algorithme dont la complexité est quadratique en fonction de $\#E + \#V$ et celle-ci peut être améliorée.

Théorème I.8.5. *Soit $G = (V, E)$ un graphe simple orienté. Le graphe G est sans cycle si et seulement si il est possible d'énumérer²⁶ les sommets de $V = \{v_1, \dots, v_n\}$ de manière telle que, pour tout $i = 1, \dots, n$, le demi-degré entrant de v_i restreint au graphe $G_i = G - v_1 - \dots - v_{i-1}$ soit nul, ce que l'on note $d_{G_i}^-(v_i) = 0$.*

Démonstration. La condition est nécessaire. Supposons G sans cycle. Ainsi, par le lemme I.8.1, il existe un sommet v_1 de $G = G_1$ tel que $d^-(v_1) = d_{G_1}^-(v_1) = 0$. D'après la proposition précédente, $G_1 - v_1 = G_2$ est sans cycle. Ainsi, il existe un sommet v_2 tel que $d_{G_2}^-(v_2) = 0$. On continue de la sorte de proche en proche et on obtient l'énumération proposée.

La condition est suffisante. Supposons disposer d'une énumération des sommets ayant les propriétés indiquées. Procédons par récurrence. Le

²⁶On entend par énumération, une bijection f de V à valeurs dans $\{1, \dots, \#V\}$. L'unique sommet v tel que $f(v) = i$ sera alors noté v_i .

graphe G_n est restreint à l'unique sommet v_n et est donc sans cycle. Le graphe G_{n-1} contient les sommets v_n et v_{n-1} . De plus, $d_{G_{n-1}}^-(v_{n-1}) = 0$. En d'autres termes, G_{n-1} possède au mieux un arc de v_{n-1} à v_n . Il est donc sans cycle. Appliquons ce raisonnement pour une étape i quelconque. Si le graphe G_{i+1} est sans cycle, alors le graphe G_i se compose du graphe G_{i+1} auquel on ajoute v_i et éventuellement des arcs de v_i vers les sommets de G_{i+1} . On en conclut que G_i est sans cycle.

■

On déduit de ce résultat un algorithme efficace.

Algorithme I.8.6. La donnée fournie à cet algorithme est un graphe simple orienté $G = (V, E)$.

```

Pour tout  $v \in V$ , initialiser  $d(v)=0$ 
Pour tout  $v \in V$ ,
    Pour tout  $w \in \text{succ}(v)$ ,  $d(w)=d(w)+1$ 
aTraiter :=  $\emptyset$ 
nbSommet := 0
Pour tout  $v \in V$ ,
    Si  $d(v) = 0$ , alors
        aTraiter = aTraiter  $\cup \{v\}$ 
        nbSommet := nbSommet + 1
Tant que aTraiter  $\neq \emptyset$ , faire
    Soit  $v$ , le premier élément de aTraiter
    aTraiter := aTraiter  $\setminus \{v\}$ 
    Pour tout  $w \in \text{Succ}(v)$ , faire
         $d(w) = d(w) - 1$ 
        si  $d(w) = 0$ , alors
            aTraiter = aTraiter  $\cup \{w\}$ 
            nbSommet := nbSommet + 1
Si nbSommet =  $\#V$ 
    alors sortie : "oui,  $G$  sans cycle"
    sinon sortie : "non,  $G$  possède un cycle"

```

La variable d associée à chaque sommet du graphe permet de stocker le demi-degré entrant du sommet (par rapport au graphe envisagé au moment de la construction). A chaque fois qu'un élément v est enlevé de la liste **aTraiter**, celui-ci est énuméré. Ainsi, on énumère d'abord les sommets de demi-degré entrant nul. Ensuite, lorsqu'un sommet est traité, on le supprime du graphe et on modifie en conséquence les demi-degrés entrants. Si tous les sommets ont été traités, cela correspond à dire que tous les sommets ont été énumérés. Au vu du théorème précédent, on en conclut que le graphe est sans cycle.

Définition I.8.7. Soit $G = (V, E)$ un graphe simple orienté. Un *tri topologique* de G est une énumération v_1, \dots, v_n des sommets de G de manière telle que si (v_i, v_j) est un arc de G , alors $i < j$.

Théorème I.8.8. *Un graphe simple orienté admet un tri topologique si et seulement si il est sans cycle.*

Démonstration. Il est clair que si un graphe possède un cycle, alors quelle que soit l'énumération de ses sommets, il ne peut s'agir d'un tri topologique.

Si un graphe est sans cycle, alors une énumération de ses sommets donnant lieu à un tri topologique est immédiatement donnée par le théorème I.8.5. ■

Remarque I.8.9. Il n'y a pas qu'un seul tri topologique pour un graphe donné $G = (V, E)$. En effet, si on dénote par

$$S(G) = \{v \in V \mid d^-(v) = 0\}$$

l'ensemble des *sources* de G , alors l'ensemble des tris topologiques de G est donné par la formule récursive suivante

$$\Pi(G) = \bigcup_{v \in S(G)} \{v.\sigma \mid \sigma \in \Pi(G - v)\}$$

où σ est un tri topologique de $G - v$ et où $v.\sigma$ désigne l'énumération des sommets de G en débutant par v puis en suivant l'énumération prescrite par σ .

9. Arbres

Les graphes connexes acycliques jouent un rôle prépondérant dans diverses applications (cf. l'exemple I.3.8 ou encore de nombreux exemples tirés de l'informatique : arbres binaires de recherche, bases de données, etc. . .).

Définition I.9.1. Un graphe simple non orienté $A = (V, E)$ est un *arbre* s'il est connexe et sans *cycle* (sous-entendu, un "véritable" cycle : une piste fermée, pas un cycle "artificiel" comme un trivial $(\{a, b\}, \{b, a\})$ qui est un cycle de longueur 2; le fait d'imposer l'absence d'une piste fermée évite de telles arêtes doublées et on pourrait de manière équivalente imposer l'absence de circuit simple). Une *forêt* est un graphe simple non orienté dont chaque composante connexe est un arbre. Un arbre $A = (V, E)$ est qualifié de *n-aire* si pour tout sommet $v \in V$, $\deg(v) \leq n$.

"Une forêt est formée d'arbres, fallait-il faire des maths pour s'en rendre compte ?"

Proposition I.9.2. *Soit $G = (V, E)$ un arbre ayant n sommets.*

- ▶ *Toute paire de sommets distincts de G est connectée par exactement un chemin simple.*
- ▶ *Soit $e \in (V \times V) \setminus E$ qui n'est pas une boucle. Le graphe $G + e$ contient un cycle (i.e., une piste fermée), c'est-à-dire, $G + e$ n'est plus un arbre.*
- ▶ *Le graphe G a exactement $n - 1$ arêtes.*

Démonstration. C'est immédiat. ■

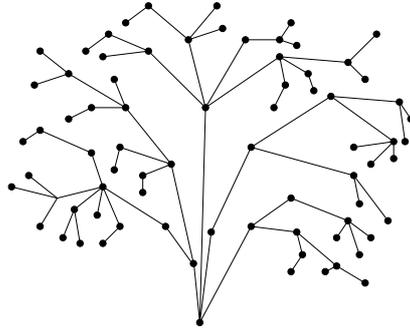


FIGURE I.55. Un arbre.

On pourra en particulier observer que tout arbre est 1-connexe, la réciproque étant bien évidemment fautive.

Proposition I.9.3. *Un graphe $G = (V, E)$ simple connexe est un arbre si et seulement si chacune de ses arêtes est une arête de coupure.*

Démonstration. Soient G un arbre et e une de ses arêtes. Puisque G est sans cycle, il ne possède aucune piste fermée. En appliquant la proposition I.6.6, on en déduit que e est une arête de coupure. Inversement, si G est un graphe connexe possédant une piste fermée alors, par la proposition I.6.6, les arêtes de cette piste ne peuvent être des arêtes de coupure. ■

Corollaire I.9.4. *Tout graphe connexe possède un sous-arbre couvrant.*

Démonstration. Soient $G = (V, E)$ un graphe connexe et $C = (V, E')$ un sous-graphe couvrant connexe minimal (i.e., on ne peut pas remplacer E' par un sous-ensemble strict tout en conservant la connexité de C). Vu la minimalité de C , chacune de ses arêtes est une arête de coupure de C . Par la proposition précédente, on en conclut que C est un arbre. ■

Corollaire I.9.5. *Si $G = (V, E)$ est un graphe (simple non orienté) connexe, alors $\#E \geq \#V - 1$.*

Démonstration. Par le corollaire précédent, G possède un sous-arbre couvrant $C = (V, E')$. De là, il vient

$$\#E \geq \#E' = \#V - 1$$

où pour la dernière égalité, on a utilisé la proposition I.9.2. ■

Définition I.9.6. Un arbre $A = (V, E)$ dans lequel on a privilégié un sommet v_0 est appelé *arbre pointé*²⁷. On le notera (A, v_0) . Le sommet v_0 est parfois appelé la *racine* de l'arbre.

²⁷en anglais, *rooted tree*.

Pour un arbre pointé (A, v_0) , les sommets de A peuvent être ordonnés suivant leur distance à v_0 . Si v est un sommet tel que $d(v_0, v) = i$, on dira que v est un sommet de niveau i . Un arbre pointé a été représenté à la figure I.56.

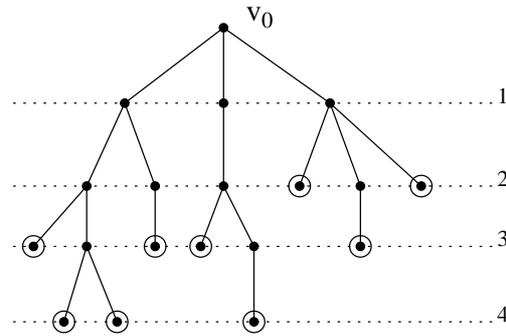


FIGURE I.56. Un arbre pointé.

Si v est un sommet de niveau i et si tous ses voisins sont de niveau $i - 1$, on dit alors que v est une *feuille* de l'arbre. A la figure I.56, les feuilles ont été marquées d'un cercle. La *hauteur* d'un arbre est le niveau maximal de ses feuilles. L'arbre de la figure I.56 est un arbre de hauteur 4.

Remarque I.9.7. Pointer un arbre définit naturellement une orientation des arêtes de l'arbre. En effet, on peut orienter les arcs de façon à ce qu'ils joignent les sommets de niveau i aux sommets de niveau $i + 1$. Dans ce cadre, on parle souvent des *fil*s (resp. du *père*) d'un noeud v pour désigner ses successeurs (resp. son unique prédécesseur). Les *descendants* (resp. *ancêtres*) de v désignent les éléments de $\text{succ}^*(v)$ (resp. $\text{pred}^*(v)$).

Définition I.9.8. Un arbre pointé est *k-aire* si tout sommet a au plus k fils. Si $k = 2$, on parle naturellement d'arbre binaire. Un arbre k -aire de hauteur n possède au plus

$$1 + k + \dots + k^n = \frac{k^{n+1} - 1}{k - 1}$$

sommets. S'il en possède exactement ce nombre, on parle d'arbre *k-aire complet*.

9.1. Parcours d'arbres. Un parcours d'un arbre est une façon d'en ordonner les noeuds. On supposera implicitement que les fils d'un noeud v_i sont ordonnés $v_{i,1}, \dots, v_{i,k_i}$ et que cet ordre est connu et fixé une fois pour toutes.

On distingue trois types de *parcours en profondeur*: les parcours préfixe, infixé et suffixé. Soit (A, v_0) un arbre pointé. Pour le *parcours préfixe*, on parcourt d'abord la racine puis on parcourt, de manière récursive et dans l'ordre, les sous-arbres pointés de racine respective $v_{0,1}, \dots, v_{0,k_0}$. Pour le *parcours suffixe*, on parcourt d'abord, de manière récursive et dans l'ordre,

les sous-arbres pointés de racine $v_{0,1}, \dots, v_{0,k_0}$, puis la racine v_0 . Pour le *parcours infixe*, nous supposons disposer d'un arbre binaire. (On peut donc parler du sous-arbre de gauche et du sous-arbre de droite.) On parcourt d'abord, de manière récursive, le sous-arbre de gauche, puis la racine, et enfin le sous-arbre de droite²⁸.

Exemple I.9.9. Considérons l'arbre binaire pointé donné à la figure I.57. Les fils d'un noeud sont ordonnés par l'ordre alphabétique de leur label. Si

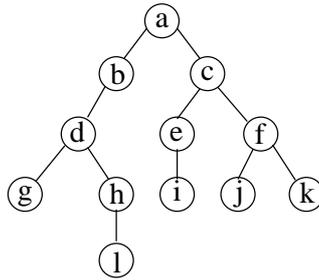


FIGURE I.57. Un arbre à parcourir.

on parcourt cet arbre de manière préfixielle, on obtient la suite :

$$a, b, d, g, h, l, c, e, i, f, j, k.$$

Pour un parcours suffixe, on a

$$g, l, h, d, b, i, e, j, k, f, c, a.$$

Enfin, pour un parcours infixe, on obtient

$$g, d, l, h, b, a, i, e, c, j, f, k.$$

On rencontre parfois des *parcours en largeur*. Dans ce cas, on parcourt les noeuds de l'arbre pointé par niveau croissant. Sur l'exemple précédent, on a simplement l'ordre a, b, c, \dots, k, l .

Remarque I.9.10. On peut également définir un parcours en profondeur d'un graphe quelconque. Bien qu'il ne s'agisse pas d'un problème concernant spécifiquement les arbres, nous pensons qu'il s'agit du moment opportun pour le définir. Soit $G = (V, E)$ un graphe (orienté ou non) que l'on peut supposer simple et connexe. Un *parcours en profondeur* de G est défini récursivement comme suit. Sélectionner un sommet v_0 . A l'étape $k \geq 1$, on choisit un voisin de v_{k-1} qui n'a pas encore été sélectionné. Si un tel voisin n'existe pas, on cherche dans l'ordre, un voisin non sélectionné de v_{k-2}, \dots, v_0 .

²⁸Si dans un arbre binaire, un sommet n'a qu'un seul fils, on prend par convention que le sous-arbre de droite de ce sommet est vide.

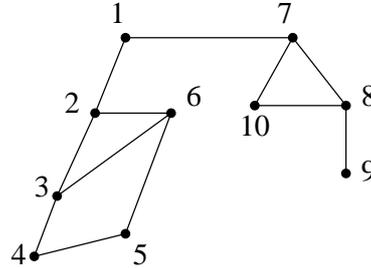


FIGURE I.58. Exemple de parcours en profondeur d'un graphe.

10. Isomorphismes de graphes

Définition I.10.1. Soient $G_i = (V_i, E_i)$, $i = 1, 2$, deux digraphes (resp. deux graphes non orientés). Une application $f : V_1 \rightarrow V_2$ est un *homomorphisme* si

$$(x, y) \in E_1 \Rightarrow (f(x), f(y)) \in E_2$$

$$\text{(resp. } \{x, y\} \in E_1 \Rightarrow \{f(x), f(y)\} \in E_2 \text{)}.$$

On parlera alors d'homomorphisme de G_1 dans G_2 . Il est clair que la composée d'homomorphismes est encore un homomorphisme.

Exemple I.10.2. Avec les graphes G et H de la figure I.59, on voit facilement qu'on a un homomorphisme de G dans H mais pas de H dans G . A la figure I.60, on donne un autre exemple d'homomorphisme entre deux

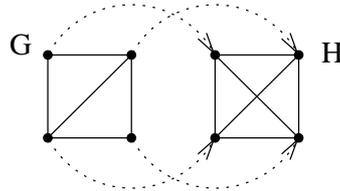


FIGURE I.59. Homomorphisme de G dans H .

graphes G et H . Cela montre que $f : V_1 \rightarrow V_2$ n'est pas nécessairement injectif. Les homomorphismes de graphes réinterviendront dans les questions de coloriage.

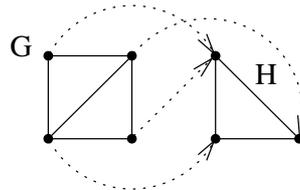


FIGURE I.60. Homomorphisme de G dans H .

Définition I.10.3. Deux digraphes (resp. deux graphes non orientés) $G_i = (V_i, E_i)$, $i = 1, 2$, sont *isomorphes* s'il existe une bijection $f : V_1 \rightarrow V_2$ qui est telle que

$$(x, y) \in E_1 \Leftrightarrow (f(x), f(y)) \in E_2$$

(resp. telle que $\{x, y\} \in E_1 \Leftrightarrow \{f(x), f(y)\} \in E_2$). Cette définition s'adapte au cas de multi-graphes orientés. Deux multi-graphes $G_i = (V_i, E_i)$, $i = 1, 2$, sont isomorphes s'il existe une bijection $f : V_1 \rightarrow V_2$ telle que (x, y) est un arc de multiplicité k de G_1 si et seulement si $(f(x), f(y))$ est un arc de multiplicité k de G_2 . Bien évidemment, une telle application f est qualifiée d'*isomorphisme* de graphes. Bien sûr, si f est un isomorphisme, il en va de même pour f^{-1} .

Exemple I.10.4. Voici deux graphes isomorphes représentés à la figure I.61. On a un isomorphisme donné par

$$\varphi : a \mapsto 4, b \mapsto 5, c \mapsto 6, d \mapsto 1, e \mapsto 2, f \mapsto 3.$$

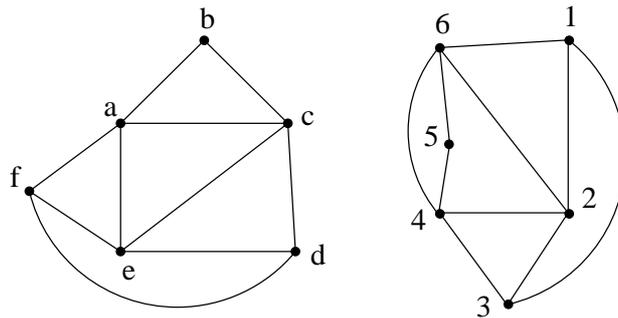


FIGURE I.61. Deux graphes isomorphes.

Définition I.10.5. Soit $G = (V, E)$ un graphe (orienté ou non). Un *automorphisme* de G est un isomorphisme de G dans G . L'ensemble des automorphismes de G muni de la loi de composition d'applications forme un groupe, noté $Aut(G)$. Il s'agit bien évidemment d'un sous-groupe du groupe symétrique \mathcal{S}_n des permutations de $n = \#V$ éléments. Un graphe pour lequel $Aut(G)$ est réduit à l'identité id_V est qualifié d'*asymétrique*.

Exemple I.10.6. Par exemple, $Aut(K_n) = \mathcal{S}_n$.

La proposition suivante est immédiate et s'adapte aux autres cas de graphes.

Proposition I.10.7. Soient G un graphe (simple non orienté) et φ un automorphisme de G . Pour tous sommets u, v , on a

- ▶ $\deg(u) = \deg(\varphi(u))$,
- ▶ $d(u, v) = d(\varphi(u), \varphi(v))$.

Remarque I.10.8. Si deux graphes $G_i = (V_i, E_i)$, $i = 1, 2$ ont leurs sommets pondérés par $p_i : V_i \rightarrow \Sigma$, la définition d'un isomorphisme $f : V_1 \rightarrow V_2$ doit naturellement s'étendre en respectant de plus la propriété

$$p_1(v) = p_2(f(v)), \quad \forall v \in V_1.$$

Voici un exemple qui concerne des arbres infinis et qui met en lumière la notion d'isomorphisme dans le cas pondéré. Pour simplifier, supposons disposer de deux symboles (ou lettres) notés a et b . Avec ces lettres, on peut écrire des mots comme : aa , bba , b ou encore $abbbaabaa$ (i.e., des suites finies de symboles). On construit un arbre binaire infini (i.e., on dispose d'une infinité de noeuds, chaque noeud ayant exactement deux fils) dont les noeuds sont en bijection avec les mots que l'on peut écrire avec les lettres a et b . Si un noeud est en bijection avec le mot m , son fils de gauche (resp. de droite) est en bijection avec ma (resp. mb). La racine de l'arbre correspond au mot n'ayant aucune lettre, le mot vide ε . Nous dirons qu'un tel arbre est un *arbre lexicographique*. Ainsi, cet arbre possède exactement 2^i noeuds de niveau i et ceux-ci correspondent de gauche à droite aux mots de longueur i : $\underbrace{a \cdots aa}_{i \times}$, $a \cdots ab, \dots b \cdots ba$, $\underbrace{b \cdots bb}_{i \times}$. Une illustration est donnée à la figure I.62.

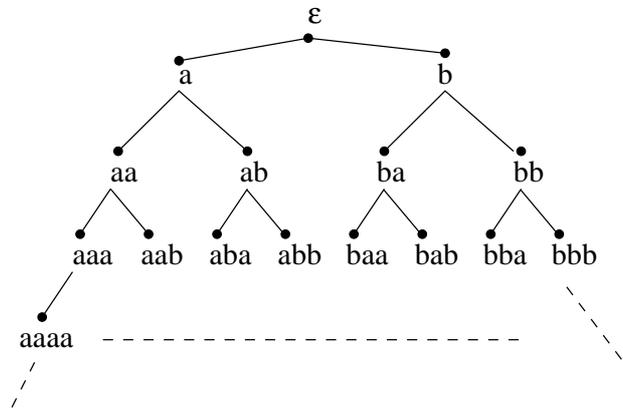


FIGURE I.62. Un arbre lexicographique infini.

Nous n'avons pas encore considéré de pondération des noeuds. Vu la bijection entre noeuds et mots, nous ne les distinguerons plus dans ce qui suit. Si on considère un ensemble L de mots écrits sur $\{a, b\}$, alors on définit la fonction p_L qui à un mot m associe 1 (resp. 0) si $m \in L$ (resp. $m \notin L$). Autrement dit, la pondération est simplement un codage définissant le *dictionnaire* des mots de L . Si on représente par un rond noir les mots appartenant à L (i.e., tel que $p_L(m) = 1$), on a, pour le langage formé des mots commençant par un nombre arbitraire de a (éventuellement aucun) et suivi par un nombre arbitraire de b (éventuellement aucun), l'arbre pondéré A_L repris à la figure I.63.

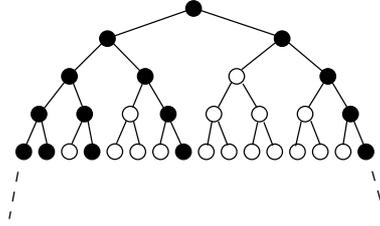


FIGURE I.63. Un arbre infini pondéré représentant le dictionnaire de A_L .

Pour tout mot m , on peut définir l'arbre A_m comme le sous-arbre obtenu à partir de A_L en considérant comme nouvelle racine le noeud m et en ne conservant dans A_m que les descendants de m (m inclus). On peut se convaincre que l'arbre A_L ne possède, à isomorphisme près, que 3 sous-arbres non isomorphes (par exemple, A_L lui-même, A_b et A_{ba}). Un arbre A_L ayant une telle propriété (nombre fini de sous-arbres non isomorphes²⁹) est qualifié de *régulier*.

Par exemple, tout ensemble fini de mots donne lieu à un arbre régulier et l'ensemble M des mots de la forme

$$a^i b^i := \underbrace{a \cdots a}_{i \times} \underbrace{b \cdots b}_{i \times}, \quad \forall i \in \mathbb{N},$$

donne quant à lui un arbre A_M non régulier (cf. figure I.64). En effet, les

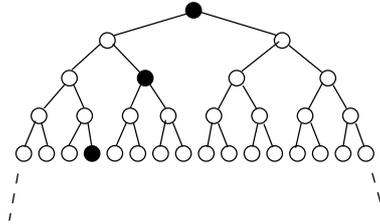


FIGURE I.64. Un arbre infini pondéré représentant le dictionnaire de A_M .

arbres $A_\varepsilon, A_a, A_{aa}, A_{aaa}, \dots$ sont tous distincts (on remarque que, dans A_{a^i} , le premier niveau sur lequel on rencontre un noeud noirci est le i -ième).

11. Graphes hamiltoniens

Nous avons vu dans la section 4.2 comment déterminer si un graphe G possédait un chemin eulérien passant une et une seule fois par chaque arête/arc de G . On peut aussi s'intéresser à la question posée initialement

²⁹Remarquons que puisque l'arbre est infini, il possède une infinité de sous-arbres mais à isomorphisme près, il n'existe que 3 sous-arbres distincts.

par Sir W. R. Hamilton qui concerne cette fois l'existence d'un circuit passant une et une seule fois par chaque sommet³⁰ de G . Il est frappant de constater qu'il n'existe, à ce jour, pas de méthode efficace³¹ permettant de répondre à cette question. Un chemin (resp. circuit) passant une et une seule fois par chaque sommet de G sera qualifié d'*hamiltonien*. Un graphe *hamiltonien* est un graphe possédant un circuit hamiltonien³².

Exemple I.11.1. On considère un dodécaèdre régulier (polyèdre régulier possédant 12 faces pentagonales et 20 sommets). Le graphe associé, ou squelette, (on procède comme à l'exemple I.3.3) est représenté à la figure I.65. La question originale posée par Hamilton était de déterminer un circuit passant une et une seule fois par chaque arête de ce graphe.

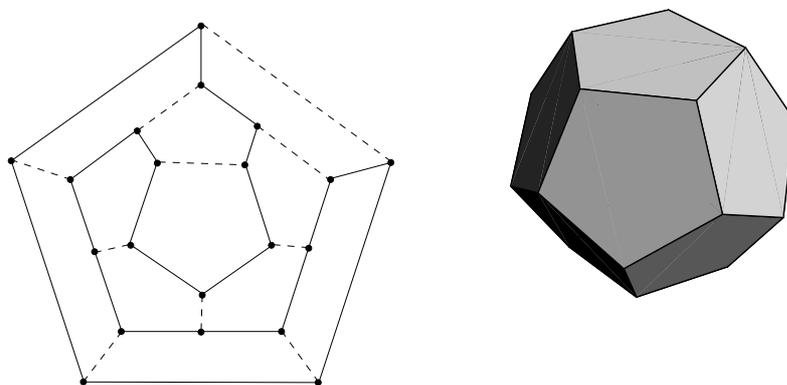


FIGURE I.65. Squelette d'un dodécaèdre et circuit hamiltonien.

Exemple I.11.2. Sur un échiquier, le cavalier se déplace de deux cases dans une direction (horizontale ou verticale) puis d'une case dans une direction orthogonale. Est-il possible qu'un cavalier passe par toutes les cases de l'échiquier sans passer deux fois par la même case et revenir à son point de départ. Il s'agit encore de la recherche d'un circuit hamiltonien. On peut considérer un graphe dont les sommets représentent les cases de l'échiquier et une arête joint deux sommets si et seulement si un cavalier peut passer de l'une des cases correspondantes à la suivante en un mouvement. (Ainsi,

³⁰Dans sa version initiale (1859), Hamilton avait choisi un dodécaèdre dont chaque sommet représentait une ville importante: Bruxelles, Canton, Delhi,...

³¹Passer en revue toutes les permutations des n sommets d'un graphe est inconcevable. On entend par "efficace", une méthode pouvant être menée à bien en un temps polynomial par rapport à la taille des données. Dans le cours de complexité, on montrera qu'il s'agit en fait d'un problème NP-complet.

³²Nous poserons qu'un graphe restreint à une unique arête $\{a, b\}$ (resp. restreint à un unique sommet) est hamiltonien. En effet, on peut considérer le circuit $\{a, b\}$, $\{b, a\}$ passant par les sommets a, b puis a . Par contre, il est facile de voir qu'un arbre contenant 3 sommets au moins n'est jamais hamiltonien.

le degré de chaque sommet varie entre 2 et 8). On peut répondre affirmativement à cette question comme le prouve la figure I.66.

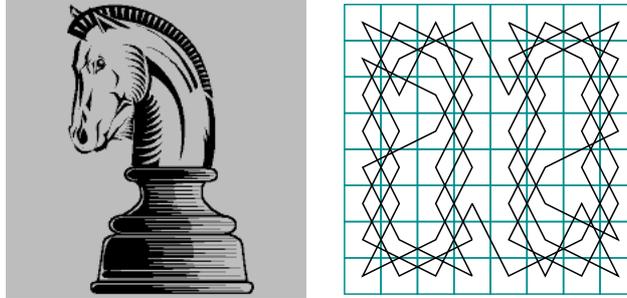


FIGURE I.66. Déplacement d'un cavalier.

Voici une première condition nécessaire pour qu'un graphe soit hamiltonien.

Proposition I.11.3. *Si $G = (V, E)$ est un graphe (simple et non orienté) hamiltonien, alors pour tout ensemble non vide $S \subseteq V$, le nombre de composantes connexes de $G - S$ est inférieur ou égal à $\#S$.*

Démonstration. Soit S un sous-ensemble non vide de V et u un sommet de S . Considérons un circuit hamiltonien C passant par u . On peut voir C comme une suite ordonnée de sommets $(u, w_1, \dots, w_{n-1}, u)$. Soient G_1, \dots, G_k , les k composantes connexes de $G - S$. Si $k = 1$, le résultat est trivial. Supposons donc $k > 1$. Pour $i = 1, \dots, k$, désignons par u_i le dernier sommet de G_i dans le circuit C et v_i , le sommet suivant u_i dans ce même circuit. Les sommets v_1, \dots, v_k appartiennent à S . En effet, on a un cycle passant par tous les sommets du graphe et on sait que la suppression des sommets de S disconnecte le graphe en k composantes. Par définition de

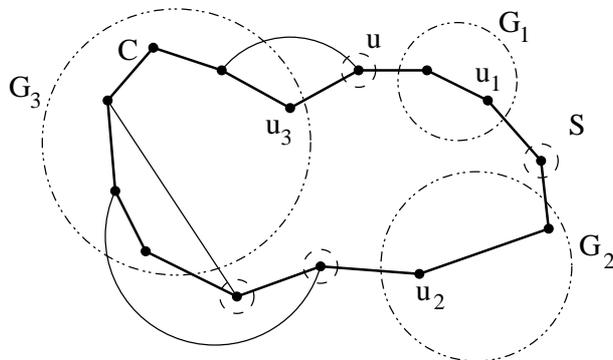


FIGURE I.67. Une illustration de la preuve de la proposition I.11.3.

ces composantes et des sommets u_i et v_i , on en conclut³³ que v_1, \dots, v_k sont des sommets distincts de S et donc $\#S \geq k$.

■

Exemple I.11.4. Ce premier résultat peut être utilisé pour vérifier que certains graphes ne sont pas hamiltoniens. Par exemple, le graphe de la figure I.68 ne vérifie pas la condition de la proposition précédente pour l'ensemble S constitué des deux sommets encadrés. En effet, en supprimant ces deux sommets, on obtient trois composantes connexes.

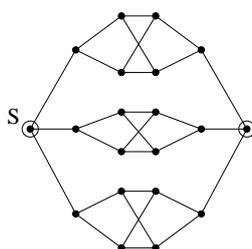


FIGURE I.68. Un graphe non hamiltonien.

Par contre, cette condition n'est pas suffisante. En effet, le *graphe de Petersen* (figure I.69) la vérifie et pourtant, ce dernier n'est pas hamiltonien (les vérifications sont laissées au lecteur). Le graphe de Petersen est souvent utilisé comme contre-exemple classique en théorie des graphes pour mettre en défaut certaines propriétés non généralisables.

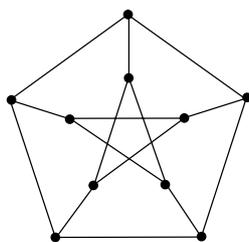


FIGURE I.69. Le graphe de Petersen.

On dispose de plusieurs conditions suffisantes (théorèmes de Dirac, de Chvátal, ou encore de Chvátal-Erdős) pour assurer qu'un graphe est hamiltonien. On supposera toujours le graphe simple et non orienté. La première de ces conditions est une condition locale, elle exprime que si chaque sommet a suffisamment de voisins, alors le graphe est hamiltonien. Pour rappel, la

³³On peut procéder par l'absurde. Si v_1 n'appartient pas à S , il n'est pas supprimé dans le graphe $G - S$. Les sommets u_1 et v_1 sont voisins (en effet, ils sont consécutifs dans le chemin C). Par conséquent, v_1 doit appartenir à G_1 . Or par définition de u_1 et de v_1 , u_1 est le dernier sommet de G_1 dans la suite donnée par C donc v_1 ne peut lui appartenir. Une contradiction.

partie entière *par excès* (resp. *par défaut*) d'un réel x est $\lceil x \rceil = \inf\{y \in \mathbb{Z} \mid y \geq x\}$ (resp. $\lfloor x \rfloor = \sup\{y \in \mathbb{Z} \mid y \leq x\}$).

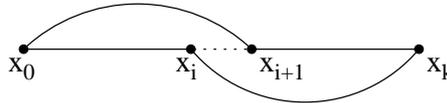
Théorème I.11.5 (Dirac). *Tout graphe G (simple et non orienté) ayant $n \geq 3$ sommets et tel que le degré de chaque sommet est au moins égal à $n/2$, possède un circuit hamiltonien.*

Démonstration. Notons $\delta = \inf_{v \in V} \deg(v)$. Puisque $\delta \geq \lceil n/2 \rceil$, on en conclut que G est connexe. En effet, si ce n'était pas le cas, G posséderait au moins deux composantes connexes distinctes dont une contient $\leq \lfloor n/2 \rfloor$ sommets³⁴. Or si $\delta \geq \lceil n/2 \rceil$, alors un des sommets de cette composante doit être voisin d'un sommet d'une autre composante, ce qui n'est pas possible (vu la maximalité des composantes connexes).

Soit C un chemin de longueur maximale passant par des sommets tous distincts (i.e., un chemin simple). Puisque G est simple, C est entièrement défini par une suite de sommets x_0, \dots, x_k . Par maximalité, tous les voisins de x_0 et de x_k appartiennent à C (en effet, sinon, on pourrait étendre C en un chemin plus long). Par conséquent, au moins $\lceil n/2 \rceil$ sommets parmi x_0, \dots, x_{k-1} sont voisins de x_k et au moins $\lceil n/2 \rceil$ sommets parmi x_1, \dots, x_k sont voisins de x_0 . De plus, $k+1 \leq n$ (ou encore $k < n$). Par conséquent³⁵, il existe un indice $i < k$ tel que

$$\{x_0, x_{i+1}\} \in E \quad \text{et} \quad \{x_i, x_k\} \in E.$$

Ainsi,



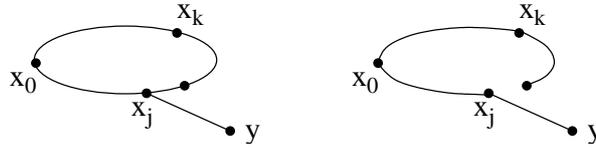
$$x_0, x_1, \dots, x_i, x_k, x_{k-1}, \dots, x_{i+2}, x_{i+1}, x_0$$

est un circuit passant une seule fois par chacun des sommets x_0, \dots, x_k . Ce circuit est hamiltonien. En effet, supposons qu'il existe un sommet y n'appartenant pas à ce circuit. Puisque G est connexe, il existe $j \in \{0, \dots, k\}$ tel que $\{x_j, y\} \in E$. De là, on peut construire un chemin passant par $k+2$ sommets distincts. C'est impossible vu le choix de C (de longueur maximale).

■

³⁴Il n'est pas possible que chaque composante ait plus de $\lceil n/2 \rceil$ éléments. Sinon le nombre total de sommets dépasserait n .

³⁵Cela paraît-il vraiment si évident ? Prouvons-le par l'absurde. Si ce n'était pas le cas, pour tout $0 \leq i < k$ tel que $\{x_0, x_{i+1}\} \in E$, on aurait $\{x_i, x_k\} \notin E$ (et réciproquement, pour tout $0 \leq j < k$ tel que $\{x_j, x_k\} \in E$, $\{x_0, x_{j+1}\} \notin E$). Il y a au moins $\lceil n/2 \rceil$ indices distincts $i \in I \subseteq \{0, \dots, k-1\}$ tels que $\{x_0, x_{i+1}\} \in E$. De plus, il y a au moins $\lceil n/2 \rceil$ indices distincts $j \in J \subseteq \{0, \dots, k-1\}$ tels que $\{x_j, x_k\} \in E$ et $I \cap J = \emptyset$. Ceci conduit à une absurdité car $I \cup J \subset \{0, \dots, k-1\}$ mais $\#(I \cup J) \geq n > k$.



Le théorème suivant, premier théorème d’Ore, sera utilisé pour prouver le théorème de Chvátal (condition suffisante à ce jour la plus aboutie pour assurer qu’un graphe soit hamiltonien).

En particulier, l’idée d’adjoindre des arêtes à un graphe va nous fournir, à la section suivante, la notion importante de fermeture d’un graphe. On peut d’ores et déjà remarquer que l’adjonction d’une arête $\{x, y\}$ est réalisée quand $\deg(x) + \deg(y) \geq \#V$.

Enfin, nous attendrons la formulation du corollaire I.11.7, le deuxième théorème d’Ore, pour obtenir une condition suffisante assurant le caractère hamiltonien plus générale que celle donnée par le théorème de Dirac.

Théorème I.11.6 (“premier” Théorème d’Ore). *Soient $G = (V, E)$ un graphe (simple et non orienté) ayant $n \geq 3$ sommets et x et y deux sommets tels que $\deg(x) + \deg(y) \geq n$. Le graphe G est hamiltonien si et seulement si le graphe $G + \{x, y\}$ l’est.*

On utilise ce résultat pour démontrer le théorème I.11.12.

Démonstration. Si l’arête $e = \{x, y\}$ appartient à E , il n’y a rien à démontrer. Nous allons donc supposer que cette arête n’appartient pas à G . De plus, la condition est trivialement nécessaire.

Supposons donc que le graphe $G + \{x, y\}$ possède un circuit hamiltonien C passant par l’arête $e = \{x, y\}$. En effet, si le circuit hamiltonien ne passe pas par e , il n’y a rien non plus à démontrer. Ainsi, le circuit C peut s’écrire comme une suite ordonnée de sommets (tous distincts exceptées les deux copies de x)

$$C = (v_1 = x, v_2, \dots, v_n = y, x).$$

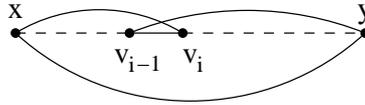
Il existe i tel que $1 < i < n$, $\{x, v_i\} \in E$ et $\{v_{i-1}, y\} \in E$: c’est une conséquence de l’hypothèse $\deg(x) + \deg(y) \geq n$ (et c’est le même genre d’argument³⁶ que dans le théorème de Dirac). On en déduit (cf. figure I.70) que

$$(v_i, v_{i+1}, \dots, y, v_{i-1}, v_{i-2}, \dots, x, v_i)$$

est un circuit hamiltonien dans G (puisqu’il ne fait plus intervenir l’arête e).

■

³⁶Puisque $e \notin E$, x et y ont à eux deux au moins n voisins parmi v_2, \dots, v_{n-1} . Il n’est pas possible que y n’ait que v_{n-1} comme unique voisin car alors, x devrait avoir au moins $n - 1$ voisins parmi v_2, \dots, v_{n-1} , ce qui est impossible. Donc y a au moins 2 voisins et par des arguments similaires, x aussi. On peut alors procéder par l’absurde. Supposons que pour tout $i \in I \subseteq \{2, \dots, n - 1\}$ tel que $\{x, v_i\} \in E$, $\{v_{i-1}, y\} \notin E$ et que pour tout $j \in J \subseteq \{2, \dots, n - 1\}$ tel que $\{v_{j-1}, y\} \in E$, $\{x, v_j\} \notin E$. De là, $I \cap J = \emptyset$, on devrait dès lors avoir $\#(I \cup J) = \#I + \#J \geq n$ mais cela contredit le fait que $I \cup J \subseteq \{2, \dots, n - 1\}$. D’où la conclusion car $\{v_1, y\} = \{x, v_n\} = \{x, y\}$ n’appartient pas à E .

FIGURE I.70. Théorème d'Ore, circuit dans G .

Nous verrons rapidement que ce théorème permet d'obtenir facilement le résultat suivant (nous pourrions d'ailleurs en donner une preuve directement mais pour éviter les redondances, nous la postposons).

Corollaire I.11.7 (“deuxième” Théorème d'Ore). *Soit $G = (V, E)$ un graphe (simple et non orienté) ayant $n \geq 3$ sommets. Si pour tout couple de sommets non adjacents (x, y) , on a $\deg(x) + \deg(y) \geq n$, alors G est hamiltonien. En particulier, si $\min_{v \in V} \deg(v) \geq n/2$, alors G est hamiltonien.*

Remarque I.11.8. On peut observer que le théorème de Dirac est un corollaire immédiat du deuxième théorème d'Ore. En effet, si pour chaque sommet $\deg(v) \geq n/2$, alors pour toute paire de sommets x et y , on a trivialement $\deg(x) + \deg(y) \geq n$.

Remarque I.11.9. Il est à noter que chronologiquement, c'est d'abord le théorème de Dirac qui fut obtenu (1952), puis celui d'Ore (1960) et enfin celui de Chvátal (1971).

11.1. Fermeture d'un graphe et théorème de Chvátal. Introduisons tout d'abord la *fermeture* d'un graphe simple et non orienté. Soit $G_0 = (V_0, E_0)$ un graphe simple et non orienté. On définit une suite finie

$$G_0, G_1, \dots, G_i = (V_i, E_i), \dots, G_k$$

de graphes (simples) de la manière suivante. Pour tout i , on ajoute à G_i une arête comme suit :

$$G_{i+1} = G_i + \{u, v\}$$

où u et v sont des sommets de G_i qui sont tels que $\{u, v\} \notin E_i$ et

$$\deg_{G_i}(u) + \deg_{G_i}(v) \geq \#V$$

où \deg_{G_i} désigne bien sûr le degré d'un sommet dans le graphe G_i . La procédure s'arrête lorsqu'il n'y a plus moyen d'ajouter de nouvelles arêtes à G_k . Ainsi, pour tous sommets u, v , soit $\{u, v\}$ appartient à E_k , soit $\deg_{G_k}(u) + \deg_{G_k}(v) < \#V$. Le graphe obtenu à la dernière étape s'appelle la *fermeture* de G_0 . Nous allons tout d'abord montrer que quels que soient les choix d'arêtes réalisés dans les étapes intermédiaires, on aboutit toujours au même graphe (ainsi la définition est bien licite) noté $\mathcal{F}(G_0)$.

Exemple I.11.10. A la figure I.71, on a considéré deux graphes et leur fermeture. A chaque fois, on a noté le degré des sommets considérés pour l'ajout d'une nouvelle arête. On peut noter que pour le graphe du haut,

la fermeture donne le graphe complet K_6 , alors que dans le second cas, la couverture n'est pas le graphe complet K_5 .

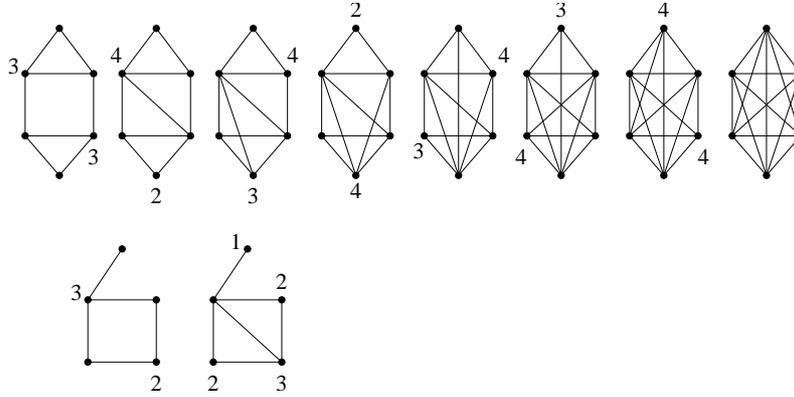


FIGURE I.71. Construction de la fermeture d'un graphe.

Lemme I.11.11. *Pour tout graphe ayant au moins trois sommets, la fermeture d'un graphe est unique.*

Démonstration. Supposons que G possède $n \geq 3$ sommets et qu'il est possible de construire une fermeture de G en adjoignant de nouvelles arêtes de deux manières distinctes. Ainsi, supposons que

$$H = G + \{e_1, \dots, e_r\} \quad \text{et} \quad H' = G + \{f_1, \dots, f_s\}$$

sont deux fermetures de G . Nous allons montrer que $H = H'$. On note $H_i = G + \{e_1, \dots, e_i\}$ et $H'_i = G + \{f_1, \dots, f_i\}$. On a $G = H_0 = H'_0$. Si $H \neq H'$, il y a au moins une arête de l'un qui n'est pas dans l'autre. Supposons que $e_k = \{u, v\}$ est la première arête de H qui n'appartient pas à H' . (Ainsi, e_1, \dots, e_{k-1} sont des arêtes de H' et e_k diffère de tous les f_i .) Puisque e_k est l'arête ajoutée à H_{k-1} pour construire H_k , on a que

$$\deg_{H_{k-1}}(u) + \deg_{H_{k-1}}(v) \geq n.$$

Il est clair que H_{k-1} est un sous-graphe de H' . De là, on en conclut que

$$\deg_{H'}(u) + \deg_{H'}(v) \geq n$$

et donc, l'arête e_k devra aussi être ajoutée à H' , une contradiction. Par conséquent, H est un sous-graphe de H' et par symétrie, on en tire que $H = H'$. ■

Le théorème suivant est en fait une simple conséquence du (premier) théorème d'Ore.

Théorème I.11.12. *Soit G un graphe (simple et non orienté) ayant au moins trois sommets.*

- ▶ *Le graphe G est hamiltonien si et seulement si sa fermeture $\mathcal{F}(G)$ l'est.*
- ▶ *Si la fermeture $\mathcal{F}(G)$ de G est un graphe complet, alors G est hamiltonien.*

Démonstration. Pour la première partie, G est un sous-graphe de $\mathcal{F}(G)$. Ainsi, si G est hamiltonien, alors $\mathcal{F}(G)$ l'est aussi. Pour la réciproque, considérons une suite de graphes $G_0 = G, \dots, G_{k-1}, G_k = \mathcal{F}(G)$ donnant la fermeture de G . Si G_k est hamiltonien, alors G_{k-1} l'est aussi. C'est une conséquence immédiate du théorème d'Ore I.11.6. De proche en proche, on en conclut que $G_0 = G$ est hamiltonien.

La deuxième partie est immédiate. Tout graphe complet étant hamiltonien, il suffit d'appliquer la première partie de ce théorème. ■

Remarque I.11.13. La condition suffisante donnée dans le théorème précédent n'est pas nécessaire. En effet, considérons un graphe constitué de $n > 4$ sommets et de n arêtes et formant un unique circuit hamiltonien. Chaque sommet étant de degré 2, le graphe est égal à sa fermeture et bien que le graphe soit hamiltonien, la fermeture n'est pas le graphe complet K_n .

Remarque I.11.14 (Preuve du corollaire I.11.7). Nous pouvons à présent déduire du théorème précédent la preuve du corollaire I.11.7. (Notons que le théorème I.11.12 repose entièrement sur le premier théorème d'Ore I.11.6. La seule raison pour laquelle nous avons attendu pour donner la preuve du corollaire I.11.7 (deuxième théorème d'Ore) réside dans l'utilisation de la fermeture d'un graphe permettant d'alléger les développements).

En effet, si un graphe G satisfait les conditions données dans l'énoncé du corollaire I.11.7, on en déduit immédiatement que sa fermeture est un graphe complet. (On peut joindre chaque paire de sommets non voisins.)

Voici enfin l'énoncé et la preuve du théorème de Chvátal. Notons que ce théorème est parfois attribué à Bondy-Chvátal.

Théorème I.11.15 (Chvátal). *Soit G un graphe (simple et non orienté) ayant $n \geq 3$ sommets ordonnés par degré croissant, i.e.,*

$$\deg(v_1) \leq \deg(v_2) \leq \dots \leq \deg(v_n).$$

Si, pour tout $k < n/2$, le graphe satisfait

$$(1) \quad \deg(v_k) \leq k \Rightarrow \deg(v_{n-k}) \geq n - k,$$

alors G possède un circuit hamiltonien.

Remarque I.11.16. La condition du théorème de Chvátal peut facilement être testée pour un graphe donné. Il suffit d'ordonner la suite des degrés et de vérifier une condition combinatoire élémentaire. Rappelons que d'un point de vue strictement logique, si pour $k < n/2$, on a $\deg(v_k) > k$, alors l'implication $\deg(v_k) \leq k \Rightarrow \deg(v_{n-k}) \geq n - k$ est toujours vraie.

A ce jour, on ne connaît pas de condition suffisante plus générale pour qu'un graphe soit hamiltonien. Dans certains cas, on peut obtenir des conditions plus fortes mais en se restreignant à des classes particulières de graphes.

Démonstration. Nous pouvons tout d'abord supposer que G est égal à sa fermeture, $G = \mathcal{F}(G)$. En effet, au vu du théorème I.11.12, un graphe est hamiltonien si et seulement si sa fermeture l'est. De plus, si un graphe G satisfait la condition (1), alors le graphe $G + e$ la satisfait encore (c'est même encore plus "facile" puisque³⁷ moins de sommets vont vérifier $\deg(v_k) \leq k$). Nous allons montrer qu'avec cette supposition, $G = K_n$.

Procédons par l'absurde et supposons $G \neq K_n$. Soient u, v deux sommets de G tels que $\deg(u) \leq \deg(v)$, $\{u, v\} \notin E$ et tels que $\deg(u) + \deg(v)$ soit maximal (nécessairement $< n$, car sinon l'arête $\{u, v\}$ appartiendrait à E). On peut d'abord remarquer que

$$\deg(u) = i < n/2$$

(car sinon, $\deg(u) + \deg(v) \geq n$). Soit

$$A = \{w \in V \mid \{w, v\} \notin E \text{ et } w \neq v\}.$$

Par notre choix de u , $\deg(w) \leq i$ pour tout $w \in A$ (en effet, on a choisi $\deg(u) + \deg(v)$ maximal). De plus, dans A , se trouvent tous les sommets distincts de v et non voisins de v , ainsi,

$$\#A = (n - 1) - \deg(v) \geq \deg(u) = i.$$

On en conclut que dans G , il y a au moins i sommets de degré au plus i . Si comme dans l'énoncé, on a ordonné les sommets de G par degré croissant, on en conclut que $\deg(v_i) \leq \deg(u) = i$. Par hypothèse, cela entraîne que

$$(2) \quad \deg(v_{n-i}) \geq n - i.$$

On effectue à présent le même raisonnement avec

$$B = \{w \in V \mid \{u, w\} \notin E \text{ et } w \neq u\}.$$

Dès lors, pour tout $w \in B$,

$$\deg(w) \leq \deg(v) < n - \deg(u) = n - i$$

et de plus,

$$\#B = (n - 1) - \deg(u) = n - 1 - i.$$

On a donc $n - 1 - i$ sommets de degré $< n - i$. En outre, le sommet u est tel que $\deg(u) < n - i$. En effet, $i + \deg(v) = \deg(u) + \deg(v) < n$ et

³⁷Soit k entiers ordonnés par ordre croissant $x_1 \leq x_2 \leq \dots \leq x_k$. Soit $i \in \{1, \dots, k\}$. Si on remplace x_i par $x_i + 1$ et que l'on note les éléments réordonnés $x'_1 \leq \dots \leq x'_k$, alors il existe $j \geq i$ tel que $x_i + 1 = x'_j$. Pour tout $t \in \{1, \dots, k\}$, montrons que $x_t \leq x'_t$ (cela montre donc que si (1) est satisfait, l'ajout d'une arête ne va pas modifier cette propriété). Tout d'abord, si $i = j$, il n'y a rien à démontrer (l'ordre des éléments n'a pas changé). On peut donc supposer $j > i$. Si $t < i$ ou $t > j$, alors $x_t = x'_t$. Si $i \leq t < j$, alors $x'_t = x_{t+1} \geq x_t$. Si $t = j$, alors $x'_j \geq x'_{t-1} = x_t = x_j$.

$\deg(u) \leq \deg(v)$ et on en tire que $\deg(u) < n - i$. En conclusion, on a donc $n - i$ sommets de degré $< n - i$ et $\deg(v_{n-i}) < n - i$ ce qui contredit (2). ■

Enfin, citons sans démonstration un dernier résultat concernant les graphes hamiltoniens.

Théorème I.11.17 (Chvátal-Erdős). *Soient G un graphe simple et non orienté ayant au moins trois sommets, $\alpha(G)$ le nombre maximal de sommets indépendants et $\kappa(G)$ la taille minimale d'un ensemble d'articulation. Si $\kappa(G) \geq \alpha(G)$, alors G est hamiltonien.*

Démonstration. Voir par exemple, R. Diestel, page 277. ■

Terminons cette section par le résultat suivant présentant un certain intérêt pour le problème du voyageur de commerce. Supposons disposer de n villes et de connections entre toutes ces villes. On supposera que tous les coûts de connexion d'une ville à une autre sont identiques (ainsi, toute permutation des n villes fournit une solution optimale). On est donc en présence du graphe complet K_n . Le voyageur de commerce désire passer une et une seule fois par chacune des villes et revenir à son point de départ (le siège de sa compagnie par exemple). Il est clair qu'une permutation quelconque des $n-1$ sommets du graphe (siège de la compagnie exclu) répond à la question. Cependant, on peut imaginer qu'il ne désire pas repasser par une route déjà empruntée précédemment. De cette manière, il pourra joindre l'utile à l'agréable en visitant la région de diverses manières. Ainsi, on désire non seulement construire un circuit hamiltonien mais de plus, on voudrait trouver un nombre maximum de tels circuits n'ayant aucune arête commune.

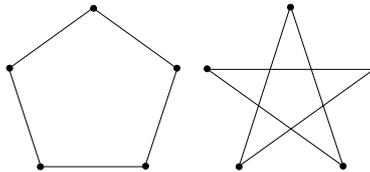


FIGURE I.72. Deux circuits hamiltoniens disjoints de K_5 .

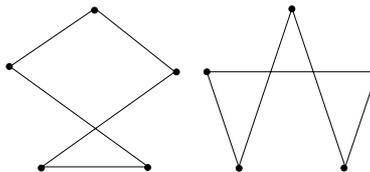
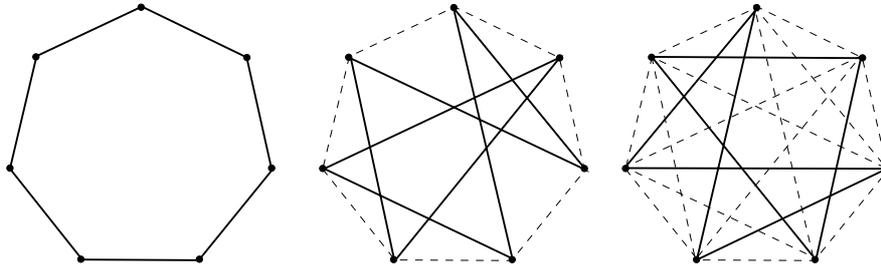


FIGURE I.73. Deux autres circuits hamiltoniens disjoints de K_5 .

FIGURE I.74. Une tentative de partition de K_7 .

Proposition I.11.18. *Pour $n \geq 3$, K_n peut être partitionné en circuits hamiltoniens disjoints si et seulement si n est impair. En particulier, le nombre de circuits formant une partition de K_n vaut $(n-1)/2$.*

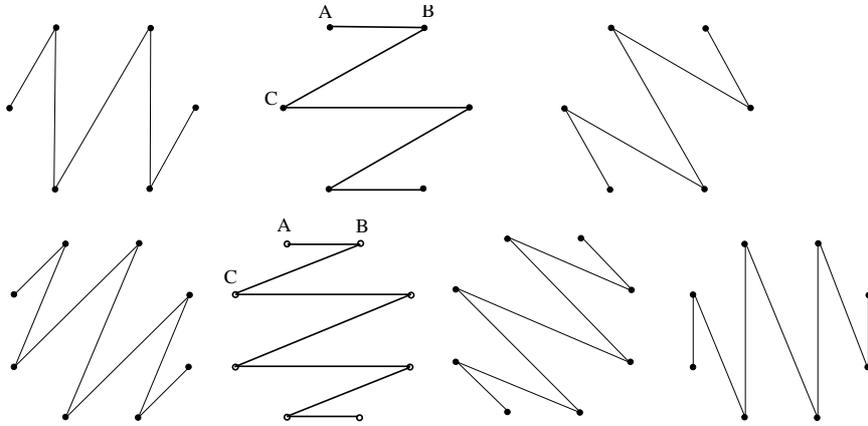
Démonstration. Nous savons que K_n est $(n-1)$ -régulier (i.e., le degré de chaque sommet vaut $n-1$). De plus, un circuit hamiltonien est 2-régulier. Par conséquent, il est nécessaire que $n-1$ soit pair pour pouvoir décomposer K_n en cycles hamiltoniens disjoints. (Sinon, certaines arêtes, une par sommet, ne feraient partie d'aucun cycle).

Supposons à présent n impair. Il est évident que K_n peut être décomposé en *au plus* $(n-1)/2$ circuits hamiltoniens disjoints. En effet, on sélectionne d'abord un premier cycle hamiltonien formé de n arêtes (on a donc sélectionné deux arêtes incidentes à chaque sommet). Pour le second cycle de K_n , puisque celui-ci doit être disjoint du précédent, le choix se restreint (ce choix correspond encore une fois à sélectionner deux arêtes incidentes à chaque sommet parmi les arêtes non déjà sélectionnées). Puisque chaque cycle choisi sélectionne deux arêtes par sommet et que K_n est $(n-1)$ -régulier, on ne peut sélectionner qu'au maximum $(n-1)/2$ circuits hamiltoniens disjoints. Notons que la construction d'exactly $(n-1)/2$ circuits disjoints peut *a priori* être réalisée de plusieurs façons (comme le montrent les figures I.72 et I.73).

A ce stade, le lecteur ne *doit* pas être convaincu (en effet, rien n'assure *a priori* que la procédure proposée fournit le résultat annoncé; le lecteur pourrait objecter qu'à une étape intermédiaire, il ne soit plus possible de continuer et il aurait tout à fait raison !). Comme le montre la figure I.74, un choix arbitraire de deux arêtes à chaque étape peut mener à l'impossibilité d'obtenir des cycles (la figure de droite montre deux cycles non connexes, l'un de longueur 4 et l'autre de longueur 3).

L'obtention d'exactly $(n-1)/2$ circuits hamiltoniens est assurée par les deux lemmes suivants. (Nous nous basons sur les constructions de Bollobás en donnant en plus des arguments géométriques.)

Lemme I.11.19. *Si n est pair, K_n peut être partitionné en $n/2$ chemins hamiltoniens disjoints.*

FIGURE I.75. Chemins hamiltoniens disjoints de K_6 et K_8 .

Pour prouver ce résultat, on peut considérer des arguments géométriques. On peut identifier les sommets de K_n avec les sommets d'un polygone régulier à n côtés possédant une symétrie orthogonale par rapport à une horizontale (autrement dit, possédant deux côtés horizontaux). On obtient un chemin hamiltonien \mathcal{C} en reliant les sommets se trouvant sur une même horizontale et en reliant le sommet gauche du niveau horizontal i avec le sommet de droite de niveau $i + 1$. On remarque³⁸ que toutes ces arêtes "obliques" ont une même pente π/n (c'est-à-dire mesure de l'angle qu'elles forment avec une horizontale). Nous affirmons qu'en effectuant une rotation de la figure \mathcal{C} de $2k\pi/n$, $k = 1, \dots, n/2 - 1$, on obtient $n/2$ chemins hamiltoniens distincts. Pour vérifier qu'ils sont tous distincts, il suffit de raisonner sur la pente respective des arêtes les constituant. Une illustration est fournie à la figure I.75. En effet, pour \mathcal{C} , les arêtes "obliques" ont pour pente π/n , puis en effectuant les rotations on obtient les pentes distinctes $3\pi/n, 5\pi/n, \dots, (1 - 1/n)\pi$.

Lemme I.11.20. *Soit $n \geq 3$ impair. Le graphe K_n peut être partitionné en $(n-1)/2$ circuits hamiltoniens disjoints si et seulement si K_{n-1} peut être partitionné en $(n-1)/2$ chemins hamiltoniens disjoints.*

Le résultat est presque immédiat. Si K_n peut être partitionné et qu'on lui supprime un sommet, on passe à K_{n-1} et chaque circuit hamiltonien \mathcal{C} donne naissance à un chemin hamiltonien (les extrémités du chemin étant les sommets voisins dans \mathcal{C} du sommet supprimé). Réciproquement, si K_{n-1} est partitionné, l'adjonction d'un sommet permet de passer à K_n en "fermant" chaque chemin hamiltonien pour obtenir des circuits disjoints.

■

³⁸Le polygone à n côtés peut être triangulé au moyen de $n - 2$ triangles. Par conséquent, la somme des mesures de ses angles vaut $(n - 2)\pi$ et donc chacun de ses angles a pour mesure $(n - 2)\pi/n$. Si on considère, comme à la figure I.75, le triangle ABC , puisque ce dernier est isocèle, l'angle en B a pour mesure $(\pi - (n - 2)\pi/n)/2 = \pi/n$.

CHAPITRE II

Un peu de théorie algébrique des graphes

Dans ce chapitre, nous ne faisons qu'esquisser quelques résultats mettant en évidence les liens entre la théorie des graphes et l'algèbre linéaire (par exemple, on étudie le spectre des graphes réguliers ou bipartis). On y présente de manière détaillée la théorie de Perron-Frobenius (sans démonstration du théorème principal) avec comme application directe, l'estimation du nombre de chemins de longueur n que l'on peut trouver dans un graphe à composantes connexes primitives. On présente également diverses questions relatives aux sous-arbres couvrants : nombre de tels sous-arbres, recherche d'un arbre de poids minimal, algorithmes de Prim et de Kruskal. En particulier, ce chapitre met en lumière quelques raisonnements de combinatoire énumérative.

1. Matrice d'adjacence

Définition II.1.1. Soit $G = (V, E)$ un multi-graphe non orienté dont les sommets sont ordonnés par $V = \{v_1, \dots, v_n\}$. La *matrice d'adjacence* de G est la matrice $A(G)$ dont l'élément $[A(G)]_{i,j}$ est égal au nombre d'arêtes $\{v_i, v_j\}$ présentes dans E , $1 \leq i, j \leq n$. (Pour rappel, E est en général un multi-ensemble.) Il s'agit donc d'une matrice symétrique à coefficients entiers naturels. Le polynôme caractéristique de G , noté $\chi_G(\lambda)$, est le polynôme caractéristique de sa matrice d'adjacence $A(G)$. Par abus de langage, on parlera des valeurs propres de G , étant sous-entendu qu'il s'agit des valeurs propres de $A(G)$. On parlera donc aussi du spectre de G .

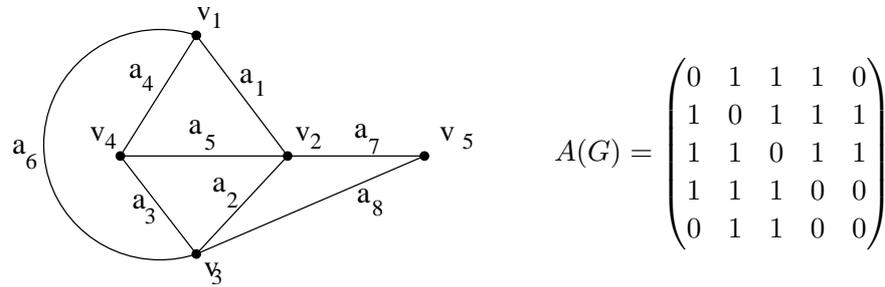
On peut remarquer que les éléments de la matrice d'adjacence d'un graphe simple appartiennent à $\{0, 1\}$ et que la trace de cette matrice vaut 0.

Remarque II.1.2. En se rappelant quelques résultats du cours d'algèbre de première année, on remarque que la matrice d'adjacence d'un graphe non orienté est toujours diagonalisable par une matrice orthogonale (pour chaque valeur propre, les multiplicités algébrique et géométrique coïncident) et que ses valeurs propres sont réelles.

Exemple II.1.3. Considérons le graphe (simple) G de la figure II.1. Avec les notations précédentes, on a aussi

$$\chi_G(\lambda) = -\lambda^5 + 8\lambda^3 + 10\lambda^2 + \lambda - 2.$$

Proposition II.1.4. Deux graphes G_1 et G_2 sont isomorphes si et seulement si ils ont, à une permutation près, la même matrice d'adjacence.

FIGURE II.1. Un graphe G et sa matrice d'adjacence.

$\det A(G_1) = \det A(G_2)$.

Autrement dit, il existe une matrice de permutation¹ P telle que

$$A(G_1) = P^{-1}A(G_2)P.$$

Démonstration. C'est immédiat et ce résultat est même transposable au cas de graphes orientés. ■

Définition II.1.5. On peut aussi définir la *matrice d'incidence "sommets/arêtes"*. Si $V = \{v_1, \dots, v_n\}$ et $E = \{e_1, \dots, e_m\}$, il s'agit d'une matrice B de dimension $n \times m$ telle que $B_{i,j} = 1$ si et seulement si e_j est incident à v_i . En poursuivant l'exemple précédent, cette matrice vaut ici

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

Définition II.1.6. Dans un graphe simple, on appelle *triangle*, tout triplet d'arêtes distinctes deux à deux de la forme $\{a, b\}$, $\{b, c\}$, $\{c, a\}$ (i.e., tout circuit de longueur trois formé d'arêtes distinctes).

Proposition II.1.7. Si le polynôme caractéristique de $G = (V, E)$ est de la forme

$$\chi_G(\lambda) = (-\lambda)^n + c_1(-\lambda)^{n-1} + c_2(-\lambda)^{n-2} + \dots + c_n,$$

alors certains coefficients de χ_G sont en relation directe avec G :

- ▶ c_1 est le nombre de boucles de G , en particulier, si G est simple, $c_1 = 0$.
- ▶ Si G est simple, alors $-c_2$ est le nombre d'arêtes de G .
- ▶ Si G est simple, alors c_3 est le double du nombre de triangles de G .

Démonstration. Le premier point est immédiat. Le coefficient c_1 est la somme des éléments diagonaux de A_G . Si G est simple, les sous-matrices

¹Une matrice de permutation est une matrice pour laquelle chaque ligne et chaque colonne contiennent exactement un 1, les autres entrées étant nulles.

diagonales² de A_G de dimension 2 sont de la forme

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{ou} \quad \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Le coefficient c_2 étant la somme des déterminants de ces sous-matrices ceux-ci valant respectivement -1 et 0 , il est clair que $c_2 = -\#E$. Pour le dernier point, on raisonne de la même façon. Les sous-matrices diagonales non nulles de A_G de dimension 3 sont d'une des formes suivantes (à une permutation des lignes et des colonnes près, ce qui ne change pas la valeur du déterminant)

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{ou} \quad \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Les deux premières ont un déterminant nul et la troisième a un déterminant égal à 2. Le coefficient c_3 étant la somme des déterminants de ces sous-matrices et la dernière matrice correspondant à la présence d'un triangle dans G , on obtient le résultat annoncé. ■

Remarque II.1.8. On voit donc que le polynôme caractéristique de $A(G)$ fournit des renseignements sur le graphe G . Cependant, deux graphes non isomorphes peuvent avoir le même polynôme caractéristique³. On parle alors de *graphes cospectraux*. Par exemple, les deux graphes de la figure II.2 ont le même spectre. En effet, ils ont tous les deux comme polynôme caractéristique,

$$-1 + 4\lambda + 7\lambda^2 - 4\lambda^3 - 7\lambda^4 + \lambda^6.$$

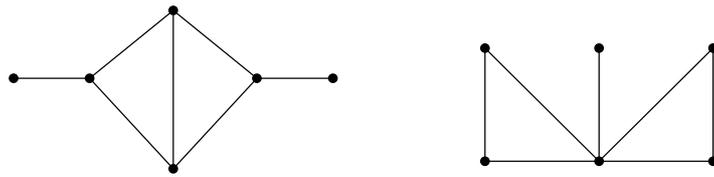


FIGURE II.2. Deux graphes cospectraux.

Proposition II.1.9. *Soit $G = (V, E)$ un graphe biparti. Si λ est valeur propre de G , alors $-\lambda$ l'est aussi. Autrement dit, le spectre d'un graphe biparti est symétrique par rapport à 0.*

Démonstration. Par hypothèse, V se partitionne en deux sous-ensembles V_1 et V_2 de manière telle que toute arête de G est de la forme $\{u, v\}$ avec

²Cela signifie qu'on sélectionne des lignes et des colonnes de même indice.

³C'est comme en algèbre matricielle, deux matrices peuvent avoir le même polynôme caractéristique sans pour autant se réduire à la même forme de Jordan.

$u \in V_1$ et $v \in V_2$. Si on ordonne les sommets de V de manière à considérer tout d'abord les sommets de V_1 , alors $A(G)$ a la forme

$$\begin{pmatrix} 0 & B \\ \tilde{B} & 0 \end{pmatrix}$$

où B est une matrice de dimension $\#V_1 \times \#V_2$. Soit x un vecteur propre non nul de $A(G)$ de valeur propre λ . Appelons x_1 (resp. x_2) le vecteur obtenu en considérant les $\#V_1$ premières (resp. les $\#V_2$ dernières) composantes de x . Ainsi,

$$\begin{pmatrix} 0 & B \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} Bx_2 \\ \tilde{B}x_1 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Nous pouvons à présent facilement exhiber un vecteur propre non nul de valeur propre $-\lambda$,

$$\begin{pmatrix} 0 & B \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ -x_2 \end{pmatrix} = \begin{pmatrix} -Bx_2 \\ \tilde{B}x_1 \end{pmatrix} = \lambda \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix} = -\lambda \begin{pmatrix} x_1 \\ -x_2 \end{pmatrix}.$$

■

Nous montrerons un peu plus loin qu'on dispose également de la réciproque de ce résultat (cf. corollaire II.2.6).

On peut aussi définir la matrice d'adjacence d'un multi-graphe orienté. La définition est analogue à celle donnée précédemment sauf qu'on tient ici compte de l'orientation. Il en résulte que la matrice obtenue n'est en général pas symétrique.

Définition II.1.10. Soit $G = (V, E)$ un multi-graphe orienté dont les sommets sont ordonnés par $V = \{v_1, \dots, v_n\}$. La *matrice d'adjacence* de G est la matrice $A(G)$ dont l'élément $[A(G)]_{i,j}$ est égal au nombre d'arcs (v_i, v_j) présents dans E , $1 \leq i, j \leq n$.

Exemple II.1.11. Si on considère le graphe de la figure I.2, on obtient la matrice d'adjacence suivante.

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

Théorème II.1.12. Soit $G = (V, E)$ un multi-graphe (orienté ou non) tel que $V = \{v_1, \dots, v_k\}$. Pour tous $i, j \in \{1, \dots, k\}$ et pour tout $n > 0$,

$$[A(G)^n]_{i,j}$$

est exactement le nombre de chemins de longueur n joignant v_i à v_j .

Démonstration. On procède par récurrence sur n . Le cas de base $n = 1$ découle de la définition même de la matrice d'adjacence. Supposons le

résultat vérifié pour $n > 0$ et vérifions-le pour $n + 1$. On a bien sûr

$$[A(G)^{n+1}]_{i,j} = \sum_{s=1}^k [A(G)^n]_{i,s} [A(G)]_{s,j}.$$

Par hypothèse de récurrence, $[A(G)^n]_{i,s}$ compte le nombre de chemins de longueur n joignant v_i à v_s . De plus, $[A(G)]_{s,j}$ compte le nombre d'arcs/arêtes joignant v_s à v_j . Par conséquent, $[A(G)^n]_{i,s} [A(G)]_{s,j}$ compte le nombre de chemins de longueur $n + 1$ joignant v_i à v_j en passant par v_s , d'où la conclusion. ■

2. Théorie de Perron-Frobenius

La connexité d'un graphe se traduit par une propriété immédiate de sa matrice d'adjacence. On peut même, dans certains cas, obtenir des renseignements plus fins sur la longueur des chemins joignant deux sommets quelconques d'une composante connexe. Donnons tout d'abord deux définitions concernant les matrices à coefficients positifs ou nuls (faites attention dans les deux énoncés à l'ordre des quantificateurs). Nous verrons ensuite le rapport entre ces matrices et les graphes.

Définition II.2.1. Une matrice carrée $A = (a_{ij})_{1 \leq i,j \leq n}$ à coefficients (réels) positifs ou nuls est *irréductible* si pour tous $i, j \in \{1, \dots, n\}$, il existe⁴ un entier $N(i, j) \geq 0$ tel que

$$[A^{N(i,j)}]_{i,j} > 0.$$

Définition II.2.2. Une matrice carrée $A = (a_{ij})_{1 \leq i,j \leq n}$ à coefficients (réels) positifs ou nuls est *primitive* s'il existe un entier $N > 0$ tel que pour tous $i, j \in \{1, \dots, n\}$

$$[A^N]_{i,j} > 0$$

ce que l'on s'autorise à noter $A^N > 0$ étant sous-entendu que les inégalités sont interprétées composante à composante. On remarque aussi que toute matrice primitive est irréductible.

Dans cette section, pour deux matrices réelles A et B de même dimension, il sera commode d'écrire $A < B$ (resp. $\leq, \geq, >$) si l'inégalité a lieu composante à composante. Cela n'entraîne pas d'ambiguïté particulière.

Proposition II.2.3. *Un multi-graphe orienté (resp. non orienté) est fortement connexe (resp. connexe) si et seulement si sa matrice d'adjacence est irréductible.*

Démonstration. C'est une conséquence directe du théorème II.1.12. ■

⁴Avec un peu d'expérience, le lecteur pourra se convaincre que, dans cette définition, on aurait aussi pu imposer de manière équivalente $N(i, j) > 0$.

Remarque II.2.4. Si la matrice d'adjacence d'un graphe (orienté ou non) est primitive, cela entraîne que le graphe est non seulement connexe mais qu'il existe N tel que, quelle que soit la paire de sommets considérée, il existe un chemin de longueur N les joignant. Par abus de langage, on s'autorisera alors à parler de *graphe primitif*.

Lorsqu'une matrice est irréductible (ou, en particulier primitive), on dispose du puissant théorème de Perron-Frobenius.

Théorème II.2.5 (Perron-Frobenius). *Soit $A \geq 0$ une matrice carrée irréductible de dimension n .*

- ▶ *La matrice A possède un vecteur propre $v_A \in \mathbb{R}^n$ (resp. $w_A \in \mathbb{R}^n$) dont les composantes sont toutes strictement positives et correspondant à une valeur propre $\lambda_A > 0$,*

$$A v_A = \lambda_A v_A \text{ (resp. } \widetilde{w}_A A = \lambda_A \widetilde{w}_A \text{)}.$$

- ▶ *Cette valeur propre λ_A possède une multiplicité algébrique (et géométrique) simple.*
- ▶ *Tout vecteur propre de A dont les composantes sont strictement positives est un multiple de v_A .*
- ▶ *Toute autre valeur propre $\mu \in \mathbb{C}$ de A est telle que $|\mu| < \lambda_A$.*
- ▶ *Si μ est une valeur propre de A telle que $|\mu| = \lambda_A$, alors*

$$\mu = \lambda_A e^{2ik\pi/d}$$

pour un certain $d \geq 1$ et $k \in \{0, \dots, d-1\}$. De plus, pour tout $k \in \{0, \dots, d-1\}$, $\lambda_A e^{2ik\pi/d}$ est une valeur propre de A .

- ▶ *Soit B une matrice réelle à coefficients positifs ou nuls de même dimension que A . Si $B \leq A$, alors pour toute valeur propre μ de B , on a $|\mu| \leq \lambda_A$ et l'égalité a lieu si et seulement si $A = B$.*

La valeur propre λ_A est appelée la *valeur propre de Perron* de A . Autrement dit, ce théorème stipule qu'une matrice irréductible possède toujours une valeur propre réelle dominante λ_A . Cependant, on peut avoir **d'autres valeurs propres** de module égal à λ_A mais dans ce cas, celles-ci sont exactement obtenues par multiplication de λ_A par les racines d -ièmes de l'unité⁵.

Démonstration. Puisqu'il s'agit d'un cours de théorie des graphes et pas d'un cours d'algèbre, nous ne donnons pas ici la preuve de ce résultat. Cette preuve n'est pas difficile mais assez longue (elle fait d'ailleurs intervenir des raisonnements de continuité et de passage à la limite). Le lecteur intéressé trouvera une preuve classique dans [26] ou [9]. Une illustration assez géométrique de la preuve dans le cas de la dimension 2 est donnée dans [14]. On trouve aussi une preuve reposant sur la notion élégante de fonctions sous-harmoniques (sorte de généralisation de la notion de vecteur propre)

⁵Remarquons que A étant réel, son polynôme caractéristique χ_A aussi. Si z_0 est racine de χ_A , alors son conjugué $\overline{z_0}$ aussi.

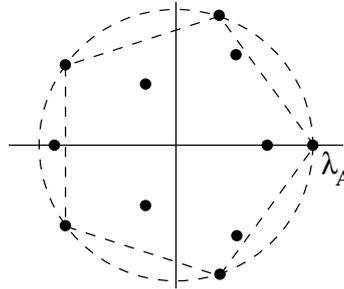


FIGURE II.3. Disposition des valeurs propres d'une matrice irréductible.

dans [11]. Enfin, une preuve détaillée et très bien structurée se trouve dans [1] où la théorie de Perron-Frobenius est utilisée dans le cadre des suites automatiques (fréquence d'apparition d'un symbole dans une suite, etc. . .).

■

Nous pouvons à présent prouver la réciproque de la proposition II.1.9.

Corollaire II.2.6. *Si $G = (V, E)$ est un graphe (non orienté simple) connexe dont le spectre est symétrique par rapport à 0, alors G est biparti.*

Démonstration. Soient λ_G la valeur propre de Perron de G et $x \neq 0$ un vecteur propre associé. Par hypothèse $-\lambda_G$ est aussi une valeur propre de G et considérons $y \neq 0$, l'un de ses vecteurs propres. Bien évidemment, x et y sont linéairement indépendants⁶. Si A est la matrice d'adjacence de G , A^2 est la matrice d'adjacence du multi-graphe $G' = (V, E')$ où une arête $\{a, b\}$ appartient à E' si et seulement si il existe $c \in V$ tel que $\{a, c\}$ et $\{b, c\}$ appartiennent à E . On pourrait appeler G' , le graphe des chemins de longueur 2 de G .

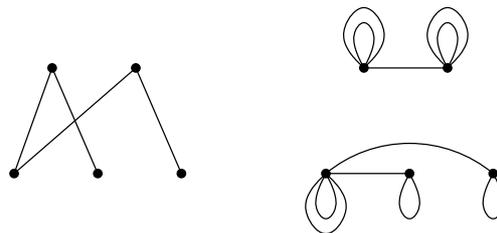


FIGURE II.4. Une illustration des graphes G et G' .

Il est clair que λ_G^2 est la valeur propre dominante de A^2 et que x et y en sont des vecteurs propres. Par conséquent, la multiplicité de λ_G^2 est au moins 2 et on en déduit que A^2 ne peut être irréductible (i.e., G' n'est pas

⁶Des vecteurs propres non nuls associés à des valeurs propres distinctes sont linéairement indépendants.

connexe). Nous allons montrer que G est biparti⁷. En effet, G' contient au moins deux composantes connexes. En fait, il en contient exactement deux. Comme nous allons l'expliquer, cela découle du fait que G est connexe.

Soit u un sommet quelconque fixé. On définit V_1 (resp. V_2) comme l'ensemble des sommets joints à u par un chemin de longueur impaire (resp. paire) dans G . Puisque G est connexe, $V_1 \cup V_2 = V$ (à ce stade, rien ne garantit qu'il s'agisse d'une partition, on pourrait imaginer qu'un sommet v soit joint à u par deux chemins, l'un de longueur paire, l'autre de longueur impaire). Tous les sommets de V_2 joints à u par un chemin de longueur paire dans G sont connectés à u dans G' , cela découle de la définition même de G' . La restriction de G' aux sommets de V_2 est donc connexe. Tous les sommets de V_1 joints à u par un chemin de longueur impaire dans G , sont connectés entre eux dans G' . En effet, considérons $v, w \in V_1$. Dans G , il existe un chemin de longueur impaire entre v_1 et u et aussi un entre u et v_2 . Il existe donc un chemin (passant par u) de longueur paire dans G entre v et w . Autrement dit, la restriction de G' à V_1 est connexe. De plus, $V_1 \cap V_2 = \emptyset$, car sinon G' serait connexe. En raisonnant sur la parité des longueurs de chemins de G , on a donc bien exactement deux composantes connexes V_1 et V_2 dans G' .

Puisque dans G' , il n'y a aucune arête entre un sommet de V_1 et un sommet de V_2 , on en conclut que dans G , il n'y a aucun chemin de longueur paire entre deux tels sommets (en effet, un chemin de longueur 2ℓ dans G se décompose en ℓ chemins de longueur 2, c'est-à-dire en un chemin de longueur ℓ dans G' , ce dernier restant dans une unique composante connexe). Autrement dit, *un chemin joignant dans G un sommet de V_1 et un sommet de V_2 est nécessairement de longueur impaire.*

Supposons à présent que, dans G , un chemin de longueur impaire $2\ell + 1$ joigne deux sommets u_1 et $u_{2\ell+2}$ de V_1 . Par symétrie, le raisonnement qui suit s'applique aussi à deux sommets de V_2 . Ce chemin, déterminé par la suite de sommets $(u_1, u_2, \dots, u_{2\ell+1}, u_{2\ell+2})$, se décompose en ℓ chemins $(u_{2i-1}, u_{2i}, u_{2i+1})$ de longueur 2, $i = 1, \dots, \ell$, plus une arête $\{u_{2\ell}, u_{2\ell+1}\}$. Dans G' , les ℓ chemins de longueur 2 correspondent à des arêtes de la composante connexe V_1 . Donc $u_1, u_3, \dots, u_{2\ell+1}$ appartiennent à V_1 . On en déduit que dans G , il existe alors une arête entre deux sommets $a = u_{2\ell+1}$ et $b = u_{2\ell+2}$ de V_1 . Mais G étant connexe, il existe aussi une arête joignant un sommet c de V_1 et un sommet d de V_2 . De là, on en conclut qu'il existe un chemin de longueur paire joignant a à d (en effet, b et c se trouvent dans la même composante connexe de G' , ils sont donc séparés par un chemin de longueur paire dans G et il suffit d'ajouter les deux arêtes $\{a, b\}$ et $\{c, d\}$). Ceci est en contradiction avec la première partie de notre raisonnement.

⁷Un graphe $G = (V, E)$ est biparti si et seulement si V peut être partitionné en deux sous-ensembles V_1 et V_2 de manière telle que tout chemin joignant un sommet de V_1 à un sommet de V_2 (resp. joignant deux sommets de V_1 ou deux sommets de V_2) soit de longueur impaire (resp. paire).

En conclusion, dans G , les sommets de V_1 (resp. de V_2) sont joints entre eux par des chemins de longueur paire exclusivement. En particulier, il n'y a aucune arête entre deux sommets de V_1 (resp. V_2). De plus, les sommets de V_1 sont joints aux sommets de V_2 par des chemins de longueur impaire exclusivement. Ceci montre que G est biparti. ■

Dans le cas d'une matrice primitive, les assertions du théorème de Perron-Frobenius sont encore renforcées (les trois premières étant inchangées).

Théorème II.2.7 (Perron). *Soit $A \geq 0$ une matrice carrée primitive de dimension n .*

- ▶ *La matrice A possède un vecteur propre $v_A \in \mathbb{R}^n$ (resp. $w_A \in \mathbb{R}^n$) dont les composantes sont toutes strictement positives et correspondant à une valeur propre $\lambda_A > 0$,*

$$A v_A = \lambda_A v_A \text{ (resp. } \widetilde{w}_A A = \lambda_A \widetilde{w}_A \text{)}.$$

- ▶ *Cette valeur propre λ_A possède une multiplicité algébrique (et géométrique) simple.*
- ▶ *Tout vecteur propre de A dont les composantes sont strictement positives est un multiple de v_A .*
- ▶ *Toute autre valeur propre $\mu \in \mathbb{C}$ de A est telle que $|\mu| < \lambda_A$.*
- ▶ *Soit B une matrice réelle à coefficients positifs ou nuls de même dimension que A . Si $B \leq A$, alors pour toute valeur propre μ de B , on a $|\mu| \leq \lambda_A$ et l'égalité a lieu si et seulement si $A = B$.*

Ainsi, la valeur propre de Perron λ_A est l'unique valeur propre dominante. Toute autre valeur propre de A a un module **strictement** inférieur à λ_A . On a donc un résultat plus fort que dans le cas irréductible (c'est assez naturel puisque les hypothèses sont plus fortes).

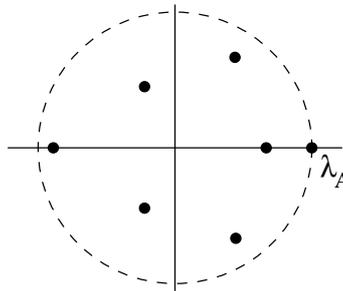


FIGURE II.5. Disposition des valeurs propres d'une matrice primitive.

Traisons à présent deux exemples. Nous y avons choisi des graphes orientés. Des constatations analogues peuvent naturellement être obtenues dans le cas non orienté.

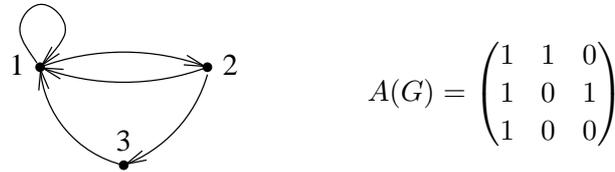


FIGURE II.6. Un graphe avec une matrice d'adjacence primitive.

Exemple II.2.8 (Cas primitif). Considérons le graphe de la figure II.6 et la matrice d'adjacence correspondante. La matrice $A(G)$ est primitive. En effet, on a

$$A(G)^2 = \begin{pmatrix} 2 & 1 & 1 \\ 2 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad \text{et} \quad A(G)^3 = \begin{pmatrix} 4 & 2 & 1 \\ 3 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix} > 0.$$

Il existe donc un chemin de longueur 3 joignant toute paire de sommets. Par exemple, on a les chemins suivants :

$$\begin{aligned} &1 \rightarrow 1 \rightarrow 1 \rightarrow 1, \quad 1 \rightarrow 1 \rightarrow 1 \rightarrow 2, \quad 1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \\ &2 \rightarrow 1 \rightarrow 1 \rightarrow 1, \quad 2 \rightarrow 1 \rightarrow 1 \rightarrow 2, \quad 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \\ &3 \rightarrow 1 \rightarrow 1 \rightarrow 1, \quad 3 \rightarrow 1 \rightarrow 1 \rightarrow 2, \quad 3 \rightarrow 1 \rightarrow 2 \rightarrow 3. \end{aligned}$$

Des valeurs approchées de valeurs propres de $A(G)$ sont

$$\lambda_A \simeq 1.83929, \quad \lambda_2 \simeq -0.41964 + 0.60629i \quad \text{et} \quad \lambda_3 \simeq -0.41964 - 0.60629i$$

et $|\lambda_2| = |\lambda_3| < \lambda_A$.

Exemple II.2.9 (Cas irréductible). Considérons à présent le graphe de la figure II.7 et la matrice d'adjacence correspondante. Il est clair que le

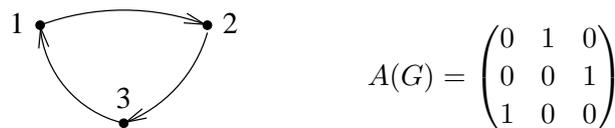


FIGURE II.7. Un graphe avec une matrice d'adjacence irréductible.

graphe est f. connexe et donc, la matrice $A(G)$ est au moins irréductible. Cependant, un chemin de longueur k joint les sommets 1 et 2 si et seulement si un chemin de longueur $k+1$ joint les sommets 1 et 3. Il n'existe donc pas d'entier n pour lequel tout sommet peut être joint à tout autre sommet par un chemin de longueur n . La matrice $A(G)$ n'est donc pas primitive. On pourrait aussi s'en convaincre en montrant que, pour tout n ,

$$A(G)^{3n} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A(G)^{3n+1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad A(G)^{3n+2} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Les valeurs propres sont ici les racines cubiques de l'unité

$$1, e^{2i\pi/3}, e^{4i\pi/3}$$

et donc, on a ici, par opposition avec l'exemple précédent, plusieurs valeurs propres de module maximum (= 1).

Il est aussi intéressant de noter que pour joindre deux sommets fixés, uniquement certaines longueurs de chemin peuvent être considérées. Par exemple, pour joindre les sommets 2 et 3, uniquement des chemins de longueur congrue à 1 modulo 3 peuvent être envisagés. Ce phénomène est en fait tout à fait général et sera explicité à la section suivante.

Corollaire II.2.10. *Soit $A \geq 0$ une matrice carrée. Les assertions suivantes sont équivalentes.*

- i) A est primitive,
- ii) il existe $N \geq 1$ tel que $A^N > 0$,
- iii) il existe $N \geq 1$ tel que $A^n > 0$ pour tout $n \geq N$.

Démonstration. Par définition, i) \Rightarrow ii) et ii) \Rightarrow i). Montrons que ii) \Rightarrow iii). Puisque $A^N > 0$, on en déduit que toute colonne de A contient au moins un élément strictement positif. Par conséquent, si $A^k > 0$, alors $A^k \cdot A > 0$ et de proche en proche, $A^{k+i} > 0$ pour tout $i \geq 0$. Enfin, il est immédiat que iii) \Rightarrow ii). ■

2.1. Période d'une matrice irréductible. Soit $(a_{ij})_{1 \leq i, j \leq d} = A \geq 0$ une matrice carrée de dimension d à coefficients positifs ou nuls. Par *indice* de A , on entend un élément de $\{1, \dots, d\}$.

Définition II.2.11. Soit i un indice. S'il existe $N > 0$ tel que $[A^N]_{i,i} > 0$, alors la *période* de l'indice i est le p.g.c.d. de l'ensemble des entiers $n > 0$ pour lesquels

$$[A^n]_{i,i} > 0.$$

On la note $p(i)$. Bien évidemment, le p.g.c.d. d'un ensemble infini d'entiers $X = \{x_1 < x_2 < \dots\} \subseteq \mathbb{N}$ est le plus grand entier p appartenant à l'ensemble fini $\{1, 2, \dots, x_1\}$ tel que pour tout $k \geq 1$, p divise x_k .

Remarque II.2.12. Cette définition est donnée en termes de matrices, mais elle possède un analogue immédiat en termes de graphes. En effet, à la matrice $A = (a_{ij})_{1 \leq i, j \leq d}$, on fait correspondre un graphe $G_A = (V_A, E_A)$ ayant pour ensemble de sommets $V_A = \{1, \dots, d\}$ et il existe un arc joignant i à j si et seulement si $a_{i,j} > 0$. Ainsi, pour définir la période d'un sommet $i \in V_A$ appartenant à une composante f. connexe du graphe G_A , on recherche le p.g.c.d. de l'ensemble des entiers k pour lesquels il existe au moins un circuit de longueur k passant par i .

Exemple II.2.13. Dans l'exemple II.2.8, on a $[A(G)^n]_{3,3} > 0$ pour tout $n \geq 3$ et le p.g.c.d. des éléments de l'ensemble $X = \{3, 4, 5, \dots\}$ est 1. Ainsi, le sommet 3 est de période 1.

Par contre, dans l'exemple II.2.9, $[A(G)^n]_{i,i} > 0$ si et seulement si n est un multiple de 3. Ainsi, les périodes des sommets de ce graphe sont toutes égales à 3.

Lemme II.2.14. Soient i, j deux indices de $A \geq 0$. S'il existe m, n tels que $[A^m]_{i,j} > 0$ et $[A^n]_{j,i} > 0$, alors $p(i) = p(j)$.

Remarque II.2.15. En utilisant la remarque II.2.12, le lemme se réexprime comme suit. S'il existe dans G_A un chemin de longueur m joignant i et j et un chemin de longueur n joignant j et i , autrement dit si $i \leftrightarrow j$, alors les deux sommets ont même période. (Ou encore, tous les sommets d'une composante f. connexe ont même période.)

Démonstration. Pour tout s tel que $[A^s]_{j,j} > 0$, on a

$$\begin{aligned} [A^{m+s+n}]_{i,i} &= \sum_{k=1}^d [A^{m+s}]_{i,k} [A^n]_{k,i} \\ &\geq [A^{m+s}]_{i,j} [A^n]_{j,i} \\ &= \sum_{k=1}^d [A^m]_{i,k} [A^s]_{k,j} [A^n]_{j,i} \\ &\geq [A^m]_{i,j} [A^s]_{j,j} [A^n]_{j,i} > 0 \end{aligned}$$

Les deux inégalités proviennent du fait que les éléments de A sont positifs ou nuls. Pour un tel s , on a aussi $[A^{2s}]_{j,j} > 0$ (en effet, si on effectue le produit matriciel $[A^s \cdot A^s]_{j,j}$, on retrouve le terme $[A^s]_{j,j} \cdot [A^s]_{j,j}$ et les autres termes de la somme sont positifs ou nuls). Dès lors, on a aussi

$$[A^{m+2s+n}]_{i,i} > 0.$$

Par conséquent, $p(i)$ divise $m + 2s + n$ et $m + s + n$ et aussi leur différence, s . En conclusion, pour tout s tel que $[A^s]_{j,j} > 0$, $p(i)$ divise⁸ s et donc $p(i) \leq p(j)$. Par symétrie, on a aussi que $p(j) \leq p(i)$ et donc $p(i) = p(j)$. ■

Grâce à ce lemme, nous pouvons introduire la définition suivante. En effet, pour une matrice irréductible, toutes les périodes sont nécessairement identiques. En particulier, tous les sommets d'une composante f. connexe de G_A ont même période.

Définition II.2.16. Une matrice irréductible $A \in \mathbb{R}_d^d$ est *cyclique* de période p si tous les indices de A sont de période $p > 1$. Sinon, tous les indices sont de période $p = 1$ et A est dite *acyclique*.

⁸ $p(i)$ est un diviseur commun des éléments de l'ensemble $\{s > 0 \mid [A^s]_{j,j} > 0\}$ et d'autre part, $p(j)$ le p.g.c.d. de cet ensemble.

Lemme II.2.17. *Soit $A \geq 0$ une matrice carrée irréductible de période $p \geq 1$. Soit i un indice de A . Il existe $N_i \geq 0$ tel que pour tout $n \geq N_i$, $[A^{np}]_{i,i} > 0$.*

Démonstration. Supposons tout d'abord que $[A^{kp}]_{i,i} > 0$ et $[A^{\ell p}]_{i,i} > 0$. Dès lors (même raisonnement que dans la preuve du lemme II.2.14),

$$[A^{(k+\ell)p}]_{i,i} \geq [A^{kp}]_{i,i} [A^{\ell p}]_{i,i} > 0.$$

Cela signifie que l'ensemble \mathcal{S} des multiples np de p qui sont tels que $[A^{np}]_{i,i} > 0$ est stable pour l'addition (et \mathcal{S} contient au moins un multiple de p). De plus, par définition, le p.g.c.d. des éléments de \mathcal{S} vaut p . La conclusion découle alors du lemme suivant. ■

Lemme II.2.18. *Soit $X \subseteq \mathbb{N}$ un ensemble d'entiers stable pour l'addition. Alors X contient tous les multiples du p.g.c.d. des éléments de X à l'exception éventuellement d'un nombre fini d'entre eux.*

Démonstration. Soit p le p.g.c.d. des éléments de X . Quitte à diviser les éléments de X par p , on peut supposer que $p = 1$. Dès lors, il existe un ensemble fini⁹ $\{x_1, \dots, x_k\} \subseteq X$ tel que

$$\text{p.g.c.d.} \{x_1, \dots, x_k\} = 1.$$

Par le théorème de Bezout, il existe des entiers relatifs $\lambda_1, \dots, \lambda_k \in \mathbb{Z}$ tels que

$$\lambda_1 x_1 + \dots + \lambda_k x_k = 1.$$

Si on regroupe tous les termes dont les coefficients λ_i sont positifs (resp. négatifs), cette somme se réécrit

$$m - n = 1$$

avec $m, n \in X$ car X est stable pour l'addition. Soit q un entier tel que $q \geq n(n-1)$. Par division euclidienne,

$$q = an + b, \quad 0 \leq b < n.$$

De plus, $a \geq n-1$. Puisque $m - n = 1$, il vient

$$q = an + b(m - n) = (a - b)n + bm$$

⁹Nous savons que le p.g.c.d. de $X = \{x_1 < x_2 < \dots\}$ vaut 1. Si on considère tout d'abord X restreint à $\{x_1\}$ seul, le p.g.c.d. potentiel serait x_1 . Puis, quand on considère $\{x_1, x_2\}$, le p.g.c.d. potentiel ne peut que diminuer (ou au mieux rester constant) puisqu'on doit considérer cette fois les facteurs premiers communs à x_1 et à x_2 . A l'étape suivante, on considère $\{x_1, x_2, x_3\}$ et ainsi de suite (à chaque étape, le p.g.c.d. décroît). Nous affirmons qu'il existe k tel que le p.g.c.d. de $\{x_1, \dots, x_k\}$ soit 1 car si tel n'était pas le cas, le p.g.c.d. de X serait > 1 , ce qui est contraire à notre supposition. Remarquons en particulier que k peut être > 2 , ne serait-ce par exemple qu'avec l'ensemble fini $\{6, 10, 15\}$ dont le p.g.c.d. vaut 1 mais qui est tel que ses éléments pris deux à deux ne sont pas premiers entre eux.

avec $a - b \geq 0$. On en conclut que q appartient à X (car $m, n \in X$). Nous avons donc montré que tout entier $q \geq n(n-1)$ appartient à X . Cela termine la preuve. ■

Le résultat suivant montre que l'exemple II.2.9 est l'archétype même de la situation rencontrée dans le cas d'un graphe irréductible.

Théorème II.2.19. *Soit $A \geq 0$ une matrice carrée irréductible de période $p \geq 1$. Pour toute paire i, j d'indices de A , il existe un unique entier $r_{i,j} \in \{0, \dots, p-1\}$ tel que*

- ▶ $[A^n]_{i,j} > 0$ entraîne $n \equiv r_{i,j} \pmod{p}$ et
- ▶ il existe $N_{i,j}$ tel que $[A^{np+r_{i,j}}]_{i,j} > 0$ pour tout $n \geq N_{i,j}$.

Exemple II.2.20. On peut reprendre l'exemple II.2.9. Dans cet exemple, $p = 3$ et si on fixe le sommet $i = 2$, on a $r_{2,1} = \mathbf{2}$, $r_{2,2} = \mathbf{0}$ et $r_{2,3} = \mathbf{1}$. En effet, $[A(G)^{3n+2}]_{2,1} = 1$, $[A(G)^{3n+0}]_{2,2} = 1$ et $[A(G)^{3n+1}]_{2,3} = 1$.

Démonstration. Supposons que $[A^m]_{i,j} > 0$ et $[A^n]_{i,j} > 0$. Nous allons montrer que $m \equiv n \pmod{p}$. Puisque A est irréductible, il existe ℓ tel que $[A^\ell]_{j,i} > 0$. Dès lors,

$$[A^{m+\ell}]_{i,i} \geq [A^m]_{i,j}[A^\ell]_{j,i} > 0 \quad \text{et} \quad [A^{n+\ell}]_{i,i} > 0.$$

La période p divise donc $m + \ell$ et $n + \ell$ donc leur différence. Autrement dit, $m - n \equiv 0 \pmod{p}$.

Passons à la deuxième partie. Puisque A est irréductible, il existe ℓ tel que $[A^\ell]_{i,j} > 0$ et au vu de la première partie,

$$\ell = mp + r_{i,j}.$$

Posons

$$N_{i,j} = N_i + m$$

(avec N_i donné dans le lemme II.2.17). Par définition de N_i , on a

$$\forall n \geq N_i, [A^{np}]_{i,i} > 0.$$

De là, si $k \geq N_{i,j}$, alors

$$kp + r_{i,j} = (n + m)p + r_{i,j} \quad \text{avec } n \geq N_i.$$

et

$$[A^{kp+r_{i,j}}]_{i,j} \geq [A^{np}]_{i,i}[A^{mp+r_{i,j}}]_{i,j} > 0.$$

■

Proposition II.2.21. *Une matrice irréductible est acyclique si et seulement si elle est primitive.*

Démonstration. Si la matrice est acyclique (i.e., de période $p = 1$), alors avec les notations du théorème II.2.19, $r_{i,j} = 0$ quels que soient les indices i et j . On en conclut que

$$[A^n]_{i,j} > 0 \quad \text{si } n \geq N_{i,j}.$$

Ainsi, si on pose $\mathcal{N} = \sup_{i,j} N_{i,j}$, alors $A^{\mathcal{N}} > 0$ et A d'être primitive.

Réciproquement si A est primitive, A est nécessairement irréductible et pour k suffisamment grand et pour tout indice i de A , $[A^k]_{i,i} > 0$ et $[A^{k+1}]_{i,i} > 0$. Le p.g.c.d. de k et de $k + 1$ étant 1, la conclusion en découle. ■

Exemple II.2.22. Terminons par une application des résultats précédents et considérons les graphes représentés à la figure II.8. Le sommet A appar-

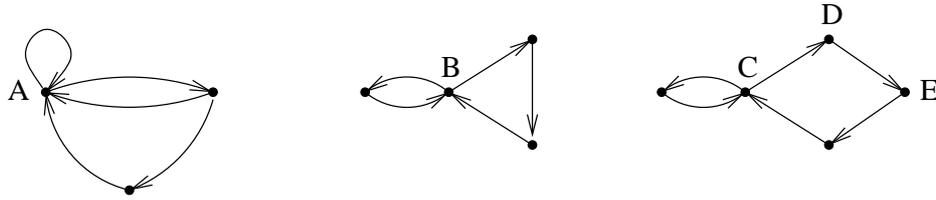


FIGURE II.8. Trois graphes f. connexes.

tient visiblement à un cycle de longueur 1 (une boucle). Par conséquent, sa période vaut 1 et le graphe est primitif au vu de la proposition précédente. Le sommet B appartient quant à lui à un cycle de longueur 2 et à un cycle de longueur 3. Ainsi, le p.g.c.d. des longueurs des cycles passant par B vaut 1 et le graphe est encore une fois primitif. En particulier, cela signifie que pour tout n suffisamment grand, il existe un chemin de longueur n entre tout couple de sommets. Enfin, tout cycle contenant C est de longueur $4p + 2q$, $p, q \geq 0$. On en conclut que la période des différents sommets de ce dernier graphe vaut 2. Pour terminer cet exemple, notons encore que les chemins joignant D à E sont de longueur 1, 5, 7, 9, ... ainsi on dispose d'un chemin entre D et E si et seulement si sa longueur est de la forme $1 + 2n$ avec $n \geq 2$ ce qui illustre parfaitement la seconde partie du théorème II.2.19.

Signalons sans démonstration un dernier résultat, sorte de réciproque au théorème de Perron-Frobenius.

Proposition II.2.23. *Si $A \geq 0$ est une matrice irréductible possédant une valeur propre dominante λ (i.e., pour toute valeur propre $\mu \neq \lambda$ de A , $|\mu| < \lambda$), alors A est primitive.*

2.2. Estimation du nombre de chemins de longueur n . Le théorème de Perron permet de donner le comportement asymptotique¹⁰ du nombre de chemins de longueur n joignant deux sommets quelconques d'un graphe dont la matrice d'adjacence est primitive.

¹⁰Autrement dit, quand $n \rightarrow +\infty$.

Avec les notations du théorème de Perron, si A est une matrice **primitive**, alors il est possible de montrer que ¹¹

$$(3) \quad \boxed{A^k = \lambda_A^k v_A \widetilde{w}_A + o(\lambda_A^k)}$$

où v_A et \widetilde{w}_A sont des vecteurs propres choisis de telle sorte que $\widetilde{w}_A \cdot v_A = 1$. Remarquez que $v_A \widetilde{w}_A$ est une matrice carrée de dimension n dont les éléments sont tous strictement positifs. Autrement dit, ce résultat stipule qu'asymptotiquement, tout élément de A^k est proportionnel à λ_A^k (la constante de proportionnalité dépendant de l'élément envisagé). Tout comme nous avons admis le théorème de Perron, nous admettrons aussi ce résultat sortant du cadre que nous nous sommes fixés dans ce cours.

Remarque II.2.24. Il est même possible d'obtenir des développements plus fins du terme d'erreur en l'exprimant à l'aide de la deuxième valeur propre (par module décroissant) de A . Voir par exemple [26].

Exemple II.2.25. Si on poursuit l'exemple II.2.8, pour tout couple $(i, j) \in \{1, 2, 3\} \times \{1, 2, 3\}$, il existe une constante $d_{i,j} > 0$ telle que le nombre $c_{i,j}(n)$ de chemins de longueur n joignant i à j satisfasse

$$\lim_{n \rightarrow \infty} \frac{c_{i,j}(n)}{d_{i,j} \lambda_A^n} = 1.$$

Remarque II.2.26. Dans le cas d'un graphe f. connexe qui n'est pas primitif, on dispose du théorème de Perron-Frobenius. Néanmoins, si on a plusieurs valeurs propres de module maximum, des compensations entre celles-ci peuvent se produire, et fournir une estimation des $c_{i,j}(n)$ semblable à celle donnée ci-dessus n'est pas si simple. En effet, si on reprend une fois encore l'exemple II.2.9 et que l'on s'intéresse à la suite formée des nombres de chemins de longueur n joignant 1 à 3 pour $n = 0, 1, 2, \dots$, on obtient

$$0, 0, 1, 0, 0, 1, 0, 0, 1, \dots$$

qui est clairement une suite divergente. Ainsi, la limite

$$\lim_{n \rightarrow \infty} \frac{c_{1,3}(n)}{\lambda_A^n}$$

n'existe pas ! Ceci s'explique par le fait que des combinaisons convenables de puissances des racines de l'unité s'annulent¹² :

$$\frac{(e^{2i\pi/3})^n + (e^{4i\pi/3})^n + 1}{3} = 0, \text{ si } n \equiv 1, 2 \pmod{3}.$$

¹¹Pour rappel, une fonction $f(x)$ est en $o(g)$ si f/g tend vers 0 si $x \rightarrow \infty$. Cela se lit "f est en petit oh de g".

¹²Par un résultat classique d'algèbre linéaire (conséquence de la théorie de la diagonalisation ou de la mise sous forme normale de Jordan), si $\lambda_1, \dots, \lambda_p$ sont les valeurs propres distinctes d'une matrice M et sont zéros de son polynôme minimum de multiplicité m_1, \dots, m_p , alors pour tout entier k ,

$$(M^k)_{i,j} = \sum_{t=1}^p P_{i,j}^{(t)} \lambda_t^k$$

2.3. Cas d'un graphe ayant plusieurs composantes f. connexes.

Nous pouvons obtenir des résultats plus fins que ceux obtenus ci-dessus. On considère le condensé \mathcal{C} d'un graphe G (ou graphe acyclique des composantes cf. définition I.4.27) dont les sommets sont les composantes f. connexes de G . Puisque le condensé est sans cycle, on peut ordonner ses sommets par tri topologique. Supposons disposer d'un tel ordonnancement. La matrice d'adjacence de \mathcal{C} a alors une forme triangulaire supérieure et si on ordonne les éléments de G en considérant les sommets des composantes f. connexes prises une à une en respectant le tri topologique, la matrice d'adjacence de G est alors une matrice bloc triangulaire supérieure.

Exemple II.2.27. Poursuivons l'exemple I.4.29. Pour celui-ci, on obtient la matrice suivante si on considère l'ordre donné à la figure I.38.

$$A(G) = \begin{pmatrix} \boxed{\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{matrix}} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \boxed{\begin{matrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{pmatrix}$$

Remarque II.2.28. Au vu de l'exemple précédent, il est immédiat d'observer que le spectre d'un graphe est l'union des spectres de ses composantes connexes.

Nous nous restreignons une fois encore au cas où les composantes f. connexes sont des sous-graphes **primitifs**.

Soit un graphe fini possédant deux composantes f. connexes primitives A et B et deux sommets $a \in A$ et $b \in B$ tels que $a \rightarrow b$. Les chemins joignant un sommet de A à un sommet de B sont en nombre fini et de longueur bornée. Si λ_A et λ_B sont les valeurs propres de Perron de A et de B respectivement, on en déduit que le nombre $c_{a,b}(n)$ de chemins de longueur n joignant a à b est *proportionnel*¹³ à

$$\sum_{i=0}^n \lambda_A^i \lambda_B^{n-i} = \lambda_B^n \sum_{i=0}^n \left(\frac{\lambda_A}{\lambda_B}\right)^i.$$

où $P_{i,j}^{(t)}$ est un polynôme de degré $< m_t$. Ainsi, lorsqu'on considère un quotient comme $(M^k)_{i,j}/\lambda_M^k$, ne subsistent à la limite que les valeurs propres de module maximum. Dans l'exemple II.2.9, n'apparaissent que les trois racines de l'unité.

¹³Dans une estimation asymptotique, c'est le comportement général qui est intéressant. La détermination précise de la constante multiplicative nous importe peu.

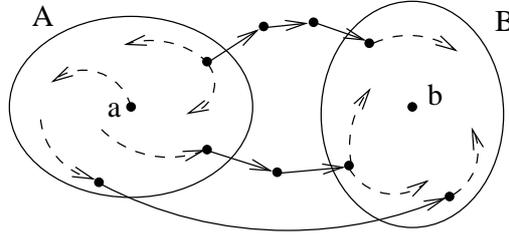


FIGURE II.9. Deux composantes f. connexes.

- Si $\lambda_A = \lambda_B$, alors¹⁴

$$c_{a,b}(n) \asymp n\lambda^n$$

- Sinon, $\lambda_A \neq \lambda_B$ et

$$\sum_{i=0}^n \lambda_A^i \lambda_B^{n-i} = \frac{\lambda_A^{n+1} - \lambda_B^{n+1}}{\lambda_A - \lambda_B} \asymp [\max(\lambda_A, \lambda_B)]^n.$$

Nous détaillons à présent le cas d'un graphe fini ayant trois composantes f. connexes A , B et C . Nous désirons estimer le nombre $c_{a,B,c}(n)$ de chemins de longueur n joignant $a \in A$ à $c \in C$ en passant par un sommet quelconque de B . Le lecteur pourra aisément adapter les raisonnements au cas général.

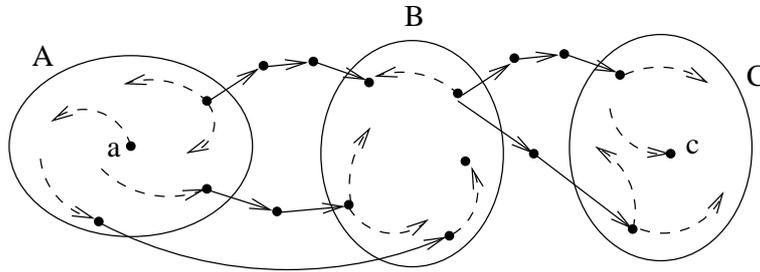


FIGURE II.10. Trois composantes f. connexes.

Avec les mêmes notations que ci-dessus, puisque le nombre de sommets de B est fini, le nombre de chemins recherché est *proportionnel* à

$$(4) \quad \sum_{i=0}^n \sum_{j=0}^{n-i} \lambda_A^i \lambda_B^j \lambda_C^{n-i-j}.$$

Nous traitons trois cas.

- Si $\lambda_A = \lambda_B = \lambda_C$, alors (4) devient

$$\lambda_A^n \sum_{i=0}^n (n+1-i) = \lambda_A^n \left[(n+1)^2 - \frac{n(n+1)}{2} \right] \asymp n^2 \lambda_A^n.$$

¹⁴On a $f \asymp g$ si et seulement si il existe deux constantes α et β telles que $\alpha g \leq f \leq \beta g$.

- Si $\lambda_A = \lambda_B \neq \lambda_C$ (par symétrie, les autres cas se traitent de la même façon), alors (4) devient

$$\sum_{i=0}^n \lambda_A^i \underbrace{\sum_{j=0}^{n-i} \lambda_A^j \lambda_C^{n-i-j}}_{\frac{\lambda_A^{n-i+1} - \lambda_C^{n-i+1}}{\lambda_A - \lambda_C}} = \frac{(n+1)\lambda_A^{n+1}}{\lambda_A - \lambda_C} - \frac{\lambda_C}{\lambda_A - \lambda_C} \underbrace{\sum_{i=0}^n \lambda_A^i \lambda_C^{n-i}}_{\frac{\lambda_A^{n+1} - \lambda_C^{n+1}}{\lambda_A - \lambda_C}}.$$

Ainsi, si $\lambda_A = \lambda_B > \lambda_C$, on trouve

$$c_{a,B,c}(n) \asymp n\lambda_A^n$$

et si $\lambda_A = \lambda_B < \lambda_C$, on trouve

$$c_{a,B,c}(n) \asymp \lambda_C^n.$$

- Enfin, si λ_A, λ_B et λ_C sont 2 à 2 distincts, alors (4) devient

$$\sum_{i=0}^n \lambda_A^i \underbrace{\sum_{j=0}^{n-i} \lambda_B^j \lambda_C^{n-i-j}}_{\frac{\lambda_B^{n-i+1} - \lambda_C^{n-i+1}}{\lambda_B - \lambda_C}} = \frac{1}{\lambda_B - \lambda_C} \left(\lambda_B \sum_{i=0}^n \lambda_A^i \lambda_B^{n-i} - \lambda_C \sum_{i=0}^n \lambda_A^i \lambda_C^{n-i} \right)$$

et on obtient

$$c_{a,B,c}(n) \asymp [\max(\lambda_A, \lambda_B, \lambda_C)]^n.$$

En conclusion, il suffit de détecter la plus grande valeur de Perron λ des différentes composantes connexes par lesquelles passent les chemins d'intérêt et de compter le nombre k de composantes ayant cette valeur propre comme valeur dominante. Le nombre de chemins de longueur n se comporte alors asymptotiquement comme $n^{k-1}\lambda^n$. Le lecteur adaptera facilement les développements effectués ci-dessus au cas d'un nombre arbitraire de composantes primitives. On pourrait aussi regarder la forme de Jordan de la matrice d'adjacence associée, un bloc de taille k associé à la valeur propre λ fait intervenir un polynôme de degré $k - 1$ multiplié par une exponentielle λ^n .

3. Une application : l'algorithme de PageRank

Dans cette section, nous allons illustrer certainement l'une des applications les plus célèbres du théorème de Perron-Frobenius : le succès de Google basé sur leur algorithme de classement des pages web ! Volontairement, plusieurs aspects ne sont pas pris en compte dans cette présentation (développement et efficacité des méthodes numériques, choix de la constante α , mise en pratique, sensibilité aux perturbations, modèle plus évolué, ...). Pour plus de détails, consulter par exemple l'excellent ouvrage de C. Meyer et A. Langville consacré entièrement à ce sujet.

Lorsqu'on effectue une recherche sur un mot clé donné, Google trie les pages contenant ce mot clé en se basant (notamment¹⁵) sur une mesure, appelée "*PageRank*", destinée à quantifier la qualité des pages et à déterminer si elles font ou non autorité dans le domaine envisagé. L'obtention d'un "bon" classement des pages sur un sujet donné est d'ailleurs l'une des raisons pour lesquelles Google est si populaire.

L'idée originale de L. Page et S. Brin¹⁶ est très simple :

- ▶ on accorde plus d'importance, i.e., un score de "*PageRank*" plus élevé, aux pages référencées par des pages qui font elles-mêmes autorité dans le domaine, c'est-à-dire qui ont un *PageRank* élevé;
- ▶ on accorde d'autant moins de crédit à une citation si elle provient d'une page qui dispose de nombreux liens (on ne peut qu'accorder moins de poids aux sites qui galvaudent leurs recommandations).

On suppose disposer d'un graphe simple $G = (V, E)$ où les sommets notés $1, \dots, n$ représentent les pages de l'Internet (en 2005, on recensait plus de huit milliards de pages) et où l'on dispose d'un arc (i, j) si et seulement si la page i possède un lien pointant vers la page j . Au vu du modèle proposé par Page et Brin, le *PageRank* $\pi_j \geq 0$ de la page $j \in \{1, \dots, n\}$ serait donné par

$$(5) \quad \pi_j = \sum_{i \in \text{pred}(j)} \frac{\pi_i}{d^+(i)}$$

qui est une formule récursive pour laquelle on ne dispose pas *a priori* de méthode permettant d'assurer l'existence, l'unicité ou le calcul efficace d'une solution $\pi = (\pi_1, \dots, \pi_n)$ non triviale. On peut de plus supposer que les scores recherchés sont normalisés,

$$\sum_{i=1}^n \pi_i = 1.$$

Le problème peut se réécrire sous forme matricielle ("*H*" comme "hyperlien"),

$$(6) \quad \pi = \pi H$$

où

$$H_{ij} = \begin{cases} A(G)_{ij}/d^+(i) & \text{si } d^+(i) > 0 \\ 0 & \text{si } d^+(i) = 0 \end{cases}$$

avec $A(G)$ la matrice d'adjacence du graphe G et rappelons que π est un vecteur ligne. Sous cette forme, il n'est pas clair que le problème possède une solution (ni qu'elle soit unique et on dispose encore moins d'une méthode

¹⁵Nous ne parlons pas ici des problèmes d'indexation de pages web, de la recherche par mots clés, etc... Nous nous concentrons uniquement sur le classement des pages sélectionnées comme apparentées à la recherche. Cette méthode de classement a d'ailleurs été appliquée à d'autres situations (championnats sportifs par exemple).

¹⁶Pour l'anecdote, il faut savoir que le père de Sergeï Brin, Michael Brin est un mathématicien spécialiste des systèmes dynamiques.

α	nombre d'itérations
0,5	34
0,75	81
0,8	104
0,85	142
0,9	219
0,95	449
0,99	2292
0,999	23015

TABLE II.1. Rôle du paramètre α

de calcul). Autrement dit, il n'y a pas de raison évidente pour laquelle 1 est valeur propre de H et qui plus est, valeur propre simple.

L'astuce pour pouvoir appliquer la théorie de Perron-Frobenius est double. Tout d'abord, pour se débarrasser des "puits", i.e., des pages ne pointant vers aucune autre page¹⁷ et pour obtenir une matrice stochastique, on introduit une matrice S ("S" comme "stochastique") définie par

$$S_{ij} = \begin{cases} A(G)_{ij}/d^+(i) & \text{si } d^+(i) > 0 \\ 1/n & \text{si } d^+(i) = 0. \end{cases}$$

Ensuite, pour assurer la forte connexité du graphe (i.e., le caractère irréductible), on construit une matrice G ("G" comme Google) donnée par la combinaison affine (et même convexe) suivante avec un réel $\alpha \in [0, 1]$ fixé,

$$G = \alpha S + (1 - \alpha) J/n$$

où $J = (1)_{1 \leq i, j \leq n}$. L'équation initiale (6) est remplacée par

$$\pi = \pi G.$$

(La matrice J/n est parfois appelée *matrice de téléportation*, cf. remarque suivante.) Le lecteur préférant manipuler des vecteurs propres à droite écrira $\tilde{\pi} = \tilde{G} \tilde{\pi}$. Google attribue à α une valeur de 0,85. Ce choix n'est pas arbitraire. Au plus α est proche de 1, au mieux on approche le modèle "naturel" (6) proposé initialement : on diminue le rôle artificiel de la matrice de téléportation. Cependant, on peut montrer que ce paramètre α contrôle la vitesse de convergence de la méthode de calcul développée et donc le nombre d'itérations à effectuer pour obtenir une estimation du vecteur π (par calcul de puissances successives de G qui contient plus de 8.10^9 entrées). Quand α tend vers 1, ce nombre devient prohibitif comme le montre la table II.1 (tirée de l'ouvrage de C. Meyer et A. Langville). Ainsi, $\alpha = 0,85$ semble un bon compromis entre le caractère artificiel introduit par la matrice de téléportation et la masse de calculs à réaliser. De plus, on peut

G est un point du segment $[G, J/n]$ de l'espace des matrices.

Monsieur Spock !

¹⁷Et dans l'ensemble des liens de l'Internet, il y a de nombreux puits; imaginez tous les liens pointant vers un fichier (image, vidéo, pdf, etc...)

montrer par une discussion plus fine sur les valeurs propres des matrices envisagées qu'au plus α est proche de 1, au plus le vecteur π est sensible aux petites perturbations de la matrice H (ce qui s'avère être gênant vu la grande volatilité du web et de sa structure, de nombreux liens apparaissent et disparaissent chaque jour). Idéalement, on désirerait obtenir une mesure peu sensible à de telles perturbations.

Remarque II.3.1. On peut donner une interprétation “probabiliste” de la matrice G . Considérons un surfeur qui, se trouvant sur une page quelconque, a deux choix possibles. Soit, avec une probabilité α , il clique au hasard et de manière uniforme sur l'un des liens de la page pour changer de page. Soit, avec une probabilité $1 - \alpha$, il choisit au hasard et de manière uniforme l'une des n pages de l'Internet tout entier. Ainsi, G_{ij} représente la probabilité de transition lorsque le surfeur se trouve sur la page i de passer à la page j .

Ainsi, partant d'une distribution initiale de probabilités, par exemple $\pi^{(0)} = (\frac{1}{n}, \dots, \frac{1}{n})$, l'application de G^k permet d'estimer la probabilité de notre surfeur de se trouver sur l'une des pages $1, \dots, n$ après k clics,

$$\pi^{(k)} = \pi^{(0)} G^k.$$

Nous allons à présent montrer que l'utilisation de cette matrice G (à la place de H) assure l'existence et l'unicité d'une distribution limite π .

Remarque II.3.2. Par rapport à l'équation initiale (5), l'emploi de la matrice G donne la formule suivante pour la détermination des “nouveaux” π_j qui seront effectivement calculés. On utilise la définition de S_{ij} pour obtenir

$$\begin{aligned} \pi_j &= \sum_{i=1}^n \pi_i \left(\alpha S_{ij} + (1 - \alpha) \frac{1}{n} \right) \\ &= \alpha \sum_{i \in \text{pred}(j)} \frac{\pi_i}{d^+(i)} + \frac{1}{n} \left(1 - \alpha + \alpha \sum_{i: d^+(i)=0} \pi_i \right). \end{aligned}$$

Nous nous sommes donc éloignés quelque peu du modèle initialement proposé mais nous allons voir que ces modifications vont permettre un calcul efficace (en assurant de plus l'existence et l'unicité d'une solution !). On observe que le premier terme, à une constante multiplicative près, est exactement (5). Alors que le second terme est une constante¹⁸ < 1 multipliée par $1/n$ (avec n grand).

Proposition II.3.3. *Les matrices S , J/n et G sont stochastiques, autrement dit, la somme des éléments d'une ligne quelconque vaut 1.*

Démonstration. Il s'agit d'une simple vérification. ■

Lemme II.3.4. *Si M est une matrice stochastique, alors 1 est valeur propre dominante de M .*

¹⁸La somme porte sur tous les puits et est donc indépendante de j .

Démonstration. Soit $M \in \mathbb{Q}^r$. En multipliant tous les éléments de M par le p.p.c.m. γ des dénominateurs des éléments de M , la matrice

$$M' = \gamma M$$

obtenue est telle que la somme des éléments de chaque ligne vaut $\gamma \in \mathbb{N}$. Il s'agit donc de la matrice d'adjacence d'un graphe γ -régulier. Comme nous le verrons, il suffit d'appliquer la proposition II.4.3 pour voir que γM possède γ comme valeur propre dominante (i.e., toute autre valeur propre μ est telle que $|\mu| \leq \gamma$). La conclusion suit aisément en divisant par γ . ■

Par construction, la matrice G est primitive puisque toutes ses entrées sont strictement positives. On peut appliquer le théorème de Perron, or par le lemme précédent, nous savons déjà que 1 est valeur propre dominante de G . Par conséquent, la valeur propre dominante 1 est simple et il existe un unique vecteur colonne $x > 0$ (resp. ligne $y > 0$) tel que

$$\sum_{i=1}^n x_i = 1 \quad (\text{resp.} \quad \sum_{i=1}^n y_i = 1) \quad \text{et} \quad Gx = x \quad (\text{resp.} \quad yG = y).$$

Ainsi, déterminer le vecteur π des "PageRanks" revient à chercher le vecteur propre y de Perron à gauche de G (ou le vecteur propre de Perron à droite de \tilde{G}). En appliquant le résultat asymptotique (3) énoncé à la page 84, puisque $e = (1 \cdots 1)$ est un vecteur propre à droite de G de valeur propre 1 (G est stochastique) et que $\pi \cdot e = 1$ (puisque les scores sont normalisés), on a que

$$(7) \quad G^k = e\pi + o(1) \quad \text{autrement dit,} \quad \lim_{k \rightarrow \infty} G^k = e\pi = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (\pi_1 \quad \cdots \quad \pi_n).$$

Nous pouvons à présent obtenir aisément une méthode itérative pour estimer π . Soit $p^{(0)} = (p_1^{(0)} \quad \cdots \quad p_n^{(0)}) > 0$ un vecteur tel que $\sum_i p_i^{(0)} = 1$. Pour tout $k \geq 1$, on pose $p^{(k)} = p^{(0)}G^k = p^{(k-1)}G$. Il nous reste à montrer que

$$\lim_{k \rightarrow \infty} p^{(k)} = \pi$$

ainsi, il suffit dès lors de partir d'une distribution initiale et d'appliquer G de manière itérative jusqu'à obtenir la précision voulue mesurée par $p^{(k)} - p^{(k-1)}$ (différence en norme, bien entendu). C'est immédiat, au vu de (7),

$$\lim_{k \rightarrow \infty} G^k = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_n \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_n \end{pmatrix} =: P$$

et

$$[p^{(0)}P]_j = \sum_{i=1}^n p_i^{(0)} \pi_j = \pi_j \underbrace{\sum_{i=1}^n p_i^{(0)}}_{=1} = \pi_j.$$

Remarque II.3.5. En pratique, une centaine d'itérations suffisent pour obtenir une approximation utilisable et ce calcul peut être réalisé hors ligne, par exemple, une fois par mois, pour mettre à jour le vecteur des scores.

Remarque II.3.6. En pratique, on se ramène à la matrice creuse H (possédant de nombreux zéros) :

$$\begin{aligned}
 p^{(k)} &= p^{(k-1)}G \\
 &= p^{(k-1)}\left(\alpha S + (1 - \alpha)\frac{J}{n}\right) \\
 &= \alpha p^{(k-1)}S + (1 - \alpha)\frac{\tilde{e}}{n} \\
 &= \alpha p^{(k-1)}\left(H + a\frac{\tilde{e}}{n}\right) + (1 - \alpha)\frac{\tilde{e}}{n} \\
 &= \alpha p^{(k-1)}H + (\alpha p^{(k-1)}a + (1 - \alpha))\frac{\tilde{e}}{n}
 \end{aligned}$$

où

$$a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

est tel que $a_i = 1$ si $d^+(i) = 0$ et $a_i = 0$ si $d^+(i) > 0$.

4. Algèbre d'adjacence

Le but principal de cette section est de fournir quelques résultats élémentaires à propos des graphes réguliers.

Soit $G = (V, E)$, un multi-graphe non orienté. On dénote par \mathcal{A}_G l'algèbre¹⁹ (sur \mathbb{C}) des polynômes en la matrice d'adjacence de G . On la dénomme l'*algèbre d'adjacence* de G .

Proposition II.4.1. *Soit $G = (V, E)$, un multi-graphe non orienté connexe ayant $A(G)$ comme matrice d'adjacence et \mathcal{A}_G comme algèbre d'adjacence. Si $\text{diam}(G) = k$, alors la dimension de \mathcal{A}_G est supérieure ou égale à $k + 1$.*

Démonstration. Soient a et b deux sommets réalisant le diamètre de G , i.e., tels que $d(a, b) = k$. Soit

$$\{a = v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{k-1}, v_k = b\}$$

un chemin de longueur k réalisant cette distance (cela implique en particulier que les v_i sont tous distincts). Par définition même du diamètre, pour tout $i \in \{1, \dots, k\}$, il existe un chemin de longueur i joignant v_0 à v_i mais aucun chemin de longueur inférieure. Par conséquent, pour tout $i \in \{1, \dots, k\}$, A^i (resp. A^j , $0 \leq j < i$) contient une entrée strictement positive (resp. nulle) pour l'entrée correspondant à v_0, v_i . On en conclut que I, A, \dots, A^i sont

¹⁹Si $A = A(G)$, un polynôme de A est une expression de la forme $\sum_{k=1}^t \lambda_k A^k$. Il est clair que l'ensemble de ces polynômes est un espace vectoriel sur \mathbb{C} et que le produit de deux polynômes en A est encore un polynôme en A . Il s'agit donc d'une algèbre avec unité. Les vérifications sont immédiates à mener.

linéairement indépendants et que \mathcal{A}_G contient donc au moins $k + 1$ éléments linéairement indépendants. ■

Soit $G = (V, E)$, un multi-graphe non orienté connexe ayant $A(G)$ comme matrice d'adjacence. Il est clair que puisque $A(G)$ est diagonalisable, son polynôme minimum ne possède que des zéros simples et son degré est donc égal au nombre de valeurs propres distinctes de G . Remarquons aussi que si le polynôme minimum de $A(G)$ est de degré k , il existe alors une relation linéaire liant $A(G)^k, A(G)^{k-1}, \dots, I$. Par conséquent, la dimension de \mathcal{A}_G ne peut dépasser k .

Corollaire II.4.2. *Soit $G = (V, E)$, un multi-graphe non orienté connexe ayant $A(G)$ comme matrice d'adjacence et \mathcal{A}_G comme algèbre d'adjacence. Si $\text{diam}(G) = k$, alors G a au moins $k + 1$ valeurs propres distinctes.*

Rappelons qu'un graphe non orienté est k -régulier si pour tout sommet v , $\deg(v) = k$. Une fois encore, ces graphes possèdent des propriétés algébriques importantes.

Proposition II.4.3. *Soit $G = (V, E)$ un multi-graphe non orienté k -régulier. Alors*

- ▶ k est une valeur propre de G ,
- ▶ pour toute valeur propre λ de G , on a $|\lambda| \leq k$,
- ▶ si G est connexe, k est valeur propre simple (i.e., les multiplicités géométrique et algébrique valent 1).

Remarque II.4.4. Cette proposition peut aussi se réexprimer en termes de multi-graphes orientés k -réguliers. Il suffit de préciser au troisième point que G est fortement connexe.

Démonstration. Le premier point est immédiat. Il est clair que le vecteur $(1, \dots, 1)$ est un vecteur propre de $A(G)$ de valeur propre k .

Passons au deuxième point et considérons une valeur propre λ de $A(G)$ ayant $y \neq 0$ comme vecteur propre. Soit y_j une composante de y de module maximum. On a

$$|\lambda| |y_j| = |[A(G)y]_j| \leq \sum_{i=1}^n [A(G)]_{j,i} |y_i| \leq |y_j| \sum_{i=1}^n [A(G)]_{j,i} = k |y_j|$$

et donc, $|\lambda| \leq k$. On a utilisé le fait que les coefficients $[A(G)]_{j,i}$ sont ≥ 0 , qu'au plus k d'entre eux sont non nuls et que $\sum_{i=1}^n [A(G)]_{j,i} = k$.

Pour le dernier point, puisque G est connexe, $A(G)$ est irréductible et on peut donc utiliser le théorème de Perron-Frobenius. La matrice $A(G)$ possède une unique valeur propre réelle dominante et au vu du point précédent, il s'agit de k . ■

Tester la régularité d'un graphe peut se faire grâce à l'algèbre d'adjacence du graphe. On dénote par J_n , ou simplement J , la matrice carrée dont tous les éléments sont égaux à 1, $J_n = (1)_{1 \leq i, j \leq n}$. Il est clair que si on dispose d'un graphe k -régulier, alors

$$A(G).J = kJ = J.A(G).$$

Théorème II.4.5 (Hoffman (1963)). *La matrice J appartient à l'algèbre d'adjacence \mathcal{A}_G si et seulement si G est un multi-graphe (non orienté) connexe et régulier.*

Démonstration. Supposons tout d'abord que J appartienne à \mathcal{A}_G . Puisque J est un polynôme de $A(G) = A$, il est clair²⁰ que $AJ = JA$. On a

$$(AJ)_{i,j} = \deg(v_i) \quad \text{et} \quad (JA)_{i,j} = \deg(v_j).$$

Par conséquent, tous les sommets ont même degré et le graphe est régulier. Il nous reste à vérifier que G est connexe. S'il ne l'était pas, il existerait deux sommets v_i et v_j qui ne seraient joints par aucun chemin de longueur $\ell \geq 0$. Autrement dit, $[A(G)^\ell]_{i,j} = 0$ pour tout $\ell \geq 0$. Cette relation restant valable pour tout polynôme en $A(G)$, on en concluerait que J ne peut appartenir à \mathcal{A}_G .

La condition est suffisante. Supposons G connexe et k -régulier. Ainsi, par la proposition précédente, puisque G est k -régulier, k est une valeur propre de $A(G)$ et le polynôme minimum de $A(G)$ est de la forme²¹ $\mathcal{M}(\lambda) = (\lambda - k)q(\lambda)$ avec $q(k) \neq 0$. En évaluant ce polynôme en A , on obtient

$$Aq(A) = kq(A).$$

Autrement dit, chaque colonne de $q(A)$ est un vecteur propre de A de valeur propre k . Puisque G est connexe, k est une valeur propre simple et on en conclut que chaque colonne de $q(A)$ est un multiple de $(1, \dots, 1)^\top$. Puisque A est symétrique, $q(A)$ aussi et donc toutes les colonnes de $q(A)$ sont égales à un même multiple de $(1, \dots, 1)^\top$. Autrement dit, $q(A)$ est un multiple de J et cette dernière matrice appartient donc à \mathcal{A}_G . ■

Nous terminons cette section par un corollaire de nature algébrique.

Corollaire II.4.6. *Soit G un multi-graphe non orienté connexe et k -régulier possédant n sommets et ayant $k = \lambda_1 > \lambda_2 > \dots > \lambda_s$ comme valeurs propres distinctes. Si*

$$q(\lambda) = \prod_{i=2}^s (\lambda - \lambda_i),$$

²⁰Deux polynômes d'une même matrice commutent toujours.

²¹Puisque A est diagonalisable, son polynôme minimum ne possède que des zéros simples. On en déduit que q est le produit de facteurs $(\lambda - \lambda_i)$ tous distincts, où les λ_i sont les valeurs propres de G distinctes de k .

alors

$$J = \frac{n}{q(k)} q(A(G)).$$

Démonstration. Au vu de la preuve précédente, $q(A(G)) = \alpha J$. Les valeurs propres de $q(A)$ sont²² les $q(\lambda_i)$, $i = 1, \dots, s$. Si $i \neq 1$, alors $q(\lambda_i) = 0$. Ainsi, $q(A(G))$ a comme seule valeur propre non nulle $q(k)$. De plus, αJ a pour valeur propre αn (il suffit de considérer le vecteur propre $(1, \dots, 1)$). On en conclut que $q(k) = \alpha n$ ou encore que $1/\alpha = n/q(k)$. ■

5. Arbres couvrants

Revenons à présent au problème présenté dans l'exemple I.3.8. Il est clair que ce problème revient à chercher dans un graphe (simple²³ et non orienté) connexe un sous-arbre couvrant de poids minimal. (Etant entendu que le poids d'un graphe pondéré vaut la somme des poids de ses arêtes.)

Tout d'abord, nous allons considérer le problème plus simple consistant à rechercher un sous-arbre couvrant. Ce problème est immédiatement réglé par une *recherche en profondeur*²⁴. L'idée est de rechercher autant que faire se peut, des voisins successifs d'un sommet donné.

Algorithme II.5.1 (Sous-arbre couvrant). La donnée de l'algorithme est un graphe connexe non orienté $G = (V, E)$.

Choisir un sommet $v_0 \in V$

Visite := $\{v_0\}$, A := \emptyset

$t := 0$, $j := 1$

Tant que Visite $\neq V$, répéter

 Peut-on choisir un élément $v_j \in \nu(v_t) \setminus \text{Visite}$?

 Si un tel choix est impossible,

 il existe $k < t$ maximum tel que $\{v_k, v_t\} \in E$ et poser $t := k$

 Si un tel choix est possible,

 Visite := $\{v_0, \dots, v_j\}$, A := $A \cup \{v_t, v_j\}$, $t := j$ et $j := j + 1$

Ainsi, la variable **Visite** mémorise les sommets déjà visités et la variable **A** mémorise quant à elle les arêtes sélectionnées. Lorsque l'algorithme s'achève, **A** correspond à un sous-arbre couvrant de G . Les justifications sont laissées au lecteur.

Exemple II.5.2. Voici une application de l'algorithme précédent au graphe représenté à la figure II.11.

Pour un graphe donné G , nous savons trouver un sous-arbre couvrant mais bien évidemment un tel arbre n'est pas nécessairement unique. Notre

²²Se rappeler les résultats sur les polynômes d'endomorphismes.

²³En effet, inutile de considérer des arêtes multiples ou des boucles dans ce type de problème.

²⁴*Depth First Search*.

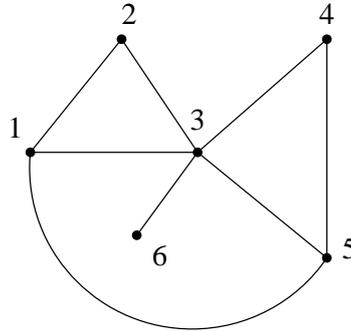


FIGURE II.11. Recherche d'un sous-arbre couvrant.

Visite	A	t	j
$\{v_0 = 1\}$	\emptyset	0	1
$\{\dots, v_1 = 5\}$	$\{\{1, 5\}\}$	1	2
$\{\dots, v_2 = 3\}$	$\{\{1, 5\}, \{5, 3\}\}$	2	3
$\{\dots, v_3 = 4\}$	$\{\{1, 5\}, \{5, 3\}, \{3, 4\}\}$	3	4
$\{\dots, v_3 = 4\}$	$\{\{1, 5\}, \{5, 3\}, \{3, 4\}\}$	2	4
$\{\dots, v_4 = 2\}$	$\{\{1, 5\}, \{5, 3\}, \{3, 4\}, \{3, 2\}\}$	4	5
$\{\dots, v_4 = 2\}$	$\{\{1, 5\}, \{5, 3\}, \{3, 4\}, \{3, 2\}\}$	2	5
$\{\dots, v_5 = 6\}$	$\{\{1, 5\}, \{5, 3\}, \{3, 4\}, \{3, 2\}, \{3, 6\}\}$	5	6

TABLE II.2. Application de l'algorithme II.5.1.

but est à présent de pouvoir compter le nombre $\tau(G)$ de tels sous-arbres. Pour répondre à ce problème de dénombrement, plusieurs résultats intermédiaires sont nécessaires. Néanmoins, on peut d'abord obtenir assez facilement une formule récursive permettant de répondre (de manière assez fastidieuse) à la question.

5.1. Une formule de Cayley. Définissons tout d'abord une opération de contraction d'un sommet. Celle-ci va nous permettre de donner une formule récursive pour obtenir $\tau(G)$.

Définition II.5.3. Soient $G = (V, E)$ un multi-graphe (non orienté) et e une arête de G . Le graphe obtenu en supprimant l'arête e et en identifiant les extrémités de celle-ci est appelé *contraction* de G (pour l'arête e) et se note $G \cdot e$.

Si G est connexe et si e est une arête qui n'est pas une boucle, il en va de même de sa contraction $G \cdot e$ qui contient une arête et un sommet de moins que G .

Proposition II.5.4. Si e est une arête (qui n'est pas une boucle) d'un multi-graphe connexe (non orienté), alors

$$\tau(G) = \tau(G - e) + \tau(G \cdot e).$$

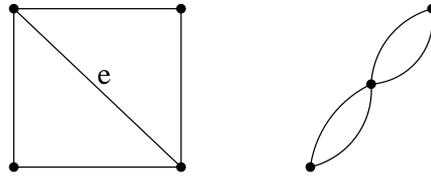


FIGURE II.12. Contraction d'un graphe G en $G \cdot e$.

Ainsi, cette formule permet d'exprimer $\tau(G)$ à l'aide de deux graphes plus simples. On arrête la procédure lorsque la suppression d'une arête rend le graphe non connexe (autrement dit, lorsqu'on a un "pseudo"-arbre ne tenant pas compte des éventuelles boucles).

Démonstration. Tout sous-arbre couvrant de G qui ne contient pas e est aussi un sous-arbre couvrant de $G - e$. Ainsi, $\tau(G - e)$ compte tous les sous-arbres couvrants de G qui ne contiennent pas e .

A chaque sous-arbre couvrant A de G qui contient e , il correspond un sous-arbre couvrant $A \cdot e$ de $G \cdot e$ et cette correspondance est une bijection. Ainsi, le nombre de sous-arbres couvrants de G qui contiennent e vaut exactement $\tau(G \cdot e)$.

■

Exemple II.5.5. Si on applique la formule de Cayley au graphe suivant de gauche sur la figure II.12, on obtient les décompositions reprises à la figure II.13. A chaque fois, on a représenté en traits pointillés, l'arête à laquelle est appliquée la formule. Ainsi, on s'aperçoit que le nombre d'arbres couvrants pour le graphe de départ vaut 8.

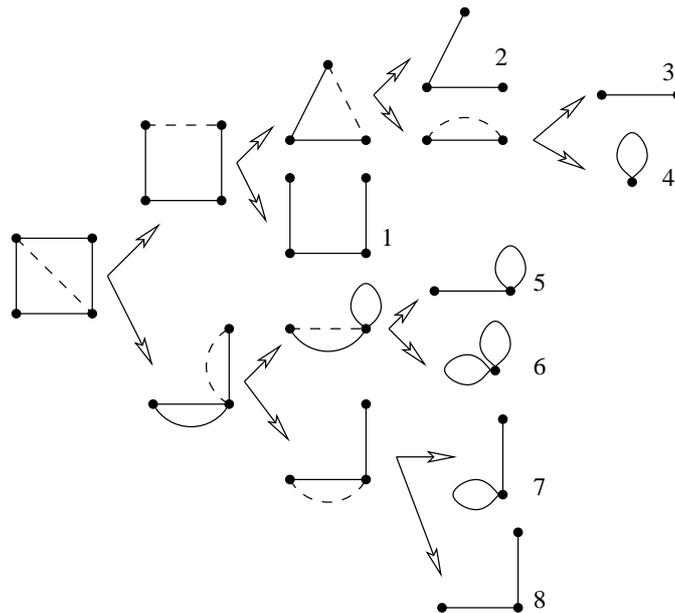


FIGURE II.13. Applications successives de la formule de Cayley.

Cet exemple montre bien que cette méthode est difficile à mettre en pratique pour des graphes de grande taille.

5.2. Une preuve bijective. En combinatoire et dans les problèmes de dénombrement, il est parfois aisé de compter le nombre d'éléments d'un ensemble en montrant que cet ensemble (fini) est en bijection avec un autre ensemble plus simple à énumérer ou à dénombrer. Cette méthode assez élégante est illustrée par le dénombrement des sous-arbres couvrants du graphe complet K_n .

Proposition II.5.6 (Cayley (1897)). *Le nombre de sous-arbres couvrants du graphe complet K_n vaut n^{n-2} . (Le nombre d'arbres à n sommets de labels distincts $\{1, \dots, n\}$ vaut n^{n-2} .)*

Remarque II.5.7. Dans l'énoncé précédent, il faut comprendre que chaque sommet du graphe est pourvu d'un label. A titre d'exemple, K_3 possède 3 sous-arbres couvrants comme indiqué à la figure II.14. En effet, bien

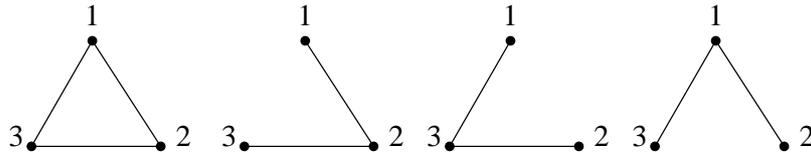


FIGURE II.14. Nombre de sous-arbres couvrants.

qu'il s'agisse de trois arbres isomorphes, on les considère comme des arbres distincts à cause des labels portés par les différents sommets. Ainsi, notre formule de comptage va prendre en compte cette distinction !

La preuve donnée ici est à l'origine dûe à Prüfer. Ainsi, le codage des arbres que nous allons présenter est parfois appelé *codage de Prüfer*.

Démonstration. Numérotons les sommets de K_n de 1 à n . Il est clair que le nombre de sous-arbres couvrant K_n est égal au nombre d'arbres distincts que l'on peut construire avec des sommets numérotés de 1 à n . Il nous suffit donc de compter ces arbres.

Nous présentons un encodage (i.e., une bijection) d'un arbre A par une suite s de $n - 2$ symboles appartenant à $\{1, \dots, n\}$. Le nombre de telles suites est bien sûr égal à n^{n-2} .

Pour obtenir le j -ième élément de s , on supprime de A le plus petit sommet a_j de degré 1 et le j -ième élément de s est donné par le sommet adjacent à a_j . Considérons par exemple, l'arbre donné à la figure II.15. Le premier sommet supprimé est 2, son voisin est 1. Ensuite on supprime 5 de voisin 4, puis 4 de voisin 3, puis 3 de voisin 1, puis 1 de voisin 6 et enfin 6 de voisin 7. Cela nous donne la suite $(1, 4, 3, 1, 6, 6)$. De par l'encodage, il est clair que les sommets apparaissant dans s sont exactement les sommets de degré au moins 2.

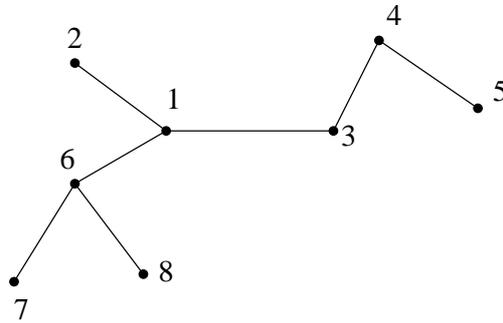


FIGURE II.15. Un arbre encodé par (1, 4, 3, 1, 6, 6).

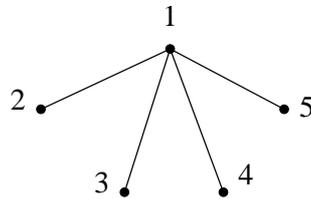


FIGURE II.16. Un arbre encodé par (1, 1, 1).

Réciproquement, si on se donne une suite $s = (s_1, \dots, s_{n-2})$ de $n - 2$ éléments de $V = \{1, \dots, n\}$, il lui correspond un arbre A à n sommets dont s est l'encodage. Les sommets de $V \setminus \{s_1, \dots, s_{n-2}\}$ sont exactement les sommets de degré 1. Soit i_1 le plus petit élément de $V \setminus \{s_1, \dots, s_{n-2}\}$. L'arête $\{i_1, s_1\}$ appartient à A . On recommence: soit i_2 le plus petit élément de $(V \setminus \{i_1\}) \setminus \{s_2, \dots, s_{n-2}\}$. L'arête $\{i_2, s_2\}$ appartient à A . A la fin de la procédure, V contient deux sommets et on ajoute A l'arête formée par ceux-ci.

■

Ce résultat peut aussi s'obtenir comme corollaire d'un résultat de dénombrement plus général : compter le nombre d'arbres dont les sommets ont des labels distincts et les degrés sont fixés.

Théorème II.5.8. *Le nombre d'arbres ayant n sommets de label respectif x_1, \dots, x_n et dont les degrés sont donnés par $\deg(x_1) = d_1, \dots, \deg(x_n) = d_n$ vaut le coefficient multinomial*

$$T_{n,d_1,\dots,d_n} := \binom{n-2}{d_1-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)! \cdots (d_n-1)!}$$

à condition qu'un arbre ayant de telles spécificités existe²⁵.

Exemple II.5.9. Pour les arbres donnés à la figure II.14 de la remarque II.5.7, il y a un seul arbre à 3 sommets de label 1, 2, 3 et de degré respectif 2, 1, 1 (il y en a aussi un pour les degrés 1, 2, 1 et 1, 1, 2). La formule donne

²⁵Par exemple, il n'existe aucun arbre ayant 5 sommets tous de degré 2.

bien

$$\binom{3-2}{2-1, 1-1, 1-1} = \frac{1!}{1!0!0!} = 1.$$

Démonstration. Commençons par quelques remarques préliminaires. Au vu de la remarque I.2.3,

$$d_1 + \cdots + d_n = 2\#E = 2(n-1).$$

Donc, $T \neq 0$ (i.e., il y a au moins un arbre correspondant aux données de l'énoncé) si et seulement si

$$\sum_{i=1}^n (d_i - 1) = 2(n-1) - n = n-2.$$

De plus, pour tout i , $1 \leq d_i \leq n-1$. Il existe une permutation ν des sommets qui est telle que $d_{\nu_1} \geq \cdots \geq d_{\nu_n}$. Pour simplifier l'écriture et ne rien changer au raisonnement, nous allons supposer que $d_1 \geq d_2 \geq \cdots \geq d_n$. Au vu de l'égalité précédente, on en conclut que $d_n = 1$. En effet, si $d_n \geq 2$, alors $\sum_{i=1}^n (d_i - 1) \geq 2n - n \geq n$, ce qui est impossible au vu de la formule ci-dessus.

On a

$$T_{n,d_1,\dots,d_n} = \sum_{i:d_i \geq 2} T_{n-1,d_1,\dots,d_{i-1},d_i-1,d_{i+1},\dots,d_{n-1}}.$$

En effet, puisque x_n est de degré 1, on considère la famille C_i des arbres ayant x_1, \dots, x_n pour sommets de degré respectif d_1, \dots, d_n et dont le sommet x_i est de degré au moins 2 et pour lequel $\{x_i, x_n\}$ est une arête de l'arbre. (Une illustration est donnée à la figure II.17.) Pour un i fixé, il suffit donc pour

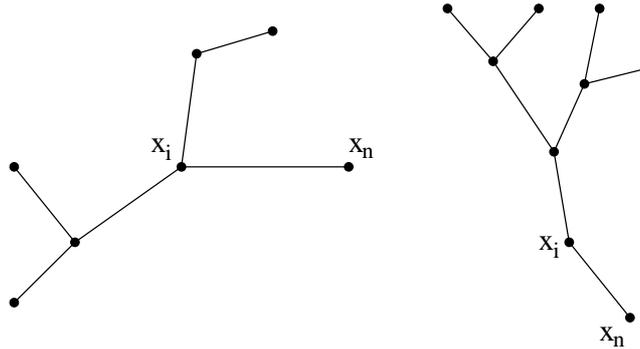


FIGURE II.17. Deux arbres de C_i pour lesquels $d_i = 2$ et $d_i = 3$.

énumérer les arbres de C_i d'énumérer les arbres ayant $n-1$ sommets de label $x_1, \dots, x_{i-1}, \mathbf{x}_i, x_{i+1}, \dots, x_{n-1}$ et de degré $d_1, \dots, d_{i-1}, \mathbf{d}_i - 1, d_{i+1}, \dots, d_{n-1}$.

Nous pouvons à présent démontrer le résultat annoncé en procédant par récurrence. Le cas de base $n = 2$ est immédiat. Supposons donc $n \geq 3$ et la

formule vérifiée pour $n - 1$. Au vu de l'argumentation développée ci-dessus, il vient

$$\begin{aligned}
 T_{n,d_1,\dots,d_n} &= \sum_{i:d_i \geq 2} T_{n-1,d_1,\dots,d_{i-1},d_i-1,d_{i+1},\dots,d_{n-1}} \\
 &= \sum_{i:d_i \geq 2} \binom{n-3}{d_1-1, \dots, d_{i-1}-1, d_i-2, d_{i+1}-1, \dots, d_{n-1}-1} \\
 &= \binom{n-2}{d_1-1, d_2-1, \dots, d_{n-1}-1} \\
 &= \binom{n-2}{d_1-1, d_2-1, \dots, d_{n-1}-1, d_n-1}
 \end{aligned}$$

où, à la deuxième ligne, on utilise l'hypothèse de récurrence et pour obtenir l'avant-dernière ligne, nous avons utilisé le résultat suivant (nous sommes dans les conditions d'application, $d_1 - 1 + \dots + d_n - 1 = n - 2$ car $d_n = 1$.)

■

Lemme II.5.10. *Pour tous $n, n_1, \dots, n_p \in \mathbb{N}$ tels que $n_1 + \dots + n_p = n$, on a*

$$\binom{n}{n_1, \dots, n_p} = \sum_{i:n_i \geq 1} \binom{n-1}{n_1, \dots, n_{i-1}, n_i-1, n_{i+1}, \dots, n_p}.$$

Démonstration. En effet, il est clair que

$$(a_1 + \dots + a_p)^n = (a_1 + \dots + a_p) (a_1 + \dots + a_p)^{n-1}.$$

En se rappelant le théorème multinomial²⁶, le terme général du membre de gauche est

$$\binom{n}{n_1, \dots, n_p} a_1^{n_1} \dots a_p^{n_p}$$

et celui du membre de droite est

$$\sum_{i:n_i \geq 1} a_i \binom{n-1}{n_1, \dots, n_{i-1}, n_i-1, n_{i+1}, \dots, n_p} a_1^{n_1} \dots a_i^{n_i-1} \dots a_p^{n_p}.$$

La conclusion s'ensuit directement.

■

Remarque II.5.11. Ainsi, la proposition II.5.6 est un corollaire immédiat du théorème II.5.8. En effet, dans la proposition II.5.6, on compte juste

²⁶Généralisation bien connue du binôme de Newton

$$(a_1 + \dots + a_p)^n = \sum_{\substack{n_1, \dots, n_p \in \mathbb{N} \\ n_1 + \dots + n_p = n}} \binom{n}{n_1, \dots, n_p} a_1^{n_1} \dots a_p^{n_p}.$$

le nombre d'arbres ayant n sommets (quel que soit leur degré). Ainsi, cela revient à calculer la somme suivante

$$\begin{aligned} \sum_{\substack{d_1, \dots, d_n \geq 1 \\ d_1 - 1 + \dots + d_n - 1 = n - 2}} \binom{n - 2}{d_1 - 1, \dots, d_n - 1} &= \sum_{\substack{i_1, \dots, i_n \geq 0 \\ i_1 + \dots + i_n = n - 2}} \binom{n - 2}{i_1, \dots, i_n} \\ &= \underbrace{(1 + \dots + 1)}_{n \times}^{n - 2} = n^{n - 2}. \end{aligned}$$

Corollaire II.5.12 (Clarke (1958)). *Le nombre d'arbres ayant n sommets de label respectif x_1, \dots, x_n et dont les degrés sont donnés par $\deg(x_1) = d_1 = k, \dots, \deg(x_n) = d_n$ vaut*

$$C_{n-2}^{k-1} (n-1)^{n-k-1}.$$

Démonstration. En effet, vu le théorème II.5.8, il suffit de calculer

$$\sum_{\substack{d_2, \dots, d_n \geq 1 \\ k-1+d_2-1+\dots+d_n-1=n-2}} \binom{n-2}{k-1, d_2-1, \dots, d_n-1}.$$

Cette somme se réécrit

$$\frac{(n-2)!}{(k-1)!(n-k-1)!} \sum_{\substack{d_2, \dots, d_n \geq 1 \\ d_2-1+\dots+d_n-1=n-k-1}} \binom{n-k-1}{d_2-1, \dots, d_n-1}.$$

D'où la conclusion en procédant comme dans la remarque précédente. ■

5.3. Nombre de sous-arbres couvrants. Nous allons tout d'abord étendre le problème posé initialement et compter les sous-arbres couvrants, pointés et orientés depuis la racine dans un **multi-graphe orienté** $G = (V, E)$ où $V = \{v_1, \dots, v_n\}$. Nous supposons de plus que ce multi-graphe est **sans boucle**. Ceci ne constitue pas une véritable restriction par rapport au problème envisagé ici²⁷.

Un arbre pointé est *orienté depuis la racine* si les arcs constituant celui-ci sont tous orientés des sommets de niveau i vers les sommets de niveau $i+1$ (pour i allant de 0 à la hauteur de l'arbre). Bien évidemment, le graphe non orienté sous-jacent est lui-même un arbre.

Exemple II.5.13. A la figure II.18, on a représenté un graphe orienté et un sous-arbre couvrant orienté depuis la racine 1.

Définition II.5.14. Soit $G = (V, E)$ un multi-graphe orienté sans boucle dont les sommets sont ordonnés par $V = \{v_1, \dots, v_n\}$. La *matrice $D(G)$ de demi-degré entrant*²⁸ est définie par

$$[D(G)]_{j,j} = d^-(v_j) \quad \text{et} \quad [D(G)]_{i,j} = -(\#\omega^+(v_i) \cap \omega^-(v_j)), \quad \text{si } i \neq j.$$

²⁷Dans un problème de couverture, les boucles ne servent à rien.

²⁸Elle est aussi parfois appelée *matrice de Kirchoff*.

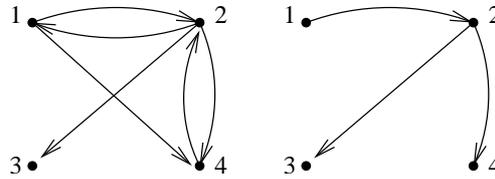


FIGURE II.18. Un sous-arbre couvrant pointé et orienté.

Autrement dit, $[D(G)]_{i,j}$ est l'opposé du nombre d'arcs joignant v_i à v_j , si $i \neq j$. En conséquence de cette définition, il est clair que la somme des éléments de toute colonne de $D(G)$ est nulle.

Exemple II.5.15. Pour le graphe de la figure II.18, on trouve

$$D(G) = \begin{pmatrix} 1 & -1 & 0 & -1 \\ -1 & 2 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 2 \end{pmatrix}.$$

Remarque II.5.16. La matrice de demi-degré entrant associée à un sous-arbre couvrant pointé et orienté possède, à l'exception de la colonne correspondant à la racine, exactement un "−1" dans chaque colonne et une diagonale formée de 1 (on aboutit exactement une fois dans chaque sommet). La colonne correspondant à la racine est nulle puisqu'aucun arc n'aboutit à la racine. Ainsi, pour le sous-arbre de la figure II.18, on a la matrice

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

Nous allons introduire des sous-graphes particuliers de G . Soient $i \in \{1, \dots, n\}$ et $G^{(i)} = (V, E \setminus \omega^-(v_i))$, i.e., le sous-graphe obtenu en supprimant les arcs de G entrant dans v_i . Un tel sous-graphe revient à sélectionner une racine pour un arbre couvrant potentiel (en effet, aucun arc n'aboutit à la racine).

Exemple II.5.17. Pour le graphe de la figure II.18, on trouve par exemple

$$D(G^{(3)}) = \begin{pmatrix} 1 & -1 & 0 & -1 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 \end{pmatrix}.$$

Si $[D(G^{(i)})]_{k,k} = r_k^{(i)} \geq 2$, alors on peut écrire, de manière unique, la k -ième colonne de $D(G^{(i)})$ comme une somme de $r_k^{(i)}$ vecteurs-colonnes

$$C_{k,1}^{(i)}, \dots, C_{k,r_k^{(i)}}^{(i)}$$

dont la k -ième composante vaut 1 et dont toutes les autres composantes sont nulles, exceptée l'une d'entre elles valant -1 . Une telle décomposition revient à sélectionner pour chaque sommet un seul arc entrant, toujours dans l'optique de construire un potentiel arbre couvrant.

Exemple II.5.18. En continuant notre exemple,

$$\begin{pmatrix} -1 \\ 2 \\ 0 \\ -1 \end{pmatrix} = \underbrace{\begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{C_{2,1}^{(3)}} + \underbrace{\begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}}_{C_{2,2}^{(3)}} \quad \text{et} \quad \begin{pmatrix} -1 \\ -1 \\ 0 \\ 2 \end{pmatrix} = \underbrace{\begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{C_{4,1}^{(3)}} + \underbrace{\begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix}}_{C_{4,2}^{(3)}}.$$

Soient $k_1^{(i)}, \dots, k_p^{(i)}$ les indices des colonnes de $D(G^{(i)})$ pour lesquelles $[D(G^{(i)})]_{k,k} = r_k^{(i)} \geq 2$. Si on choisit pour chacune de ces colonnes un des vecteurs $C_{k_j,n}^{(i)}$, pour $j = 1, \dots, p$ et $n = 1, \dots, r_{k_j}^{(i)}$. Le nombre total de tels choix possibles vaut donc $m^{(i)} = r_{k_1}^{(i)} \cdots r_{k_p}^{(i)} \geq 2^p$ et à chacun de ces choix, il correspond une matrice

$$D^{(i)}_1, \dots, D^{(i)}_{m^{(i)}}.$$

En particulier, il est clair, par linéarité du déterminant par rapport aux colonnes, que

$$(8) \quad \det(D(G^{(i)})) = \sum_{j=1}^{m^{(i)}} \det(D^{(i)}_j).$$

Remarquons aussi que pour un multi-graphe, il se peut que plusieurs matrices $D^{(i)}_j$ soient identiques. Cela n'est nullement gênant et il faut en tenir compte pour compter exactement le nombre de sous-arbres couvrants.

Exemple II.5.19. Si on poursuit l'exemple, $[D(G^{(3)})]_{2,2} = [D(G^{(3)})]_{4,4} = 2$ et on a donc $m^{(3)} = 4$ matrices

$$D_1^{(3)} = \begin{pmatrix} 1 & -1 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D_2^{(3)} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

et

$$D_3^{(3)} = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, \quad D_4^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$$

telles que $\det(D(G^{(3)})) = \sum_{i=1}^4 \det(D_i^{(3)})$.

A ce stade, à chaque matrice $D(G^{(k)})$ correspond une sélection d'une racine et une sélection d'un arc entrant pour chaque sommet distinct de la racine. Pour un sélection convenable, il s'agira d'un arbre couvrant.

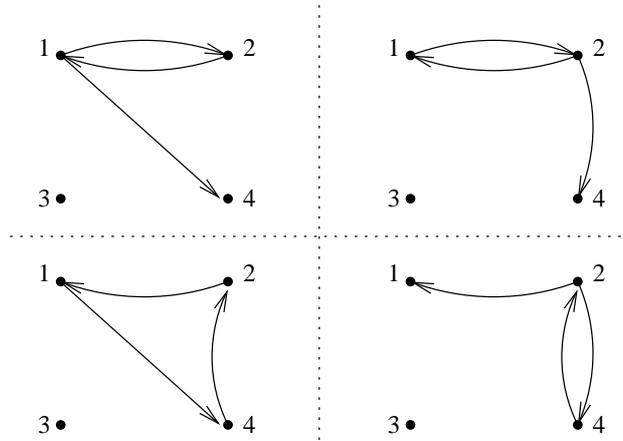


FIGURE II.19. Graphes correspondant aux matrices $D_1^{(3)}, D_2^{(3)}, D_3^{(3)}$ et $D_4^{(3)}$.

Exemple II.5.20. Aux matrices $D_1^{(3)}, D_2^{(3)}, D_3^{(3)}$ et $D_4^{(3)}$ de l'exemple précédent, correspondent les graphes de la figure II.19.

Remarque II.5.21. Supposons qu'un sous-arbre couvrant pointé en v_i et orienté existe. Alors, au vu de la remarque II.5.16, la matrice associée à cet arbre est exactement égale à une des matrices $D^{(i)}_1, \dots, D^{(i)}_{m(i)}$. Pour compter le nombre de sous-arbres couvrants pointés en v_i , il suffit donc de pouvoir distinguer les matrices $D^{(i)}_j$ qui correspondent à un sous-arbre couvrant de celles qui n'y correspondent pas.

Toute matrice $D^{(i)}_j$ est une matrice de demi-degré entrant pour un certain sous-graphe de G , noté $G^{(i)}_j$, dont chaque sommet a un demi-degré entrant valant au plus 1.

Théorème II.5.22. Soit $G = (V, E)$ un graphe orienté dont le demi-degré entrant de chaque sommet vaut au plus 1. Le mineur $M_{t,t}(G)$ de la matrice $D(G)$ obtenu en supprimant la ligne et la colonne correspondant à v_t vaut

$$\begin{cases} 1, & \text{si } G \text{ contient un sous-arbre couvrant pointé en } v_t \text{ et orienté;} \\ 0, & \text{sinon.} \end{cases}$$

Démonstration. Supposons que G contienne un sous-arbre couvrant A orienté et pointé en v_t . Puisque par hypothèse, le demi-degré entrant de chaque sommet vaut au plus 1, alors $G = A$ ou bien $G = A + e$ où e est une arête entrant dans v_t . On peut renuméroter les sommets de A (donc de G) par un parcours en largeur de l'arbre. Ainsi, l'arête v_t devient v_1 et $[D(G)]_{1,1} \leq 1$. Pour $i \geq 2$, $[D(G)]_{i,i} = 1$ et

$$\text{si } i > j \geq 2, \text{ alors } [D(G)]_{i,j} = 0$$

(c'est une conséquence de la renumérotation des sommets). Si $G = A + e$, alors $D(G)$ contient un "−1" dans la première colonne, mais cela n'a guère d'importance pour la suite. On conclut que la matrice $D(G)$ privée de

sa première ligne et de sa première colonne est une matrice triangulaire supérieure de déterminant 1.

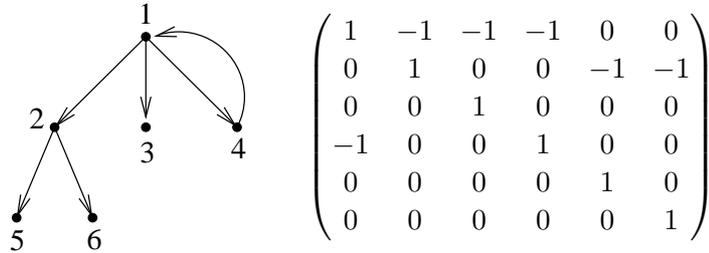


FIGURE II.20. Un arbre pointé parcouru en largeur (avec un arc supplémentaire).

Supposons à présent que G ne contienne pas de sous-arbre couvrant orienté et pointé en v_t . Si pour $j \neq t$, $d^-(v_j) = 0$, alors la j -ième colonne de $D(G)$ est nulle et on en conclut que $M_{t,t}(G) = 0$. Nous pouvons donc supposer que pour tout $j \neq t$, $d^-(v_j) = 1$.

Soit $j_1 \neq t$. Puisque $d^-(v_{j_1}) = 1$, il existe un indice j_2 et un arc joignant v_{j_2} à v_{j_1} . En continuant de la sorte, on obtient des sommets tous distincts v_{j_k}, \dots, v_{j_1} tels que

$$v_{j_k} \longrightarrow v_{j_{k-1}} \longrightarrow \dots \longrightarrow v_{j_2} \longrightarrow v_{j_1}.$$

Si en effectuant cette construction, on rencontre le sommet v_t , on considère alors un autre sommet que v_{j_1} pour initialiser la construction. Nous affirmons qu'il existe au moins un sommet de G pour lequel la construction ne rencontrera pas v_t . En effet, si ce n'était pas le cas, on aurait identifié à chaque fois un sous-arbre pointé en v_t et on obtiendrait dès lors un sous-arbre couvrant pointé en v_t , ce qui n'est pas possible dans la situation envisagée ici. Dans la construction de v_{j_k}, \dots, v_{j_1} , puisque le graphe contient un nombre fini de sommets, on finit par identifier un cycle. Les colonnes de $D(G)$ correspondant aux sommets de ce cycle sont linéairement dépendantes car leur somme fait zéro. De là, on en conclut que $M_{t,t}(G) = 0$.

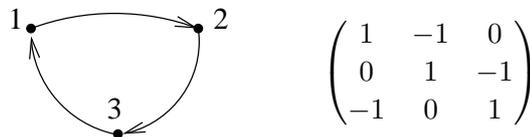


FIGURE II.21. Un cycle et des colonnes dont la somme est nulle.

■

Les graphes $G^{(i)}_j$ satisfont l'hypothèse du théorème II.5.22. Au vu de ce théorème et de la remarque II.5.21, le nombre de sous-arbres couvrant $G^{(i)}$ pointés en v_i vaut

$$\sum_{j=1}^{m^{(i)}} M_{i,i}(G^{(i)}_j) = M_{i,i}(G^{(i)}).$$

On a utilisé un raisonnement analogue à (8) appliqué ici au calcul d'un mineur. Ce nombre est aussi, bien évidemment, égal au nombre de sous-arbres couvrant G tout entier et pointés en v_i . Les matrices $D(G^{(i)})$ et $D(G)$ étant égales à l'exception de la i -ème colonne, on obtient le résultat suivant.

Théorème II.5.23 (Bott-Mayberry (1954)). *Soit G un multi-graphe orienté sans boucle. Le nombre de sous-arbres couvrant G pointés au sommet v_i et orientés est égal au mineur $M_{i,i}(G)$ de la matrice de demi-degré entrant de G .*

Nous pouvons à présent reconsidérer notre problème initial concernant des graphes non orientés. A un graphe non orienté $G = (V, E)$ correspond un graphe orienté $G' = (V, E')$ où chaque arête $\{u, v\}$ de G donne lieu aux arcs (u, v) et (v, u) dans G' . Il est clair qu'à tout arbre couvrant dans G , il correspond dans G' exactement un arbre couvrant pointé en a et orienté depuis a et ce, pour tout sommet a de G' . La réciproque est également vraie. A tout arbre couvrant G' (pointé en un quelconque sommet et orienté), il correspond un arbre couvrant G .

Corollaire II.5.24. *Le nombre de sous-arbres couvrant un multi-graphe $G = (V, E)$ non orienté sans boucle vaut $M_{i,i}(G')$ quel que soit i , où G' est le graphe symétrique orienté déduit de G .*

Exemple II.5.25. Sur la figure II.22, on a représenté un graphe non orienté G , le graphe orienté symétrique G' correspondant à G , ainsi qu'un arbre couvrant G et les arbres couvrants correspondants dans G' avec comme racines respectives les différents sommets de G' .

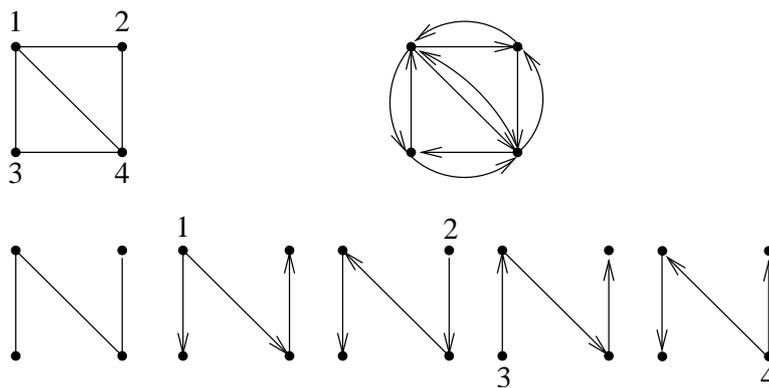


FIGURE II.22. Nombre de sous-arbres couvrants pour G et G' .

$$D(G') = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}.$$

On vérifie que tous les mineurs principaux valent 8. On a ainsi 8 arbres couvrants distincts (comme le montre la figure II.23). Insistons une fois

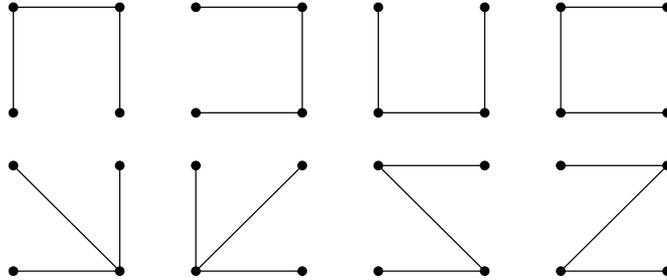


FIGURE II.23. Les sous-arbres couvrants du graphe G .

encore sur le fait que nous avons compté l'ensemble des sous-arbres d'un graphe dont les sommets ont été numérotés. Ainsi, parmi les 8 arbres trouvés ci-dessus, seulement 3 sont 2 à 2 non isomorphes.

6. Arbres couvrants de poids minimal

Nous allons à présent considérer le problème de trouver non seulement un sous-arbre couvrant mais un sous-arbre couvrant de poids minimal. Rappelons que l'on se restreint au cas de graphes (non orientés) simples. (En effet, si on disposait de plusieurs arêtes joignant deux sommets, il suffirait de conserver l'arête de plus petit poids.) L'algorithme de Prim répond à cette question.

Soit $G = (V, E)$ un graphe connexe non orienté dont les arêtes sont pondérées par une fonction $p : E \rightarrow \mathbb{R}^+$. L'idée de cet algorithme est de construire le sous-arbre de proche en proche. Si on dispose déjà d'un sous-graphe $G' = (V', A)$ connexe d'un arbre de poids minimal couvrant G , on lui ajoute une arête de poids minimal parmi les arêtes joignant un sommet de V' à un sommet de $V \setminus V'$. Puisqu'il faut couvrir l'arbre tout entier, on débute la procédure avec un sous-graphe restreint à une quelconque arête de G .

Nous utilisons les mêmes conventions qu'à la remarque I.4.9. Ainsi, $p(\{u, v\}) = +\infty$ si $\{u, v\} \notin E$.

Algorithme II.6.1 (Prim). La donnée de l'algorithme est un graphe connexe non orienté $G = (V, E)$.

Choisir un sommet $v_0 \in V$

$V' := \{v_0\}$, $A := \emptyset$

Pour tout $v \in V \setminus V'$,

$L(v) := p(\{v_0, v\})$

Tant que $V' \neq V$, répéter

Choisir $u \in V \setminus V'$ tel que $L(u)$ soit minimal

et e l'arête réalisant ce poids

$V' := V' \cup \{u\}$

$A := A \cup \{e\}$
 Pour tout $v \in V \setminus V'$,
 si $p(\{u, v\}) < L(v)$, alors $L(v) := p(\{u, v\})$

Exemple II.6.2. Appliquons l'algorithme au graphe de la figure II.24.

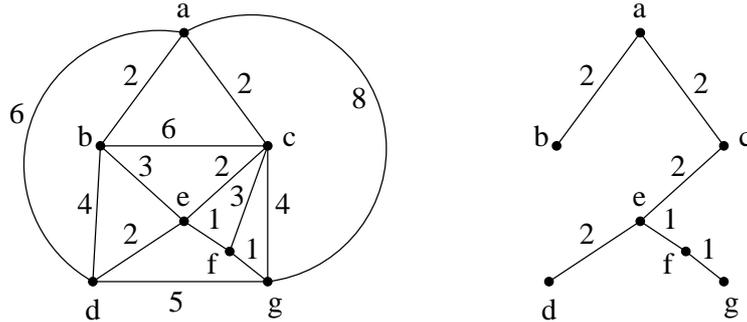


FIGURE II.24. Application de l'algorithme de Prim.

V'	A	$L(a)$	$L(b)$	$L(c)$	$L(d)$	$L(e)$	$L(f)$	$L(g)$
$\{a\}$	\emptyset	–	2	2	6	$+\infty$	$+\infty$	8
$\{a, b\}$	$\{\{a, b\}\}$	–	–	2	4	3	$+\infty$	8
$\{a, b, c\}$	$\{\dots, \{a, c\}\}$	–	–	–	4	2	3	4
$\{a, b, c, e\}$	$\{\dots, \{c, e\}\}$	–	–	–	2	–	1	4
$\{a, b, c, e, f\}$	$\{\dots, \{e, f\}\}$	–	–	–	2	–	–	1
$\{a, b, c, e, f, g\}$	$\{\dots, \{f, g\}\}$	–	–	–	2	–	–	–
$\{a, b, c, e, f, g, d\}$	$\{\dots, \{d, e\}\}$	–	–	–	–	–	–	–

TABLE II.3. Application de l'algorithme de Prim.

Démonstration. On procède par récurrence sur $\#V'$, le cas de base $\#V' = 1$ étant immédiat. Le choix de l'arête e joignant un sommet de V' à un sommet de $V \setminus V'$ assure que l'on obtienne un arbre. En effet, si $G' = (V', A)$ est un arbre, il ne contient pas de cycle et l'arête e ajoutée ne saurait créer un cycle. Ainsi $G' + e$ est encore un arbre.

Par hypothèse de récurrence, G' est un sous-arbre d'un arbre T couvrant G et de poids minimal. Supposons que e ne soit pas une arête de T . Ainsi, $T + e$ contient un cycle contenant e mais aussi une autre arête e' joignant un sommet de V' à un sommet de $V \setminus V'$. Considérons l'arbre $S = (T + e) - e'$. Il s'agit encore d'un arbre couvrant. De par les choix réalisés dans l'algorithme, son poids

$$p(S) = p(T) + p(e) - p(e')$$

est tel que $p(S) \leq p(T)$ (car $p(e) \leq p(e')$). Or T est de poids minimal. Donc $p(S) = p(T)$ et on en conclut que e appartient bien à un arbre couvrant G et de poids minimal.

■

Remarque II.6.3. L'algorithme s'adapte aisément pour rechercher un sous-arbre couvrant de poids maximal. Il est facile de se convaincre que l'algorithme de Prim a une complexité temporelle en $\mathcal{O}((\#V)^2)$.

L'algorithme de Prim n'est pas le seul répondant à cette question. On rencontre souvent l'algorithme de Kruskal brièvement décrit ci-après.

Algorithme II.6.4 (Kruskal). La donnée de l'algorithme est un graphe connexe non orienté $G = (V, E)$.

Enumérer les arêtes e_1, e_2, \dots de G de telle sorte que

si $p(e_i) > p(e_j)$, alors $i > j$

$A := \emptyset$

Pour i allant de 1 à $\#E$

Si $A \cup \{e_i\}$ est acyclique alors $A := A \cup \{e_i\}$.

CHAPITRE III

Graphes planaires

1. Introduction

Dans ce chapitre, on considère uniquement des graphes non orientés. En effet, l'orientation ne joue aucun rôle par rapport aux questions de planarité. Attirons de plus l'attention du lecteur sur le fait que la plupart des résultats (comme la formule d'Euler) ne sont applicables qu'à des graphes finis.

Définition III.1.1. Un multi-graphe G est *planaire* (ou *planaire topologique*) s'il est possible de le représenter dans le plan affiné euclidien de manière telle que les sommets distincts de G soient des points distincts du plan et les arêtes soient des courbes simples¹. On impose en plus que deux arêtes distinctes ne se rencontrent pas en dehors de leurs extrémités. On parlera alors de *représentation planaire* de G .

On ne considérera pas comme distinctes deux représentations planaires que l'on peut amener à coïncider par "déformation élastique du plan"²

Remarque III.1.2. Un même graphe planaire peut avoir plusieurs représentations planaires distinctes (cf. figure III.4).

Remarque III.1.3. Pour rappel, le théorème de Jordan stipule que pour toute courbe simple fermée Γ , $\mathbb{R}^2 \setminus \Gamma$ possède exactement deux composantes connexes. Cela va donc nous permettre de définir la notion de face de manière rigoureuse. En effet, un circuit simple détermine une courbe simple.

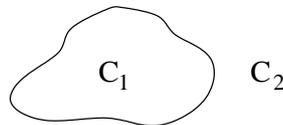


FIGURE III.1. Théorème de Jordan

Définition III.1.4. Soit G un multi-graphe planaire. Une *face* de G est une région du plan délimitée par des arêtes formant un circuit simple. Par conséquent, deux points arbitraires de cette région peuvent être joints par un

¹Aucun point multiple.

²On considère l'ensemble quotient des représentations homéomorphes (au sens topologique du terme), cf. un cours de topologie générale, un homéomorphisme est une bijection continue d'inverse continu.

trait continu ne rencontrant aucun sommet ni arête de G . La *frontière* ∂F d'une face F est l'ensemble des arêtes qui "touchent"³ la face (on omet donc les sommets). Par extension et pour simplifier notre exposé, on s'autorisera à parler des sommets et des arêtes appartenant à une face (si les arêtes appartiennent à la frontière de la face considérée ou si un sommet est une extrémité d'une telle arête). Deux faces F et F' sont *adjacentes* si leur frontière ont au moins une arête commune, i.e., $\partial F \cap \partial F' \neq \emptyset$. En particulier, deux faces ne se touchant que par un sommet ne sont pas considérées comme adjacentes. Un multi-graphe planaire (fini) contient toujours une et une seule *face infinie*, i.e., non bornée. On peut néanmoins toujours parler de la frontière de la face infinie.

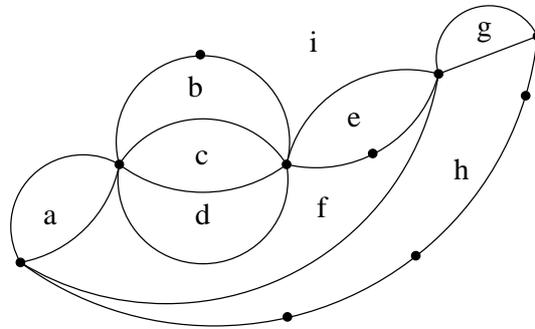


FIGURE III.2. Un graphe planaire et ses faces.

Remarque III.1.5. A titre indicatif, citons le théorème de Steinitz (1922). Rappelons que le *squelette* d'un polyèdre P est le graphe dont les sommets sont les sommets de P et dont les arêtes sont aussi celles de P . Par exemple, le graphe de gauche à la figure I.13 est le squelette d'un cube. Le résultat s'énonce comme suit. *Un graphe est le squelette d'un polyèdre convexe (borné) de \mathbb{R}^3 si et seulement si c'est un graphe planaire au moins 3-connexe (i.e., ne pouvant pas être disconnecté en retirant moins de trois sommets, $\kappa(G) \geq 3$).*

Proposition III.1.6. *Si G est un multi-graphe planaire et Δ une face de G . Il est possible d'obtenir une représentation planaire du graphe G dans le plan affine euclidien de manière telle que Δ soit la face infinie. En particulier, pour tout sommet x de G , on peut obtenir une représentation planaire du graphe G telle que x appartienne à la face infinie.*

Démonstration. Par projection stéréographique (cf. figure IV.3), on se ramène à une représentation du graphe sur la sphère et inversement. Par rotation (cf. figure III.3), on peut toujours amener la face Δ à contenir le pôle de la projection. Il s'agit d'un argument purement géométrique.

³Nous nous autorisons une définition naïve pour ne pas compliquer inutilement l'exposé. Il suffit de parler de la frontière au sens topologique du terme.

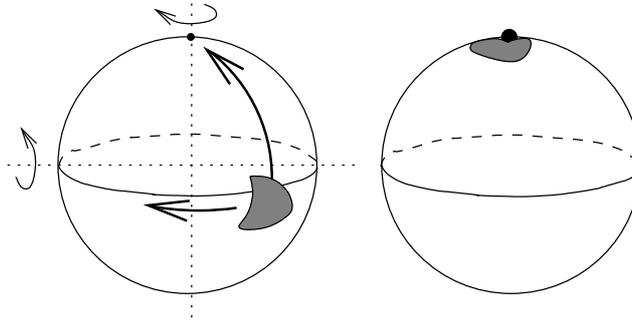


FIGURE III.3. Rotation sur la sphère

■

Exemple III.1.7. A la figure III.4, on a représenté un même graphe et à chaque fois, la face infinie est différente. Dans le premier cas, la frontière de la face infinie est donnée par $\partial F = \{\{1, 4\}, \{4, 3\}, \{3, 1\}\}$ et dans les deux autres cas par $\partial F = \{\{1, 3\}, \{3, 2\}, \{2, 1\}\}$ et $\partial F = \{\{1, 2\}, \{2, 4\}, \{4, 1\}\}$.

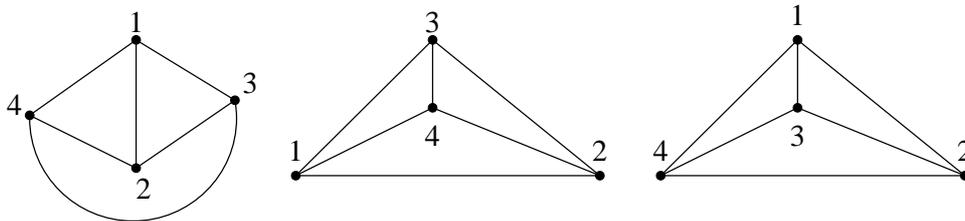


FIGURE III.4. Choix de la face infinie.

2. Formule d'Euler

Le théorème suivant peut parfois être utilisé pour vérifier qu'un graphe donné n'est pas planaire. En effet, c'est principalement grâce à ce résultat que nous montrerons que ni K_5 ni $K_{3,3}$ ne sont planaires.

Théorème III.2.1 (Formule d'Euler⁴). *Dans un multi-graphe planaire connexe (fini) possédant s sommets, a arêtes et f faces, on a*

$$s - a + f = 2.$$

Démonstration. On procède par récurrence sur le nombre f de faces. Si $f = 1$, le graphe possède uniquement une face infinie. Par conséquent, le graphe connexe ne possède aucun cycle, il s'agit d'un arbre. Ainsi, $s = a + 1$ et la formule est vérifiée.

Supposons à présent la formule d'Euler satisfaite pour les valeurs $< f$ et démontrons-la pour $f \geq 2$. Soit e une arête d'un cycle du graphe. Cette

⁴Formule également bien connue en géométrie pour les polyèdres convexes (considéré comme enveloppes convexes d'un nombre fini de points).

arête appartient à la frontière de deux faces A et B . Si on supprime l'arête e , on obtient un nouveau graphe ayant $a - 1$ arêtes, le même nombre s de sommets et $f - 1$ faces (en effet, A et B forment une face de ce nouveau graphe). Par hypothèse de récurrence, on a $s - (a - 1) + f - 1 = 2$ ce qui suffit.

■

Exemple III.2.2. Pour le graphe de la figure III.2, on a $s = 10$, $a = 17$ et $f = 9$.

Le simple argument de comptage utilisé dans la preuve du résultat suivant nous sera utile à plusieurs reprises dans la suite de ce chapitre.

Corollaire III.2.3. Dans un graphe $G = (V, E)$ simple et planaire, il existe un sommet x tel que $\deg(x) \leq 5$.

Démonstration. Quitte à considérer séparément chaque composante connexe de G , nous allons supposer G connexe. On peut tout d'abord éliminer les arêtes ne délimitant pas de face (celles-ci ont une extrémité de degré 1 et le résultat est alors immédiat, cf. figure III.5). Cela ne fait que renforcer encore le résultat.

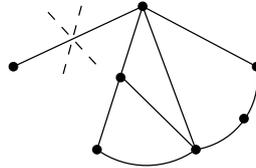


FIGURE III.5. Suppression d'arêtes.

Puisque G est simple, la frontière de toute face contient au moins 3 arêtes. Donc en comptant le nombre de faces, on compte à chaque fois au moins 3 arêtes, ainsi, $3f \leq a$. On peut même dire plus car, dans ce comptage, chaque arête est comptée deux fois puisqu'elle apparaît dans la frontière de deux faces. On en conclut donc que

$$(9) \quad \boxed{f \leq \frac{2a}{3}}$$

Procédons par l'absurde et supposons que pour tout sommet x de G , $\deg(x) \geq 6$. Dans ce cas,

$$s \leq \frac{2a}{6}.$$

Si on applique la formule d'Euler,

$$2 = s - a + f \leq \frac{a}{3} - a + \frac{2a}{3} = 0$$

et on obtient une absurdité.

■

3. Graphes homéomorphes

Nous définissons ici la notion de graphes homéomorphes⁵ nécessaire à l'énoncé du théorème de Kuratowski.

Définition III.3.1. La *subdivision* d'une arête $e = \{a, b\}$ d'un graphe $G = (V, E)$ consiste à remplacer cette arête par deux nouvelles arêtes $e_1 = \{a, u\}$ et $e_2 = \{u, b\}$ où u est un nouveau sommet n'appartenant pas à V . Le



FIGURE III.6. Subdivision d'une arête.

graphe obtenu est alors $G' = (V \cup \{u\}, (E \cup \{e_1, e_2\}) \setminus \{e\})$.

Définition III.3.2. Deux graphes sont *homéomorphes* s'ils peuvent être obtenus par une suite finie (voir vide) de subdivisions à partir d'un même graphe. En particulier, si un graphe G' résulte de subdivisions d'arêtes de G , alors G et G' sont homéomorphes.

Exemple III.3.3. La figure III.7 reprend deux graphes homéomorphes construits à partir d'un même graphe.

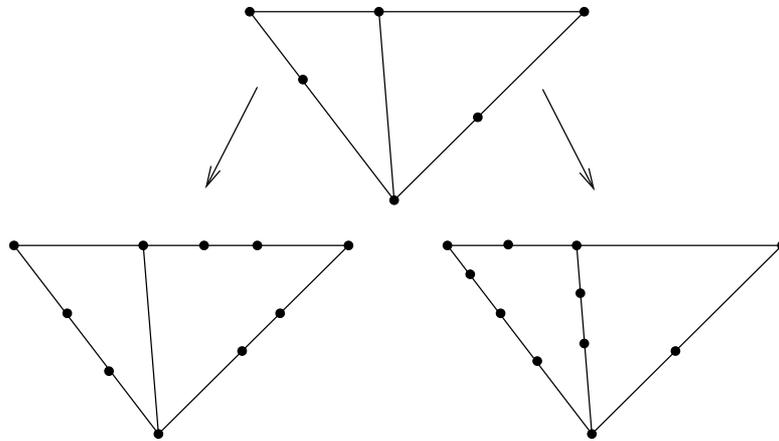


FIGURE III.7. Graphes homéomorphes.

Remarque III.3.4. On se convainc assez facilement que la relation “être homéomorphe” est une relation d'équivalence sur l'ensemble des multi-graphes finis (non orientés). Dans une classe d'équivalence, on peut considérer le graphe ayant un nombre minimum d'arêtes (ou de sommets).

⁵Il est dommage d'utiliser un tel terme car la notion que nous allons définir ici est différente de la notion d'homéomorphisme rencontrée en topologie.

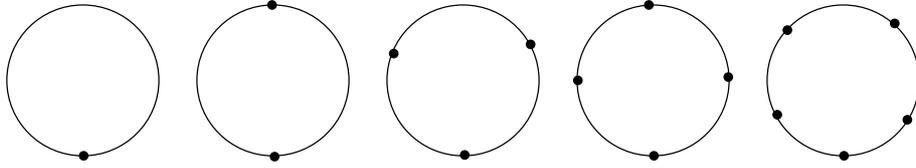


FIGURE III.8. Graphes homéomorphes.

4. Théorème de Kuratowski

Les graphes K_5 et $K_{3,3}$ sont l'archétype même des graphes non planaires.

Lemme III.4.1. *Le graphe K_5 n'est pas planaire.*

Démonstration. On utilise les notations de la formule d'Euler. D'une manière générale, dans un graphe simple et planaire, de la relation $3f \leq 2a$ démontrée en (9) et de la formule d'Euler ($3a - 3f = 3s - 6$), on tire que

$$a \leq 3s - 6.$$

Or, K_5 est un graphe simple qui possède 5 sommets et 10 arêtes et $10 \not\leq 3 \cdot 5 - 6$. On en conclut que K_5 ne peut être planaire. ■

Lemme III.4.2. *Le graphe $K_{3,3}$ n'est pas planaire.*

Démonstration. On utilise une fois encore le même raisonnement que celui ayant permis d'obtenir la relation (9), mais cette fois dans un graphe simple, planaire et biparti. Dès lors, chaque face a une frontière déterminée par au moins 4 arêtes et on en tire que $4f \leq 2a$, i.e., $2f \leq a$. De la formule d'Euler, $2a - 2f = 2s - 4$, on en tire que

$$a \leq 2s - 4.$$

Or, $K_{3,3}$ est un graphe biparti simple qui possède 6 sommets et 9 arêtes. Il ne peut donc pas être planaire car $9 \not\leq 2 \cdot 6 - 4$. ■

Théorème III.4.3. *Un multi-graphe (non orienté) est planaire si et seulement si il ne contient pas de sous-graphe homéomorphe à K_5 ou à $K_{3,3}$.*

La preuve de ce résultat résulte des quatre lemmes suivants (et principalement des lemmes III.4.4 et III.4.7). La présentation adoptée ici est due originellement à Thomassen (1981). Nous dirons qu'un graphe possède la propriété **(K)** s'il contient un sous-graphe homéomorphe à K_5 ou à $K_{3,3}$. En fait, de nombreuses recherches actuelles et hautement non triviales essaient de déterminer des classes de graphes (comme les graphes planaires) qui seraient caractérisées par l'exclusion de certains graphes précis. Le théorème de Kuratowski est le premier exemple d'un tel *théorème d'exclusion*.

Lemme III.4.4. Soit $G = (V, E)$ un graphe non planaire ne satisfaisant pas (\mathbf{K}) et pour lequel $\#E$ est minimal pour ces propriétés. Alors G est 3-connexe.

Pour rappel, l'opération de contraction a été donnée à la définition II.5.3.

Lemme III.4.5. Soit $G = (V, E)$ un graphe 3-connexe ayant au moins 5 sommets. Il existe une arête $e \in E$ telle que la contraction $G \cdot e$ soit encore 3-connexe.

Lemme III.4.6. Soit $G = (V, E)$ un graphe. Si pour une arête $e \in E$, $G \cdot e$ satisfait (\mathbf{K}) , alors G aussi.

Lemme III.4.7. Tout graphe 3-connexe et qui ne satisfait pas (\mathbf{K}) est planaire.

Démonstration. (Théorème de Kuratowski) Au vu des lemmes III.4.1 et III.4.2, il est clair que si G satisfait (\mathbf{K}) , alors G n'est pas planaire.

Il suffit donc de vérifier que si G n'est pas planaire, alors G satisfait (\mathbf{K}) . Procédons par l'absurde et supposons que G est non planaire et satisfait (\mathbf{K}) . Quitte à remplacer G par un graphe qui lui est homéomorphe (cela ne change en rien la (non-)planarité), on peut supposer que G contient un nombre minimal d'arêtes. Ainsi, par le lemme III.4.4, G est 3-connexe. Du lemme III.4.7, on tire que G est planaire. Ceci achève la preuve. ■

Démonstration. (Lemme III.4.4) Montrons d'abord que G est 2-connexe en procédant par l'absurde et en supposant que G possède un point d'articulation v . Soient A_1, \dots, A_k les composantes connexes de $G - v$, $k \geq 2$. Pour tout i , le sous-graphe G_i de G induit par les sommets de A_i et v possède un nombre d'arêtes strictement inférieur au nombre d'arêtes de G . Par hypothèse, G est un graphe non planaire ne satisfaisant pas (\mathbf{K}) et ayant un nombre minimal d'arêtes pour ces propriétés. On en conclut que G_i est planaire⁶ et par la proposition III.1.6, on peut même supposer que v se trouve sur la frontière extérieure de G_i . De là, on en conclut que G est

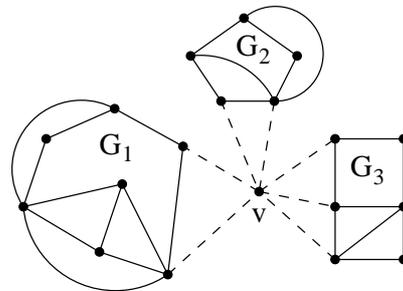


FIGURE III.9. Si G n'était pas 2-connexe.

⁶En effet, si G ne satisfait pas (\mathbf{K}) , il en est de même pour ses sous-graphes.

planaire, une contradiction.

Supposons à présent que G possède un ensemble d'articulation formé de deux sommets: $S = \{u, v\}$. (On ne peut pas faire moins car G est au moins 2-connexe, vu la première partie de la preuve). Soient $G_1 = (E_1, V_1)$ et $G_2 = (E_2, V_2)$ deux sous-graphes de G tels que $E = E_1 \cup E_2$, $V_1 \cap V_2 = S$ et contenant chacun une composante connexe de $G - S$. Il existe dans G_1 (resp. dans G_2) un chemin joignant u et v (car u comme v sont adjacents à un sommet de chaque composante connexe de $G - S$).

On utilise le même raisonnement que précédemment. Soit H_i le sous-graphe de G induit par les sommets de G_i et de S , $i = 1, 2$, auquel on ajoute éventuellement l'arête $\{u, v\}$ (cette arête peut ou non appartenir à E). Si H_1 et H_2 étaient tous les deux planaires, on pourrait en conclure que G lui-même est planaire en plaçant l'arête $\{u, v\}$ à sur la frontière extérieure de H_1 et H_2 (ce qui est possible vu la proposition III.1.6).

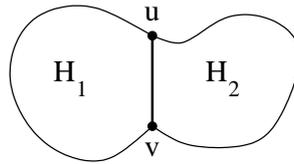


FIGURE III.10. H_1 , H_2 et l'arête $\{u, v\}$.

Ainsi, supposons que H_1 est non planaire. Puisqu'il contient moins d'arêtes que G , alors H_1 satisfait **(K)**. Cependant, il nous faut encore retirer l'arête $\{u, v\}$ si cette dernière n'appartient pas à G . Néanmoins, G va lui aussi satisfaire **(K)** car on dispose d'un chemin dans G_2 joignant u et v . On peut donc substituer ce chemin à $\{u, v\}$ et assurer que G satisfait **(K)** sans utiliser l'arête $\{u, v\}$. On a donc obtenu une contradiction et G ne peut contenir un ensemble d'articulation contenant uniquement 2 sommets. ■

Démonstration. (Lemme III.4.5) Procédons par l'absurde et supposons que, quelle que soit l'arête $e \in E$, le graphe $G \cdot e$ possède un ensemble de coupure S tel que $\#S = 2$ (on suppose donc que $G \cdot e$ n'est plus 3-connexe). Soient $e = \{u, v\}$ et x le sommet obtenu après contraction de e . Ce sommet x appartient à S car sinon, G lui-même posséderait S comme ensemble de coupure (or, par hypothèse G est 3-connexe). Ainsi, on a $S = \{x, z\}$ pour un certain sommet z . On peut en conclure que

$$T = \{u, v, z\}$$

est un ensemble de coupure de G . Puisqu'un tel ensemble existe, on choisit⁷ e et z de manière telle que $G - T$ possède une composante connexe A contenant un nombre minimal de sommets.

⁷Il n'y a qu'un nombre fini de choix possibles. Il suffit de les passer tous en revue.

Il existe dans A un sommet y tel que $\{y, z\}$ est une arête de E car sinon, $\{u, v\}$ séparerait A du reste du graphe G (ce qui n'est pas possible car G est 3-connexe). Si on contracte l'arête $\{y, z\}$ dans le graphe G , au vu de notre

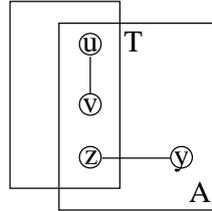


FIGURE III.11. Lemme III.4.5

supposition, le graphe $G \cdot \{y, z\}$ n'est pas 3-connexe. On applique donc à nouveau le même raisonnement et on en tire qu'il existe un sommet w tel que

$$R = \{y, z, w\}$$

sépare G (w peut très bien être égal à u ou à v , ce n'est pas interdit).

Si on regarde le graphe $G - R$, il possède une composante connexe B ne contenant ni u ni v . En effet, $\{u, v\}$ est une arête de G (donc soit les deux sommets appartiennent à une même composante distincte de B , soit, lorsque $w = u$ ou $w = v$, un de ces deux sommets est supprimé et l'autre appartient à une composante distincte de B). Cette composante connexe B contient un sommet y' tel que $\{y, y'\} \in E$. C'est le même raisonnement que précédemment, si ce n'était pas le cas, alors $\{z, w\}$ séparerait G .

Le sous-graphe de $G - T$ induit par les sommets de B est connexe. Puisque $y \in A$, $y' \in B$ et $\{y, y'\} \in E$, on en conclut que B est inclus dans A et cette inclusion est propre car y n'appartient pas à B . De là $\#B < \#A$ et on obtient une contradiction, vu le choix de A . ■

Démonstration. (Lemme III.4.6) Soit H , un sous-graphe de $G \cdot e$ homéomorphe à K_5 ou $K_{3,3}$ dont x est un sommet. On suppose de plus que ce sommet x de $G \cdot e$ est obtenu par contraction de $e = \{u, v\}$. Si x est de degré 2, alors il est immédiat que G possède la propriété **(K)**. Il reste à envisager les cas $\deg(x) = 3$ et $\deg(x) = 4$ (car $K_{3,3}$ est 3-régulier et K_5 est 4-régulier). S'il existe au plus une arête $\{x, y\}$ dans H telle que $\{u, y\}$ ou $\{v, y\}$ appartienne à E (ou exclusif), alors il est clair que G satisfait **(K)**.

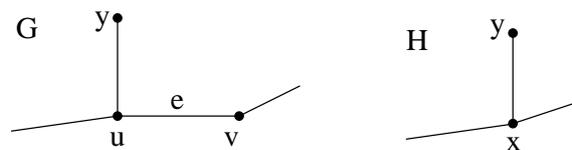


FIGURE III.12. Si on dispose d'au plus une arête.

Il reste donc le cas où à la fois u et v ont 3 voisins dans le sous-graphe de G correspondant à H . En particulier, H est donc homéomorphe à K_5 . Grâce à la figure III.13, on se convainc que G contient un sous-graphe homéomorphe à $K_{3,3}$.

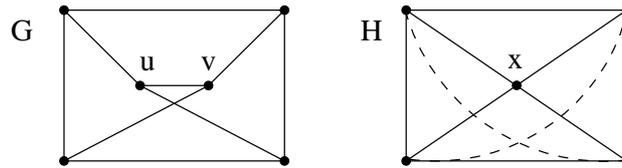


FIGURE III.13. G contient un sous-graphe homéomorphe à $K_{3,3}$.

■

Démonstration. (Lemme III.4.7) On procède par récurrence sur $\#V$. Le seul graphe 3-connexe à 4 sommets est K_4 et il est planaire. On peut donc supposer $\#V \geq 5$. Par le lemme III.4.5, il existe une arête $e = \{u, v\}$ telle que $G \cdot e$ (où l'arête contractée donne le sommet x) est 3-connexe. Par le lemme III.4.6, puisque G ne vérifie pas **(K)**, il en est de même pour $G \cdot e$. Puisque $G \cdot e$ possède un sommet de moins que G , on peut appliquer l'hypothèse de récurrence et en conclure que $G \cdot e$ est planaire.

Considérons une représentation planaire de $G \cdot e$ et C la frontière de la face de $(G \cdot e) - x$ qui contiendrait x . Puisque G est 3-connexe, C est un cycle.

Il est clair que $G - \{u, v\} = (G \cdot e) - x$. Par conséquent, $\nu_G(u) \subset C \cup \{v\}$ et $\nu_G(v) \subset C \cup \{u\}$ (où l'on a identifié le cycle C et l'ensemble de ses sommets). Supposons que, dans G , $\deg(v) \leq \deg(u)$ (on pourrait par symétrie traiter l'autre situation). Ordonnons les sommets de $\nu_G(v) \setminus \{u\} = \{v_1, \dots, v_k\}$ en respectant le cycle C . Soit $P_{i,j}$ le chemin le long de C joignant v_i à v_j par indices croissants. En se basant sur la représentation planaire de $G - \{u, v\}$, on obtient une représentation planaire de $G - u$ en traçant les segments de droites $\{v, v_i\}$.

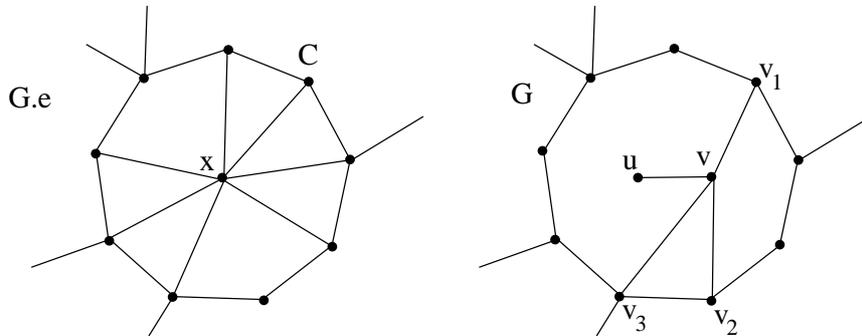


FIGURE III.14. Illustration du lemme III.4.7.

- (a) Si $\nu_G(u) \setminus \{v\} \subseteq P_{i,i+1}$ (où $i + 1$ est interprété modulo k) pour un certain i , alors G est planaire. Il suffit de positionner u dans le triangle v, v_i, v_{i+1} .

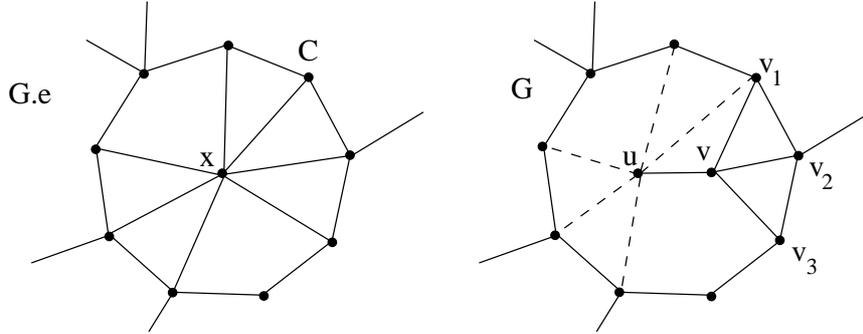


FIGURE III.15. Illustration du lemme III.4.7, cas (a).

- (b) S'il existe $y, z \in \nu_G(u) \setminus \{u\}$ tels que $y \in P_{i,j}$ et $z \notin P_{i,j}$ avec $y, z \notin \{v_i, v_j\}$. Dans ce cas, $\{u, v_i, v_{i+1}\}$ et $\{v, z, y\}$ forment une copie homéomorphe à $K_{3,3}$, ce qui est impossible.
- (c) Le dernier cas à envisager est donc $\nu_G(u) \setminus \{v\} \subset \nu_G(v)$. Or nous avons supposé que $\deg(v) \leq \deg(u)$, de là,

$$\nu_G(u) \setminus \{v\} = \nu_G(v) \setminus \{v\} \text{ et } \deg(u) = \deg(v).$$

Si $\deg(u) = \deg(v) \leq 3$, le graphe G est planaire. Sinon, il contient une copie homéomorphe à K_5 .

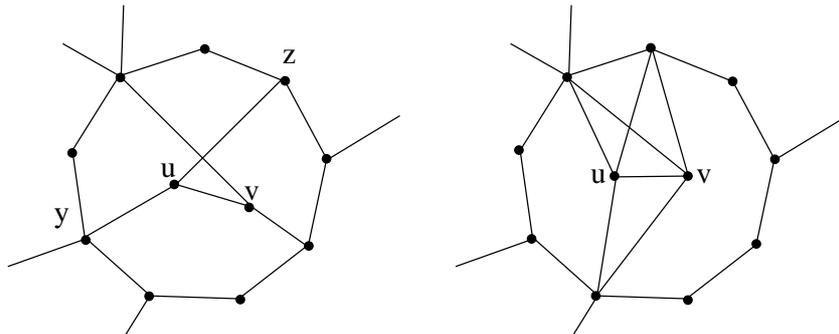


FIGURE III.16. Illustration du lemme III.4.7, cas (b) et (c).

■

CHAPITRE IV

Coloriage

Dans ce chapitre, on considère le problème de coloriage de sommets d'un graphe, le théorème des cinq couleurs et une introduction à la théorie de Ramsey. Nous supposons rencontrer des multi-graphes non orientés (les boucles n'ont ici que peu d'importance).

1. Nombre chromatique

Définition IV.1.1. Soit $c : V \rightarrow \Sigma$ un étiquetage des sommets du graphe où l'on suppose que Σ est fini. Dans le contexte qui nous intéresse, on parlera plutôt de *coloriage* et pour tout sommet u , on dira que $c(u)$ est la couleur de u . Un coloriage c des sommets d'un graphe est dit *propre* si deux sommets voisins ont des couleurs distinctes pour c .

S'il existe un coloriage propre $c : V \rightarrow \Sigma$ de G tel que $\#\Sigma = k$, on dira que G est k -colorable. La valeur minimum de k pour laquelle G est k -colorable est appelée le *nombre chromatique* de G et est noté $\chi(G)$.

Remarque IV.1.2. Pour obtenir un coloriage propre, on peut attribuer à des sommets indépendants (définition I.5.1) une même couleur. Ainsi, le nombre chromatique d'un graphe $G = (V, E)$ est le nombre minimum de sous-ensembles de sommets indépendants nécessaires pour partitionner V .

Lemme IV.1.3. *Le nombre chromatique d'un graphe $G = (V, E)$ (non orienté et simple) est le plus petit entier n tel qu'il existe un homomorphisme de G dans K_n .*

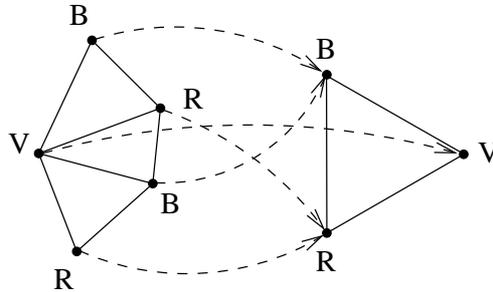


FIGURE IV.1. Une illustration du lemme IV.1.3.

Démonstration. Soient $H = (V', E')$ un graphe simple et $f : V \rightarrow V'$ un homomorphisme de graphes. Pour tout $y \in V'$,

$$f^{-1}(y) = \{x \in V : f(x) = y\}.$$

Puisque H est simple, il ne contient aucune boucle et donc, $f^{-1}(y)$ est formé de sommets indépendants (cf. définition I.5.1).

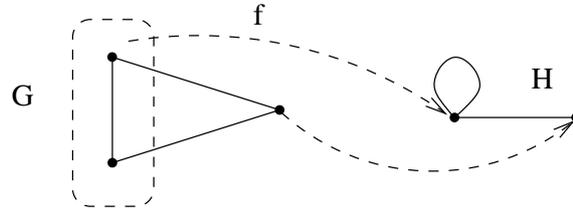


FIGURE IV.2. Si H n'est pas simple.

Par conséquent, s'il existe un homomorphisme de G vers un graphe simple à n sommets (comme K_n par exemple), alors G est n -colorable ($\chi(G) \leq n$). Réciproquement, si G est n -colorable, alors l'application qui envoie les sommets de G d'une même couleur sur un sommet de K_n est un homomorphisme.

■

Proposition IV.1.4. *Un graphe $G = (V, E)$ est biparti si et seulement si il est 2-colorable.*

Démonstration. C'est immédiat.

■

Remarque IV.1.5. Un graphe $G = (V, E)$ est 1-colorable si et seulement si $E = \emptyset$.

2. Le théorème des cinq couleurs

On peut reprendre le problème de cartographie énoncé à l'exemple I.3.4. Tout d'abord, on peut remarquer que par *projection stéréographique*, tout problème de planarité ou de coloriage de graphes sur la sphère revient à un problème analogue dans le plan.

Le lecteur attentif aura remarqué que le problème de cartographie demande de colorier des faces adjacentes d'un graphe planaire avec des couleurs distinctes. Pourtant, dans la section précédente, nous nous sommes intéressés au coloriage des sommets d'un graphe. On pourrait donc penser qu'il s'agit d'un nouveau type de problème. En fait, il n'en est rien comme nous allons le voir en introduisant le dual d'un graphe planaire.

Définition IV.2.1. Soit $G = (V, E)$ un multi-graphe planaire. Le dual G^* de G est le graphe dont les sommets sont les faces de G . A chaque arête de G appartenant à la frontière de deux faces F et F' , il correspond une arête dans G^* joignant les sommets correspondants aux faces F et F' . Il est clair que le dual d'un graphe planaire est encore planaire et que $(G^*)^* = G$.

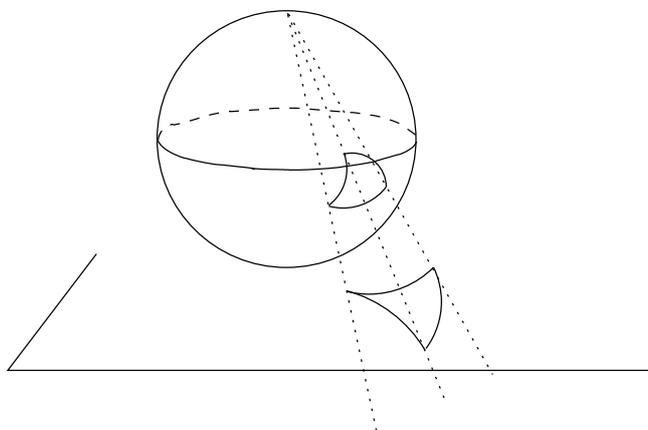


FIGURE IV.3. Projection stéréographique.

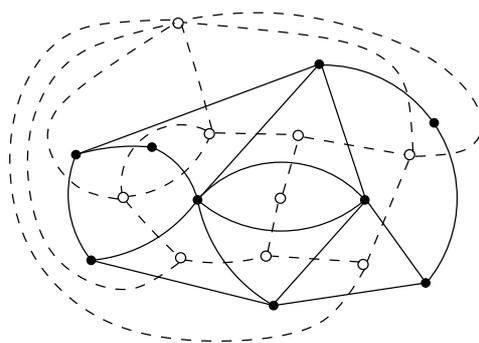


FIGURE IV.4. Un graphe et son dual.

Ainsi grâce au dual, colorier les faces d'un graphe planaire, revient à colorier les sommets de son dual et inversement.

Remarque IV.2.2. La trace la plus ancienne du “problème des quatre couleurs” remonte à une lettre de De Morgan datée du 23 octobre 1852 et destinée à Hamilton : “*A student of mine asked me today to give him reason for a fact which I did not know was a fact and do not yet. He says that if a figure be anyhow divided and the compartments differently colored, so that figures with any portion of common boundary line are differently coloured — four colours may be wanted but no more.*”. L'étudiant dont il est question dans la lettre est Frederick Guthrie. Futur physicien, il avait pris connaissance de ce problème par son frère Francis, futur mathématicien, qui avait fait cette conjecture en observant la carte d'Angleterre. Un résumé intéressant de l'histoire du théorème des quatre couleurs peut être trouvé dans¹ “*Four Colours Suffice, or how to colour a map*, R. Wilson, European Math. Society Newsletter **46** December 2002, 15–19.”

¹<http://www.emis.de/newsletter/newsletter46.pdf>



FIGURE IV.5. Francis Guthrie.

Dans la suite, nous nous intéressons donc au coloriage des faces d'un multi-graphe planaire de manière telle que deux faces adjacentes (y compris la face infinie) reçoivent des couleurs distinctes. On parlera alors simplement de *coloriage*, les contraintes précitées étant sous-entendues.

Par convention, les couleurs seront désignées par des lettres grecques minuscules. Il est tout d'abord clair qu'il existe des graphes pour lesquels **quatre couleurs sont nécessaires**. Il suffit de considérer le graphe de la figure III.4.

Nous allons ici démontrer une version plus faible que le véritable théorème des quatre couleurs. Notre présentation est basée sur celle d'Ore [20]. En effet, nous allons démontrer le théorème suivant. Ce résultat contient les idées essentielles du théorème des quatre couleurs et présente l'avantage de pouvoir être démontré de manière directe.

Théorème IV.2.3. *Cinq couleurs suffisent pour colorier les faces d'un multi-graphe planaire de manière telle que deux faces adjacentes reçoivent des couleurs distinctes.*

Pour démontrer ce résultat, on peut tout d'abord se restreindre à des multi-graphes **3-réguliers**. Tout d'abord, les sommets de degré 1 peuvent être éliminés. Ils n'interviennent pas dans la définition d'une face. Ensuite, on peut modifier le graphe comme à la figure IV.6 pour ne plus avoir de sommets de degré 2. Il est clair que cette opération ne modifie en rien un

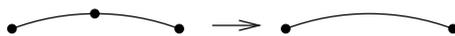
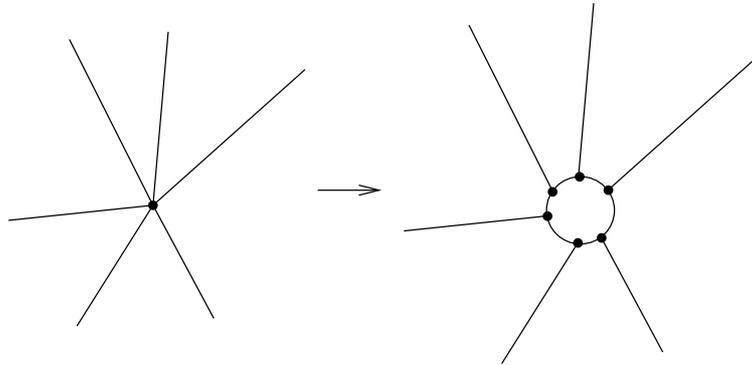


FIGURE IV.6. Suppression des sommets de degré 2.

coloriage valide. Ainsi, le graphe résultant possède un coloriage valide si et seulement si le graphe de départ en possède un.

Nous pouvons enfin modifier le graphe comme à la figure IV.7 pour ne plus avoir de sommets de degré ≥ 4 . Si le graphe résultant possède un coloriage valide avec le graphe de départ aussi.

FIGURE IV.7. Suppression des sommets de degré ≥ 4 .

Ainsi, nous pouvons considérer avoir un graphe planaire 3-régulier G . Si nous sommes capables de colorier les graphes planaires 3-réguliers, nous serons capables de colorier n'importe quel graphe planaire. Dénotons par φ_i , le nombre de faces de G dont la frontière est déterminée par exactement i arêtes (ou de manière équivalente, par i sommets), $i \geq 2$. Le graphe étant 3-régulier, chaque sommet appartient à 3 faces. Si pour chaque face, nous en comptons les sommets², on obtient

$$3s = 2\varphi_2 + 3\varphi_3 + 4\varphi_4 + 5\varphi_5 + \dots .$$

Si pour chaque face, nous en comptons les arêtes³, on obtient

$$2a = 2\varphi_2 + 3\varphi_3 + 4\varphi_4 + 5\varphi_5 + \dots .$$

Enfin, il est clair que

$$f = \varphi_2 + \varphi_3 + \varphi_4 + \varphi_5 + \dots .$$

Si on substitue les termes apparaissant dans la formule d'Euler $12 = 6s - 6a + 6f$ par les valeurs obtenues ci-dessus, on obtient la relation

$$12 = 4\varphi_2 + 3\varphi_3 + 2\varphi_4 + \varphi_5 - \varphi_7 - 2\varphi_8 - \dots .$$

Or les φ_i sont positifs ou nuls. Dès lors, on en tire le résultat suivant.

Lemme IV.2.4. *Tout multi-graphe planaire 3-régulier contient une face dont la frontière est délimitée par 2, 3, 4 ou 5 arêtes.*

Notre but est de considérer, dans le graphe G , une face délimitée par 2, 3, 4 ou 5 arêtes et d'en supprimer une arête. (L'existence d'une telle face est assurée par le lemme IV.2.4.) De cette façon, on obtient un graphe G' ayant **au moins une face de moins** que le graphe G de départ. Les constructions que nous allons développer auront deux propriétés.

²Chaque sommet est donc compté trois fois.

³Chaque arête est donc comptée deux fois car elle appartient à la frontière d'exactly deux faces.

R.1 La construction préserve le caractère 3-régulier (on pourra donc l'appliquer itérativement).

R.2 Si le graphe G' peut être colorié avec au plus 5 couleurs, alors il en est de même pour le graphe de départ G .

On parle en général de *réduction*. On s'est ramené à un problème pour un graphe ayant moins de faces que le graphe initial.

Ainsi, grâce à la propriété **R.1**, chaque étape fournit un graphe 3-régulier. Au vu du lemme IV.2.4, on peut opérer une nouvelle réduction et diminuer le nombre de faces du graphe. En répétant la procédure, on obtiendra un graphe ayant au plus 5 faces qu'il est donc toujours possible de colorier ! La propriété **R.2** assure alors que le graphe de départ peut, lui aussi, être correctement colorié.

• **Réduction d'une face délimitée par deux arêtes**

Considérons la situation donnée à la figure IV.8 et sa réduction. Il est facile

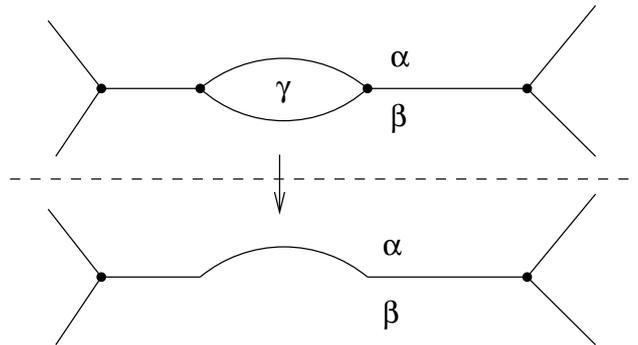


FIGURE IV.8. Réduction d'une face délimitée par deux arêtes.

de se convaincre que les propriétés **R.1** et **R.2** sont satisfaites.

• **Réduction d'une face délimitée par trois arêtes**

Considérons la situation donnée à la figure IV.9 et sa réduction. Il est facile de se convaincre que les propriétés **R.1** et **R.2** sont satisfaites.

• **Réduction d'une face délimitée par quatre arêtes**

Ici, la situation est un peu plus délicate. Si une face F est délimitée par quatre arêtes, comme on le voit à la figure IV.10, quatre faces A, B, C, D peuvent lui être adjacentes. Dans le premier cas, A et C ne sont pas adjacentes, dans le deuxième cas (gardez à l'esprit que le graphe doit être 3-régulier), elles ont une arête commune. Mais **dans tous les cas**, B et D ne font pas partie d'une même face et ne sont pas adjacents (ils pourront donc recevoir la même couleur). Enfin, dans le dernier cas, A et C constituent une seule face. Il faut se convaincre qu'il s'agit des seules situations envisageables (quitte à renommer les différentes faces). En effet, il n'est pas possible dans le plan que simultanément, les faces A et C soient adjacentes de même que B et D . On peut supprimer certains sommets et arêtes pour

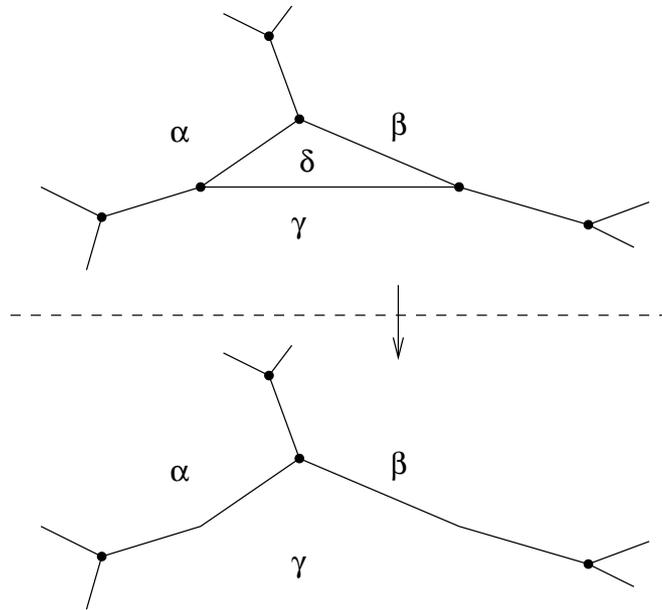


FIGURE IV.9. Réduction d'une face délimitée par deux arêtes.

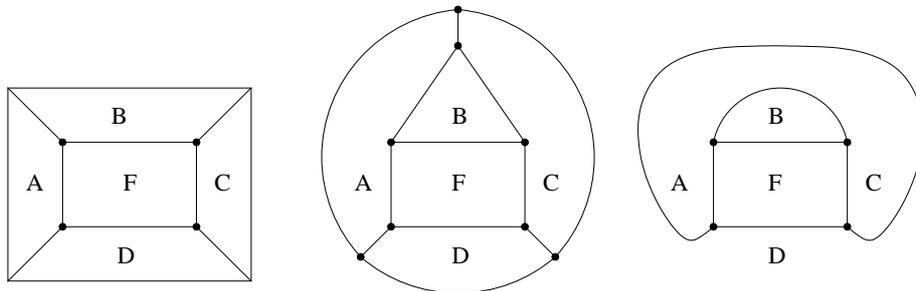


FIGURE IV.10. Configurations pour la réduction d'une face rectangulaire.

obtenir les configurations de la figure IV.11. Ces dernières peuvent être coloriées et la propriété **R.1** est bien satisfaite. Ensuite, il reste à se convaincre,

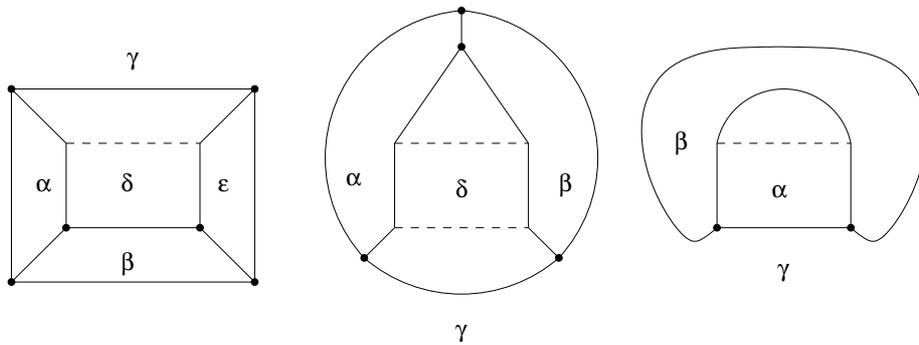


FIGURE IV.11. Réductions d'une face rectangulaire.

grâce à la figure IV.12 que **R.2** est aussi satisfait.

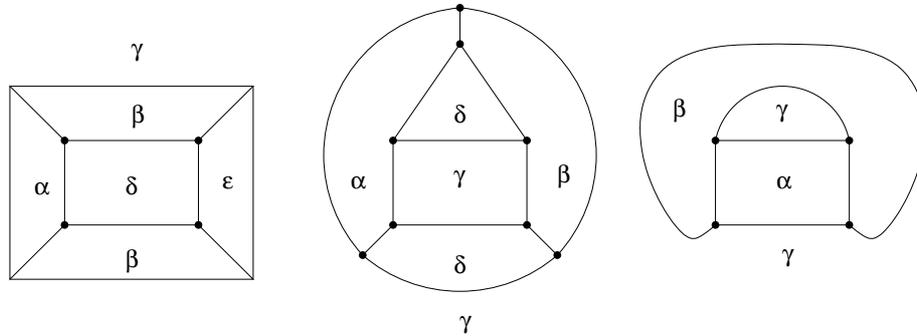


FIGURE IV.12. Satisfaction de **R.2**.

• **Réduction d'une face délimitée par cinq arêtes**

Le raisonnement est semblable au cas précédent. Il existe toujours deux faces opposées, disons A et C , adjacentes à une face pentagonale F telles que A et C ne font pas partie d'une même face et ne sont pas adjacentes. On modifie le graphe comme indiqué à la figure IV.13. Les faces B , D et E peuvent nécessiter 3 couleurs distinctes α , β et γ . Ayant 5 couleurs à notre disposition, il est possible d'assurer **R.2**. Cela ne serait pas le cas avec uniquement 4 couleurs. Ceci démontre entièrement le théorème des cinq

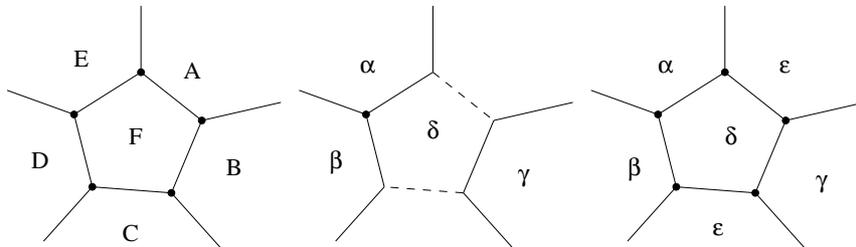


FIGURE IV.13. Réductions d'une face pentagonale.

couleurs. Comme le montre la figure IV.14, dans le cas de face pentagonale, l'utilisation de 5 couleurs est inévitable (en utilisant la méthode préconisée ci-dessus, à savoir donné la même couleur aux deux faces non adjacentes). Bien sûr, il est possible de réaliser un coloriage ne nécessitant pour cela que 4 couleurs, mais dès lors, on sort de la méthode de réduction proposée.

Enonçons enfin le théorème des quatre couleurs. La démonstration de ce dernier nécessite le même type de réductions que celles effectuées pour le théorème des cinq couleurs mais ici, le nombre de configurations à considérer est de l'ordre de 600 pour la preuve fournie par N. Robertson, D. Sanders, P. Seymour et R. Thomas (1996). La preuve originale de K. Appel et W. Haken utilisait quant à elle près de 2000 configurations inévitables. Par ailleurs, cette dernière preuve a en fait été partiellement réalisée par un

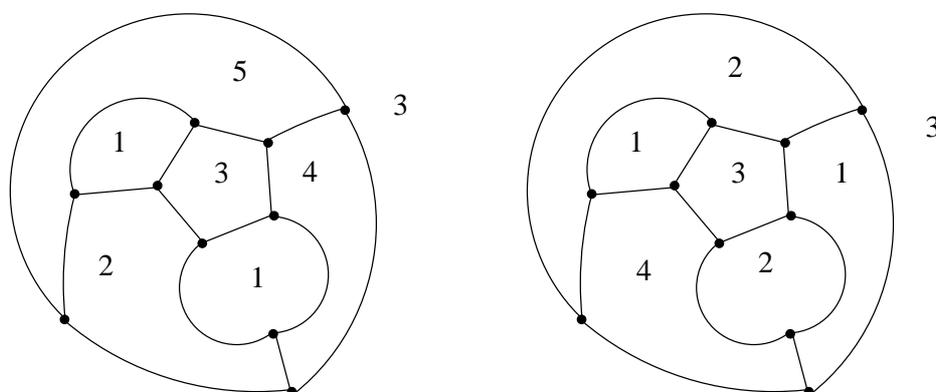


FIGURE IV.14. Cinq couleurs effectives.

programme informatique⁴ passant en revue et testant l'ensemble de ces configurations.

Théorème IV.2.5 (K. Appel, W. Haken (1976)). *Quatre couleurs suffisent pour colorier les faces d'un multi-graphe planaire de manière telle que deux faces adjacentes reçoivent des couleurs distinctes.*

Remarque IV.2.6. On pourrait s'intéresser à des questions de coloriage (ou de planarité) pour des graphes tracés sur d'autres surfaces que la sphère ou le plan. On dispose par exemple du *théorème des sept couleurs* : Sept couleurs sont suffisantes pour colorier tout multi-graphe planaire sur un tore. De plus, il existe un multi-graphe planaire sur un tore nécessitant sept couleurs.

Remarque IV.2.7. Le *genre* d'une surface est un invariant topologique. Il s'agit du nombre maximum de courbes simples fermées que l'on peut tracer sur la surface sans la disconnecter. De manière grossière, cela correspond

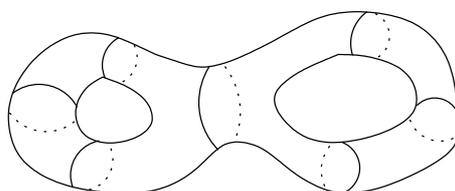


FIGURE IV.15. Une surface de genre 2.

au nombre de "trous" que la surface comporte. Ainsi, le genre d'une sphère vaut 0 et celui d'un tore vaut 1. On dispose de la formule d'Heawood (1980) qui stipule que si un graphe peut être représenté de manière planaire sur

⁴Cela pose d'ailleurs la question de ce que l'on entend par preuve mathématique. Certains mathématiciens n'acceptent pas aisément une "preuve assistée par ordinateur" et mettent par exemple en cause d'hypothétiques failles matérielles ("on n'a pas démontré l'exactitude des résultats fournis par le processeur, le compilateur, ...").

g	0	1	2	3	4	5	6	7	8	9	10
c_g	4	7	8	9	10	11	12	12	13	13	14

TABLE IV.1. Les premières valeurs de c_g .

une surface de genre g , alors ses faces peuvent être colorées avec c_g couleurs où

$$c_g = \left\lfloor \frac{1}{2}(7 + \sqrt{1 + 48g}) \right\rfloor.$$

La partie délicate consiste à montrer qu'il existe effectivement un graphe nécessitant ce nombre de couleurs. En particulier, pour $g = 0, 1$, on retrouve le théorème des 4 et des 7 couleurs.

3. Polynôme chromatique

Jusqu'à présent, nous nous sommes intéressés à la possibilité de colorier ou non un graphe donné à l'aide de k couleurs. La question suivante serait de déterminer le nombre de façons distinctes de réaliser un tel coloriage.

Définition IV.3.1. Soit $G = (V, E)$ un multi-graphe non orienté ayant n sommets. On dénote par $m_{k,G}$, le nombre de coloriages propres distincts⁵ de G utilisant **exactement** k couleurs et par $z^{\underline{k}}$, le polynôme⁶ en $z \in \mathbb{C}$ de degré k

$$z^{\underline{k}} = z(z-1)\cdots(z-k+1).$$

Avec ces notations, le *polynôme chromatique* de G est donné par

$$\begin{aligned} \pi_G(z) &= \sum_{i=1}^n \frac{m_{i,G}}{i!} z^{\underline{i}} \\ &= \frac{m_{1,G}}{1!} z + \frac{m_{2,G}}{2!} z(z-1) + \frac{m_{3,G}}{3!} z(z-1)(z-2) + \cdots \\ &\quad \cdots + \frac{m_{n,G}}{n!} z(z-1)\cdots(z-n+1). \end{aligned}$$

Il s'agit d'un polynôme de degré n en la variable z .

Remarque IV.3.2. La quantité

$$\frac{m_{k,G}}{k!}$$

est le nombre de partitions de V en k sous-ensembles (disjoints et non vides) donnant lieu à tous les coloriages propres possibles de G utilisant exactement k couleurs. En fait, cette quantité équivaut au *nombre de partitions de V en k sous-ensembles (disjoints et non vides) de sommets indépendants*.

Exemple IV.3.3. Soit le graphe de la figure IV.16. Considérons les par-

⁵On suppose que les sommets de G sont énumérés une fois pour toutes : $V = \{v_1, \dots, v_n\}$ et que les k couleurs sont données par $\{1, \dots, k\}$. On énumère donc les fonctions de coloriage $c : V \rightarrow \Sigma$ qui sont propres.

⁶Il s'agit d'une notation classique "falling power" que l'on retrouve chez Graham, Knuth et Patashnik. Remarquez que $1, z, \dots, z^{\underline{k}}$ forment une base de l'ensemble des polynômes de degré au plus k .

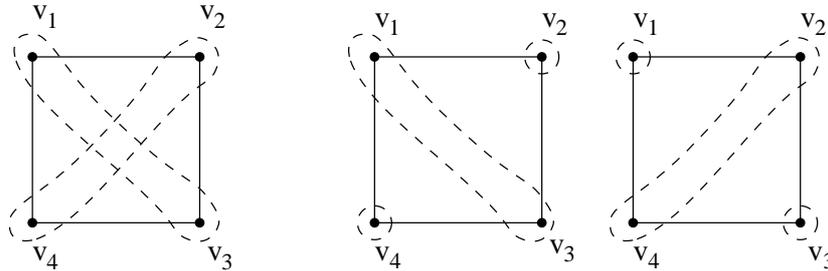


FIGURE IV.16. Illustration de $m_{k,G}$.

titions de V en k sous-ensembles (disjoints et non vides) donnant lieu à tous les coloriages propres possibles de G utilisant exactement k couleurs, $k = 2, 3$. Pour $k = 2$, on a la partition $V = \{v_1, v_3\} \cup \{v_2, v_4\}$ et les coloriages $c_1 : v_1, v_3 \mapsto 1, v_2, v_4 \mapsto 2$ et $c_2 : v_1, v_3 \mapsto 2, v_2, v_4 \mapsto 1$. Ainsi, $m_{2,G} = 2$ et $m_{2,G}/2! = 1$ correspond bien à la seule partition convenable de V . Pour $k = 3$, on a deux partitions possibles de V en $\{v_1, v_3\} \cup \{v_2\} \cup \{v_4\}$ ou bien $\{v_1\} \cup \{v_3\} \cup \{v_2, v_4\}$. Chaque partition donne lieu à $3!$ coloriages propres distincts de G . Ainsi, $m_{3,G} = 12$ et $m_{3,G}/3! = 2$. Pour $k = 4$, il y a une seule partition de V en quatre singletons donc $m_{4,G}/4! = 1$. Par conséquent, le polynôme chromatique du graphe est donné par

$$\underbrace{m_{1,G}}_{=0} z + \underbrace{\frac{m_{2,G}}{2!}}_{=1} z(z-1) + \underbrace{\frac{m_{3,G}}{3!}}_{=2} z(z-1)(z-2) + \underbrace{\frac{m_{4,G}}{4!}}_{=1} z(z-1)(z-2)(z-3)$$

ou encore

$$\pi_G(z) = z^4 - 4z^3 + 6z^2 - 3z.$$

Exemple IV.3.4. Pour le graphe complet K_n , les seuls ensembles de sommets indépendants sont les singletons. Ainsi, pour tout $k < n$,

$$\frac{m_{k,K_n}}{k!} = 0 \quad \text{et} \quad \pi_{K_n}(z) = z^n.$$

Remarque IV.3.5. Quelques remarques immédiates.

- ▶ Si G possède n sommets, alors $m_{n,G} = n!$ car on assigne une couleur par sommet. On en déduit que le polynôme chromatique est monique.
- ▶ Si G n'est pas connexe mais possède 2 composantes G_1 et G_2 , alors

$$\pi_G(z) = \pi_{G_1}(z) \cdot \pi_{G_2}(z).$$

Cela résulte du fait que les sommets de G_1 peuvent être colorés indépendamment de ceux de G_2 .

- ▶ Il est clair que $\pi_G(0) = 0$ pour tout graphe G .
- ▶ Il est impossible de colorier un graphe non vide avec aucune couleur.

Proposition IV.3.6. Soit $k \in \mathbb{N}$. Le nombre $\pi_G(k)$ est le nombre de coloriages propres de G utilisant au plus k couleurs.

Exemple IV.3.7. Avec le graphe de la figure IV.16, si on calcule $\pi_G(4)$, on trouve $\pi_G(4) = 84$ et donc, il y est possible de colorier proprement G avec au plus 4 couleurs de 84 façons distinctes.

Démonstration. Tout d'abord, il est inutile de considérer dans l'expression de $\pi_G(k)$, les termes d'exposant $> k$. En effet, si $i > k$, alors z^i évalué en k est nul. Ainsi, on a

$$\pi_G(k) = \sum_{i=1}^k \frac{m_{i,G}}{i!} k^i.$$

Ensuite, pour tout $i \leq k$, on dispose de $\frac{m_{i,G}}{i!}$ partitions de V en i sous-ensembles de sommets indépendants. Considérons ces partitions et autorisons-nous cette fois à choisir i couleurs parmi k , pour chaque partition $C_1 \cup \dots \cup C_i$ de V , à C_1 , on peut assigner k couleurs, pour C_2 , on a $k-1$ choix possibles, ... et pour C_i , on a $k-i+1$ choix possibles. Ainsi, chacune des $\frac{m_{i,G}}{i!}$ partitions de V en i sous-ensembles donnent lieu à k^i coloriages utilisant exactement i des k couleurs disponibles. La conclusion en découle. ■

Exemple IV.3.8. En continuant l'exemple précédent, le nombre de coloriages du graphe repris à la figure IV.16 utilisant au plus 3 couleurs vaut

$$\pi_G(3) = \underbrace{m_{1,G}}_{=0} 3 + \frac{m_{2,G}}{2!} 3 \cdot 2 + \frac{m_{3,G}}{3!} 3 \cdot 2 \cdot 1 = 0 + 6 + 2 \cdot 6 = 18.$$

Ces coloriages sont donnés par

v_1	v_2	v_3	v_4
1	2	1	2
2	1	2	1
1	3	1	3
3	1	3	1
2	3	2	3
3	2	3	2

v_1	v_2	v_3	v_4
1	2	1	3
1	3	1	2
2	1	2	3
2	3	2	1
3	1	3	2
3	2	3	1

v_1	v_2	v_3	v_4
2	1	3	1
3	1	2	1
1	2	3	2
3	2	1	2
1	3	2	3
2	3	1	3

Corollaire IV.3.9. *Le nombre chromatique de G est le plus petit entier k tel que $\pi_G(k) \neq 0$.*

Ainsi, la détermination du nombre chromatique d'un graphe à n sommets peut se ramener à l'estimation des zéros d'un polynôme de degré n . Par exemple, avec le graphe repris à la figure IV.16, on a $\pi_G(1) = 0$ et $\pi_G(2) \neq 0$.

Le polynôme chromatique d'un graphe peut être obtenu par une technique semblable à celle développée pour obtenir la formule de Cayley comptant le nombre de sous-arbres couvrants (proposition II.5.4).

Proposition IV.3.10. *Si e est une arête de G (qui n'est pas une boucle), alors le polynôme chromatique satisfait la relation*

$$\pi_G(z) = \pi_{G-e}(z) - \pi_{G \cdot e}(z).$$

Démonstration. Si on considère tous les coloriages propres de $G - e$ avec exactement k couleurs. Il y en a de deux types : ceux pour lesquels on assigne aux extrémités de e deux couleurs distinctes (resp. une même couleur). Ceux du premier type sont en bijection avec les coloriages propres de G utilisant k couleurs. Pour conclure, on remarque que ceux du second type sont en bijection avec les coloriages propres de $G \cdot e$ utilisant k couleurs. ■

Corollaire IV.3.11. *Le polynôme chromatique d'un arbre T à n sommets vaut*

$$\pi_T(z) = z(z-1)^{n-1}.$$

Démonstration. On procède par récurrence sur n . Le cas $n = 1$ est immédiat. Supposons le résultat acquis pour $n \geq 1$ et vérifions-le pour $n + 1$. Si un arbre possède $n + 1$ sommets, il a au moins un sommet de degré 1 et soit e , l'arête incidente à ce sommet. Ainsi, $T - e$ possède deux composantes : un sommet isolé dont le polynôme chromatique vaut z et un arbre à n sommets (qui n'est autre que la contraction $T \cdot e$) de polynôme chromatique $z(z-1)^{n-1}$ (par hypothèse de récurrence). On en conclut que

$$\pi_{T-e}(z) = z \pi_{T \cdot e}(z).$$

Par la proposition précédente,

$$\pi_T(z) = \pi_{T-e}(z) - \pi_{T \cdot e}(z) = (z-1) \pi_{T \cdot e}(z) = z(z-1)^n.$$

■

4. Coloriage d'arêtes et théorème de Ramsey

Bien évidemment, on peut considérer le problème de coloriage "dual"⁷ consistant à vouloir attribuer à chaque arête d'un graphe une couleur de façon telle que deux arêtes adjacentes reçoivent des couleurs distinctes. Nous préférons dans cette section introduire le célèbre théorème de Ramsey qui possède de nombreuses applications dans diverses branches des mathématiques⁸; aussi bien en théorie des nombres qu'en analyse harmonique ou qu'en géométrie.

La question que l'on se pose est la suivante. Etant donné un entier s , existe-t-il un entier n (dépendant de s) tel pour tout coloriage des arêtes de K_n avec deux couleurs (par convention, le rouge et le bleu), K_n contient un sous-graphe K_s formé d'arêtes d'une même couleur ? Il n'est pas évident qu'*a priori* cette question possède une solution. En effet, on pourrait penser que certains coloriages permettent de mettre en défaut la propriété désirée quel que soit n . En fait, il n'en est rien, pour n suffisamment grand, il existe toujours dans K_n une copie de K_s monochromatique et ce, quel que soit le coloriage envisagé.

⁷Au sens, dualisme sommet/arête.

⁸V. Rosta, *Ramsey Theory Applications*, www.combinatorics.org/Surveys/ds13.pdf

En fait, nous allons définir le nombre $R(s, t)$ comme étant le plus petit n tel que K_n contienne une copie de K_s rouge ou une copie de K_t bleue. Il nous faudra bien sûr **montrer que ces nombres existent** (cf. théorème IV.4.3). Mais si de tels nombres existent, par définition même de $R(s, t)$, on peut énoncer le résultat suivant.

Théorème IV.4.1 (Ramsey (1930)). *Il existe un plus petit entier $R(s, t)$ tel que pour tout $n \geq R(s, t)$, tout coloriage de $K_n = (V, E)$, $c : E \rightarrow \{\mathcal{R}, \mathcal{B}\}$, est tel que G contient une copie de K_s colorée en \mathcal{R} ou une copie de K_t colorée en \mathcal{B} .*

Il est tout d'abord clair que $R(s, t) = R(t, s)$ pour tous $s, t \geq 2$. En effet, il suffit d'inverser les couleurs rouge et bleue attribuées aux différentes arêtes dans les coloriages envisagés.

De plus,

$$(10) \quad R(s, 2) = R(2, s) = s.$$

En effet, dans tout coloriage des arêtes de K_s , au moins une arête est bleue ou alors elles sont toutes rouges. De plus, il est clair qu'une valeur inférieure à s ne peut convenir.

Exemple IV.4.2. Un rapide exemple montre que $R(3, 3) > 5$. En effet, il suffit d'exhiber un coloriage des arêtes de K_5 ne contenant aucune copie de K_3 monochromatique. A la figure IV.17, on distingue les deux couleurs grâce à des traits pleins et des traits pointillés. Pour vérifier que $R(3, 3) = 6$, il faut

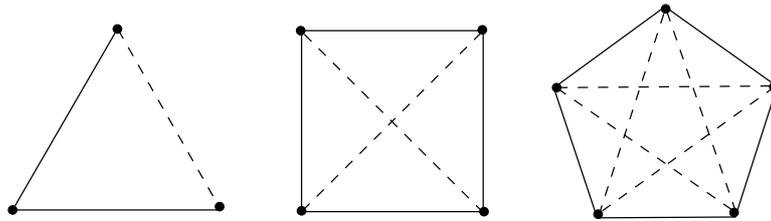


FIGURE IV.17. $R(3, 3) > 3, 4, 5$.

passer en revue tous les coloriages de K_6 et s'assurer qu'ils contiennent tous une copie de K_3 monochromatique ! Ainsi, K_n contient C_n^2 arêtes pouvant chacune être coloriée en rouge ou en bleu. Le tableau suivant reprend le nombre de coloriages possibles des arêtes de K_n pour les premières valeurs

de n ,

n	$2^{\binom{n}{2}}$
3	8
4	64
5	1024
6	32768
7	2097152
8	268435456
9	68719476736
10	35184372088832
11	36028797018963968
12	73786976294838206464
13	302231454903657293676544
14	2475880078570760549798248448

Il nous suffit de démontrer le résultat suivant.

Théorème IV.4.3 (Erdős, Szekeres (1935)). *Pour tous $s, t \geq 2$, le nombre $R(s, t)$ existe. De plus, on a*

$$R(s, t) \leq C_{s+t-2}^{s-1}$$

et si $s, t \geq 3$, alors

$$R(s, t) \leq R(s-1, t) + R(s, t-1).$$

Démonstration. Pour la première partie de la preuve, on procède par récurrence sur $s+t$. Au vu de (10), on sait que $R(2, t)$ et $R(s, 2)$ existent toujours. Donc, en particulier, $R(s, t)$ existe si $s+t \leq 5$. Nous allons supposer que $R(s, t)$ existe pour tous s, t tels que $s+t < N$. Nous allons montrer que $R(s, t)$ existe encore pour $s+t = N$, avec $N \geq 6$. Si s ou t valent 2, le résultat est entièrement démontré. Nous pouvons donc supposer $s, t \geq 3$.

Il nous suffit de trouver un entier n tel que tout coloriage de K_n contient toujours une copie de K_s rouge ou de K_t bleue. De cette manière, on aura majoré $R(s, t)$. Pour cela, nous allons montrer que *tout graphe $G = (V, E)$ ayant $n = R(s-1, t) + R(s, t-1)$ sommets contient un sous-graphe isomorphe à K_s ou un ensemble de t sommets indépendants.*

FIGURE IV.18. Un graphe et un coloriage de K_n .

En effet, par hypothèse de récurrence, les nombres $R(s-1, t)$ et $R(s, t-1)$ existent. De plus, prendre un graphe arbitraire G ayant n sommets revient exactement à considérer un sous-graphe monochromatique de K_n (grâce à G , on sélectionne certaines arêtes qui reçoivent toutes la même couleur, les autres arêtes de K_n non présentes dans G recevant la seconde couleur⁹.)

Soit v un sommet de G . Désignons par $A_v = V \setminus (\nu(v) \cup \{v\})$, l'ensemble des sommets distincts de v et qui ne sont pas voisins de v . Puisque G a

$$\#V \setminus \{v\} = R(s-1, t) + R(s, t-1) - 1$$

sommets distincts de v , alors

$$\#\nu(v) \geq R(s-1, t) \text{ ou } \#A_v \geq R(s, t-1).$$

En effet, $\nu(v)$ et A_v partitionnent $V \setminus \{v\}$ (donc $\#\nu(v) + \#A_v = \#V \setminus \{v\}$).

Supposons tout d'abord que $\#\nu(v) \geq R(s-1, t)$. Par définition du nombre de Ramsey $R(s-1, t)$, le sous-graphe de G induit par $\nu(v)$ contient un sous-graphe B isomorphe à K_{s-1} ou un ensemble de sommets indépendants de taille t . Dans le premier cas, le sous-graphe induit par $B \cup \{v\}$ est isomorphe à K_s (en effet, v est adjacent à tous les sommets de B). Dans le second cas, on dispose directement d'un sous-ensemble de sommets indépendants de taille t .

Supposons à présent que $\#A_v \geq R(s, t-1)$. Par définition du nombre de Ramsey $R(s, t-1)$, le sous-graphe de G induit par A_v contient un sous-graphe isomorphe à K_s (ce qui suffit) ou un ensemble C de sommets indépendants de taille $t-1$. Dans ce dernier cas, $C \cup \{v\}$ est un ensemble de sommets indépendants de taille t (en effet, v n'est adjacent à aucun sommet de C).

L'inégalité est facile à vérifier. On procède une fois encore par récurrence sur $s+t$. Pour $s=2$ (ou $t=2$), on a bien

$$\underbrace{R(2, t)}_{=t} \leq \underbrace{C_t^1}_{=t}.$$

Supposons que $R(s, t) \leq C_{s+t-2}^{s-1}$ pour tous s, t tels que $s+t < N$ et vérifions-le pour $s+t = N$. Une fois encore, nous pouvons supposer $s, t \geq 3$. Par

⁹Par exemple, le graphe représenté à gauche de la figure IV.18 est un graphe quelconque à cinq sommets. Il lui correspond un coloriage de K_5 où toutes les arêtes présentes (resp. non présentes) dans le graphe de gauche reçoivent, dans le graphe de droite, la même couleur. Il y a bijection entre l'ensemble des graphes simples à n sommets et l'ensemble des coloriages bicouleurs des arêtes de K_n .

la première partie de la preuve et en utilisant l'hypothèse de récurrence, il vient

$$R(s, t) \leq R(s-1, t) + R(s, t-1) \leq C_{s+t-3}^{s-2} + C_{s+t-3}^{s-1}.$$

On conclut par la formule du triangle de Pascal.

■

Exemple IV.4.4. La détermination précise de $R(s, t)$ est extrêmement complexe (il est difficile d'un point de vue combinatoire d'énumérer tous les coloriage de K_n). Par exemple, la détermination de $R(3, 3)$ date de 1955, alors que l'estimation de $R(3, 12)$ reprise ci-contre date de 2001.

j	3	4	5	6	7	8	9	10	11	12
$R(3, j)$	6	9	14	18	23	28	36	[40, 43]	[46, 51]	[52, 59]

On consultera par exemple l'article "Small Ramsey Number, S. P. Radziszowski, <http://www.combinatorics.org/Surveys/ds1.pdf>" qui donne un état de l'art en la matière.

CHAPITRE V

Annexe : implémentation en C

1. Pointeurs, allocation mémoire et listes chaînées

Dans cette courte section, nous rappelons quelques fondements essentiels du langage C dans le but d'appréhender au mieux les structures données permettant de coder les graphes. Nos rappels concernent principalement la gestion et l'allocation dynamique de la mémoire. Nous supposons connus¹ les boucles, les fonctions ou encore la gestion des entrées/sorties.

1.1. Adresses. Toute variable possède une adresse mémoire. Cette adresse est accessible grâce à l'opérateur `&`. Autrement dit, si `x` est l'identificateur d'une variable, alors `&x` représente l'adresse de `x`.

Exemple V.1.1. Après compilation et exécution du court programme suivant,

```
#include <stdio.h>
main()
{
    int i;
    printf("> %u \n",&i);
}
```

vous devriez obtenir un affichage ressemblant à

```
> 3221219076
```

Ce programme affiche simplement (sous forme décimale) l'adresse mémoire² de la variable `i`.

On peut accéder à une variable non seulement grâce à son identificateur (si elle en possède un), mais aussi grâce à son adresse. Cela est rendu possible au moyen de l'opérateur `*`. L'adresse d'une variable est une donnée qui peut être stockée dans une variable. Une telle variable est appelée *pointeur*.

Exemple V.1.2. Dans cet exemple, on définit deux variables : un entier `i` et un pointeur vers un entier. Cette dernière variable va être utilisée pour stocker l'adresse de la première variable. Connaissant l'adresse de `i`, on peut modifier le contenu de `i` en se référant au pointeur vers ce dernier.

¹Consulter si nécessaire un ouvrage de référence sur le C comme [13].

²La gestion de la mémoire est laissée au système d'exploitation. Vous pouvez dès lors obtenir des résultats différents suivant la machine employée et même d'une exécution à l'autre sur une machine fixée.

```
#include <stdio.h>
main()
{
    int i;
    int *pointeur;
    i=10;
    pointeur=&i;
    *pointeur=55;
    printf("> %d \n",*pointeur);
}
```

Ce programme affiche : 55.

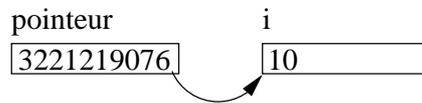


FIGURE V.1. Un pointeur et une variable.

Exemple V.1.3. Puisqu'un pointeur est lui-même une variable (qui contient l'adresse d'une autre variable), on peut définir un pointeur vers un pointeur (i.e., une variable contenant l'adresse d'un pointeur).

```
#include <stdio.h>
main()
{
    int a=1;
    int *p;
    int **q;
    printf("%d, ",a);
    p=&a;
    *p=2;
    printf("%d, ",a);
    q=&p;
    **q=3;
    printf("%d \n",a);
}
```

Affichage : 1, 2, 3.

L'utilisation des pointeurs présente deux intérêts majeurs : passer l'adresse d'une variable à une fonction et allouer dynamiquement de la mémoire.

Exemple V.1.4. Un exemple archi-classique consiste à définir une fonction inversant le contenu de deux variables. Voici d'abord un programme erroné.

```
/* Exemple erroné ----- */
#include <stdio.h>
void swap(int x, int y);
main()
{
    int a=1,b=2;
```

```
    printf("> a=%d, b=%d \n",a,b);
    swap(a,b);
    printf("> a=%d, b=%d \n",a,b);
}
```

```
void swap(int x, int y)
{
    int temp;
    temp=x;
    x=y;
    y=temp;
}
```

Lorsqu'on compile et exécute ce programme, on obtient l'affichage suivant et on n'a donc pas le résultat escompté.

```
> a=1, b=2
> a=1, b=2
```

La raison en est toute simple: lorsqu'une variable est passée à une fonction, C effectue une copie de cette variable et la fonction ne travaille donc pas sur les variables transmises mais sur une copie de celles-ci³. Lorsque la fonction s'achève, les copies sont détruites. On peut régler ce problème en transmettant l'adresse des variables à considérer. En plus de résoudre notre problème, cela permet de faire des économies de mémoire qui peuvent s'avérer importantes lorsqu'on transmet des variables de grande taille (dans ce cas, on transmet uniquement une adresse et ce n'est pas toute la variable qui est dupliquée). Voici à présent le programme corrigé. Le lecteur remarque la présence des opérateurs & et *.

```
/* Exemple correct ----- */
#include <stdio.h>
void swap(int *x, int *y);
main()
{
    int a=1,b=2;
    printf("> a=%d, b=%d \n",a,b);
    swap(&a,&b);
    printf("> a=%d, b=%d \n",a,b);
}
```

```
void swap(int *x, int *y)
{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
}
```

Cette fois, le programme a bien le comportement souhaité !

³On parle parfois de *transmission par valeurs*.

```
> a=1, b=2
> a=2, b=1
```

En effet, la fonction `swap` a accès au contenu des variables `a` et `b` grâce à leur adresse. On a schématiquement la situation représentée à la figure V.2.

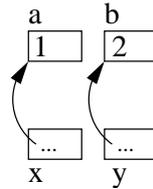


FIGURE V.2. L'échange de deux variables.

1.2. Allocation dynamique. Il se peut qu'en cours d'exécution d'un programme, il soit nécessaire d'affecter une valeur à une nouvelle variable. Si cette dernière n'a pas été déclarée avant la compilation, le programme ne peut réaliser cette affectation. On peut pallier à cet inconvénient grâce à une instruction⁴ comme `malloc`. Cette fonction crée un espace mémoire⁵ pour y stocker une donnée. Il faut donc lui préciser la taille de l'espace mémoire à réserver (En effet, chaque type de variable consomme une quantité plus ou moins grande de mémoire). De plus, `malloc` renvoie l'adresse de la zone mémoire qui vient d'être réservée. Cette dernière sera donc stockée par un pointeur. Si l'allocation mémoire n'est pas possible, `malloc` renvoie `NULL`.

Exemple V.1.5. Remarquer que, dans l'exemple qui suit, l'unique variable définie `p` est un pointeur vers un entier.

```
#include <stdio.h>
main()
{
    int *p;
    p=(int *) malloc(sizeof(int));
    *p=3;
    printf("adresse : %u contenu : %d \n",p,*p);
}
```

En fait, `malloc` renvoie un pointeur "générique". Ainsi, au lieu de taper simplement

```
p=malloc(sizeof(int));
```

on a effectué ce qu'on appelle un transtypage ("casting"), i.e., on transforme le pointeur "générique" en un pointeur vers un entier. Sans ce transtypage, le compilateur (ici `gcc`) produit en général un message d'alerte comme ci-dessous

```
test.c:5: warning: assignment makes pointer from integer without a cast
```

⁴voir aussi `realloc` et `calloc`.

⁵Sur le tas.

Remarque V.1.6. Allouer dynamiquement la mémoire nécessite aussi de rendre la mémoire au système d'exploitation lorsque le programme n'a plus besoin d'une zone mémoire affectée précédemment. Pour se faire, on utilise l'instruction `free`. Ainsi, dans l'exemple précédent, on devrait ajouter en fin de programme la ligne suivante.

```
free(p);
```

1.3. Structures. On peut définir des variables plus complexes en utilisant la notion de structure. Cela permet de regrouper un ensemble de variables en une seule et même variable.

Exemple V.1.7. Par cet exemple, nous rappelons comment définir une structure et comment accéder aux différents champs qui la constituent. Ici, nous déclarons simplement une variable et initialisons les deux champs correspondant.

```
struct couple {
    int x;
    int y;
};

main()
{
    struct couple a;
    a.x=3;
    a.y=5;
}
```

En utilisant l'instruction `typedef`, nous pouvons définir des synonymes. Cela permet d'alléger quelque peu le code (par exemple, on ne doit plus réécrire `struct` dans les déclarations de variables et les en-têtes de fonctions). Ainsi, le programme précédent peut se réécrire comme suit.

```
struct couple {
    int x;
    int y;
};

typedef struct couple PAIRE;

main()
{
    PAIRE a;
    a.x=3;
    a.y=5;
}
```

Nous pouvons bien évidemment considérer un pointeur vers une structure. Avec la définition de `PAIRE` donnée dans l'exemple précédent, on a par exemple les affectations licites suivantes.

```
main()
```

```
{
    PAIRE *a;
    (*a).x=3;
    a->y=5;
}
```

Ici, `a` est un pointeur vers une `PAIRE`. Ainsi, il est normal d'accéder au premier champ de la paire avec `(*a).x`. Enfin, nous en profitons pour rappeler que `a->y` est un synonyme de `(*a).y`.

1.4. Listes chaînées. Nous allons présenter cette construction à l'aide d'un exemple. Imaginons vouloir stocker les éléments d'une liste de taille n d'entiers. Si n varie, il n'est pas possible de prévoir à l'avance la taille d'un tableau devant recevoir les n entiers. Ainsi, un tel tableau doit être créé dynamiquement, i.e., au moment où le programme est en cours d'exécution. Nous définissons une structure.

```
struct elt {
    int x;
    struct elt *suivant;
};
```

Celle-ci est constituée d'un premier champ entier `x` et d'un second champ `suivant` qui est un pointeur vers une structure du même type. Pour ne pas alourdir l'exposé, notre exemple ne va pas inclure d'interaction avec un utilisateur devant introduire une liste de nombres. Nous construisons simplement une liste chaînée contenant 6 éléments à l'aide de deux pointeurs.

Remarque V.1.8. Nous en profitons également pour vous rappeler que le langage C est "*case sensitive*", i.e., la distinction est faite entre minuscules et majuscules. Ainsi, "`elt`" diffère de "`ELT`" (ou de "`Elt`").

```
#include <stdio.h>
struct elt{
    int x;
    struct elt *suivant;
};
typedef struct elt ELT;

main()
{
    int i;
    ELT *p,*q;

    /* construction de la liste chainee ----- */

    q=(ELT *) malloc(sizeof(ELT));
    q->x=0;
    q->suivant=NULL;

    for (i=1; i<=5; i++)
```

```

{
    p=(ELT *) malloc(sizeof(ELT));
    p->x=i;
    p->suitant=q;
    q=p;
}

/* parcours de la liste chainee ----- */

while (p->suitant !=NULL)
{
    printf("> %d -- adresse %u \n",p->x,p);
    p=p->suitant;
}
printf("> %d -- adresse %u \n",p->x,p);

/* libere la memoire ----- */

p=q;
while (p->suitant !=NULL)
{
    q=p;
    p=p->suitant;
    free(q);
}
free(p);
}

```

Schématiquement, la création de la liste chaînée se passe comme indiqué à la figure V.3.

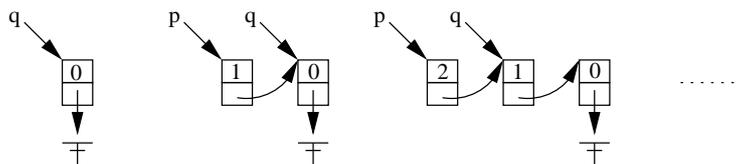


FIGURE V.3. Création d'une liste chaînée.

Le programme affiche un résultat semblable à :

```

> 5 -- adresse 134520920
> 4 -- adresse 134520904
> 3 -- adresse 134520888
> 2 -- adresse 134520872
> 1 -- adresse 134520856
> 0 -- adresse 134520840

```

Pour connaître le nombre d'octets nécessaires pour stocker un type de données sur une architecture donnée, on peut utiliser une instruction comme

```
printf("%d \n",sizeof(int));
printf("%d \n",sizeof(ELT));
```

2. Liste d'adjacence

Dans ce chapitre, on va principalement considérer l'implémentation des graphes orientés (i.e., avec boucle et sans multi-arcs). Le cas des multigraphes se traite de manière sensiblement analogue, le cas échéant, nous attirerons l'attention du lecteur sur les éventuelles différences apparaissant dans le traitement des multigraphes. Bien sûr, les graphes non orientés peuvent être vus comme un cas particulier de graphes orientés symétriques. (On pourrait dès lors présenter une structure de données plus fine nécessitant un espace mémoire moins important, ici la structure de données générales code deux fois trop d'information).

Nous supposons également disposer de deux fonctions de poids, l'une pour les arcs, l'autre pour les sommets. Ainsi, si $G = (V, E)$ est un graphe, alors on dispose des fonctions

$$f : V \rightarrow \mathbb{N} \quad \text{et} \quad g : E \rightarrow \mathbb{N}.$$

Un moyen commode de représenter un graphe est de considérer des listes d'adjacence. Cette structure de données est particulièrement bien adaptée aux graphes “**peu denses**”⁶ (c'est-à-dire comportant peu d'arcs par sommet). En effet, si un graphe comporte de nombreux arcs par sommet, il est plus efficace de coder le graphe par une matrice (on évite alors la manipulation plus lourde des listes chaînées).

Une *représentation par liste d'adjacence* est une liste chaînée de structures de liste d'adjacence. Ces dernières sont formées de deux champs : un sommet et une liste chaînée des sommets adjacents à celui-ci. Trois structures sont utilisées pour coder un graphe : `ELTADJ`, `SOMMET` et `GRAPHE`. La figure V.4 reprend schématiquement un graphe et son codage.

La structure la plus fondamentale permet de lister les sommets adjacents à un sommet donné. Autrement dit, elle sert à construire une liste d'adjacence des arcs issus d'un sommet donné. Le champ `dest` contient l'extrémité de l'arc, `info` contient un entier pouvant recevoir une quelconque information (un poids par exemple) à propos de l'arc codé et enfin, `suivant` est un pointeur vers l'élément suivant de la liste chaînée. La figure V.5 reprend schématiquement une liste d'adjacence.

Ensuite, la structure suivante permet de définir une liste chaînée destinée à coder les sommets du graphe. Chaque sommet possède un label (un indice) ainsi qu'une information qui lui est propre (par exemple, une couleur ou une valeur permettant de retenir si le sommet a déjà été visité). Les deux derniers champs de la structure sont plus spécifiques, `suivant` pointe simplement vers l'élément suivant de la liste chaînée. Enfin, `adj` est un pointeur vers une variable de type `eltadj`. Cela permet donc de lier un sommet à la liste

⁶En anglais, “*sparse graphs*”.

C'est un choix : simplifier la présentation ! De même, la structure d'arbre ne sera pas envisagée en tant que telle.

Gardez ce fait à l'esprit !

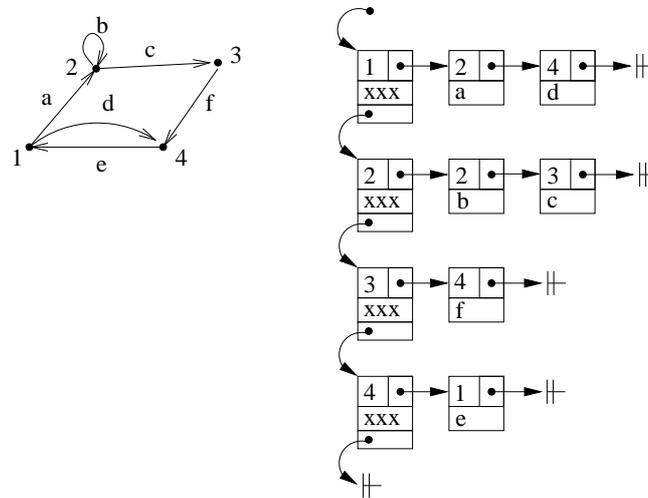


FIGURE V.4. Implémentation d'un graphe.

```

struct eltadj
{
    int dest;
    int info;
    struct eltadj *suivant;
};

```

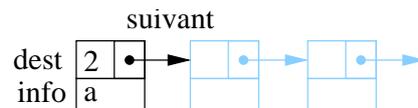


FIGURE V.5. Le type eltadj.

des sommets qui lui sont adjacents. La figure V.6 reprend schématiquement une liste chaînée de sommets.

```

struct sommet {
    int label;
    int info;
    struct sommet *suivant;
    struct eltadj *adj;
};

```

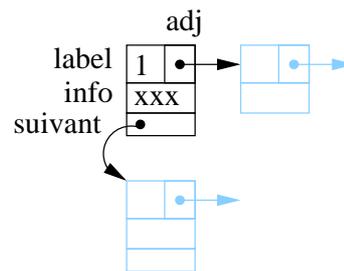


FIGURE V.6. Le type sommet.

Enfin, la structure graphe est principalement un pointeur vers un sommet (`premierSommet`). Les autres champs de la structure permettent de stocker quelques informations utiles comme le nombre de sommets ou d'arcs du graphe ainsi qu'un pointeur vers le dernier sommet de la liste chaînée. Dans de nombreuses procédures, les sommets reçoivent un indice. La variable `maxS` stocke le plus grand indice qui a été attribué à un sommet du graphe. Ainsi, en attribuant à un nouveau sommet l'indice `maxS+1`, on est

certain que cet indice n'est pas déjà employé. La figure V.7 reprend schématiquement la structure de graphe.

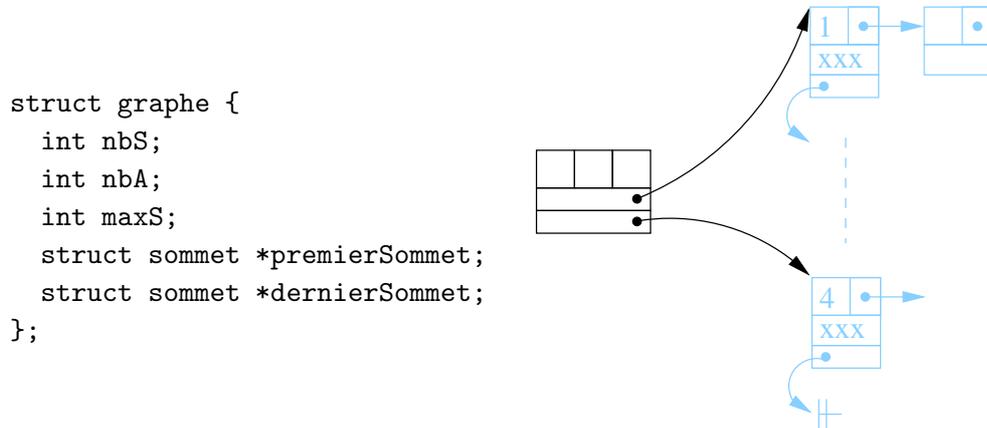


FIGURE V.7. Le type graphe.

2.1. Une manipulation simplifiée. Nous mettons à la disposition du lecteur une série de fonctions préprogrammées pour faciliter la manipulation des graphes et se concentrer principalement sur les algorithmes relevant de ce cours.

.....

- `void initialiserGraphe(GRAPHE *)`;

On transmet à cette fonction l'adresse d'une variable de type GRAPHE. La variable correspondante sera initialisée au graphe vide.

.....

- `int ajouterSommet(GRAPHE *, int info)`;

On transmet à cette fonction l'adresse d'une variable de type GRAPHE et un entier. Le champ `maxS` est incrémenté d'une unité, un sommet de label `maxS` (après incrémentation) est ajouté au graphe en fin de liste et ce sommet porte l'information `info` fournie en argument.

Erreur	-1	si l'allocation de mémoire n'est pas possible
--------	----	---

.....

- `int ajouterArc(GRAPHE *, int a, int b, int info)`;

Crée un nouvel arc du sommet de label `a` vers le sommet de label `b` avec l'information supplémentaire donnée par `info`. Les arcs sont rangés dans la liste d'adjacence par indice croissant. Si un arc entre `a` et `b` existe déjà, le champ `info` est mis à jour sans créer de nouvel arc dans la liste d'adjacence. Cette fonction ne renvoie rien.

Erreur	-1	si <code>a</code> n'existe pas
	-2	si <code>b</code> n'existe pas
	-3	si l'allocation de mémoire n'est pas possible

.....

- `int supprimerSommet(GRAPHE *, int label);`
 Supprime un sommet et les arcs correspondants (i.e., les arcs ayant leur origine ou leur extrémité correspondant à `label`). Cette fonction libère la mémoire correspondante.

Erreur	-1	si le sommet n'existe pas
--------	----	---------------------------

.....

- `int supprimerArc(GRAPHE *, int a, int b);`
 Supprime l'arc joignant `a` à `b`. Cette fonction libère la mémoire correspondante.

Erreur	-1	si l'arc n'existe pas
--------	----	-----------------------

.....

- `void supprimerGraphe(GRAPHE *);`
 Supprime entièrement un graphe en libérant la mémoire correspondante. En fin de procédure, le graphe est automatiquement réinitialisé par la fonction `initialiserGraphe`.

.....

- `void afficherGraphe(GRAPHE *);`
 Il s'agit exclusivement d'une fonction d'affichage. Elle affiche à l'écran la valeur des champs `nbS`, `nbA` et `maxS` ainsi que l'ensemble des sommets, des arcs et des informations qu'ils portent.

.....

- `int lireFichier(char *nomf, GRAPHE *);`
 On fournit à cette fonction le nom d'un fichier de données et un pointeur vers une variable de type `GRAPHE`. Ce fichier doit contenir des champs séparés par des virgules (et peut contenir arbitrairement des tabulations et des espaces, ceux-ci ne sont pas pris en compte). Un fichier valide contient des champs formés de nombres ou du seul caractère `x`. Chaque ligne (terminée par un retour chariot) doit contenir le même nombre de champs (les lignes vides sont ignorées). Le k -ième champ de la j -ième ligne du fichier contient le poids de l'arc entre les sommets j et k d'un graphe (on se limite ici à des poids positifs ou nuls). Si ce champ est `x`, il n'y a pas d'arcs entre les sommets j et k . Il est inutile d'initialiser le graphe avant d'appeler cette fonction. En effet, un tel appel est effectué en début de fonction.

Erreur	-1	si le fichier n'est pas valide
--------	----	--------------------------------

Exemple V.2.1. Voici un fichier codant le graphe donné à la figure V.4. Ici, les labels a, b, \dots, f ont été remplacés par les entiers de 1 à 6.

```

x,1,x,4
x,2,3,x
x,x,x,6
5,x,x,x

```

Remarque V.2.2. L'utilisation d'un tableur peut s'avérer commode pour l'encodage d'un graphe dans un fichier. En effet, après avoir placé les divers éléments dans les cases d'un tableau, le tableur propose une option du type "Save as" dans laquelle on peut spécifier que le type du fichier en sortie sera un (simple) texte. Le tableur demande alors à l'utilisateur de préciser un délimiteur pour les champs (il convient de choisir la virgule) et un délimiteur pour le texte (il convient de ne prendre aucun délimiteur).

2.2. Utilisation des fonctions.

Exemple V.2.3. Voici deux variantes de programmes permettant d'encoder le graphe de la figure V.4.

```

/* FICHIER main.c ----- */
#include <stdio.h>
#include "graphes.h"
int main()
{
    GRAPHE g;
    int i;
    initialiserGraphe(&g);
    for (i=1; i<=4; i++) ajouterSommet(&g,0);
    ajouterArc(&g,1,2,1);
    ajouterArc(&g,1,4,4);
    ajouterArc(&g,2,2,2);
    ajouterArc(&g,2,3,3);
    ajouterArc(&g,3,4,6);
    ajouterArc(&g,4,1,5);
    afficherGraphe(&g);
    supprimerGraphe(&g);
}

```

Dans ce second fichier, on utilise un fichier de données externes comme celui donné dans l'exemple V.2.1.

```

/* FICHIER main.c ----- */

#include <stdio.h>
#include "graphes.h"
int main()
{
    GRAPHE g;
    int i;
    lireFichier("data.gr",&g);
    afficherGraphe(&g);
    supprimerGraphe(&g);
}

```

```
}

```

Ainsi, pour compiler le fichier et obtenir un fichier exécutable, vous avez besoin dans le répertoire courant non seulement du fichier source `main.c` mais aussi du fichier `graphes.c` et du fichier d'en-tête `graphes.c`. La compilation peut être lancée par la commande suivante⁷.

```
> gcc graphes.c main.c -o nom_du_programme

```

Bien évidemment, il vous faut également un fichier de données nommé `data.gr` pour utiliser la seconde variante.

2.3. Détail de l'implémentation. Le fichier d'en-tête est on ne peut plus classique. Il sert uniquement à la déclaration des différents types de données, des fonctions et d'une constante `MAX` utilisée pour la lecture de fichiers.

```
/* graphes.h ----- */
#ifdef GRAPHER_H
#else
#define GRAPHER_H

#define MAX 10000

struct eltadj {
    int dest;
    int info;
    struct eltadj *suivant;
};
struct sommet {
    int label;
    int info;
    struct sommet *suivant;
    struct eltadj *adj;
};
struct graphe {
    int nbS;
    int nbA;
    int maxS;
    struct sommet *premierSommet;
    struct sommet *dernierSommet;
};

typedef struct graphe GRAPHE;
typedef struct sommet SOMMET;
typedef struct eltadj ELTADJ;

void initialiserGraphe(GRAPHE *);
int ajouterSommet(GRAPHE *, int info);

```

⁷Ou une instruction similaire suivant le système employé.

```

int ajouterArc(GRAPHE *, int a, int b, int info);
int supprimerSommet(GRAPHE *, int label);
int supprimerArc(GRAPHE *, int a, int b);
void supprimerGraphe(GRAPHE *);
void afficherGraphe(GRAPHE *);
int lireFichier(char *nomf, GRAPHE *);

#endif

```

Nous allons à présent détailler l'implémentation des fonctions du module `graphes.c` permettant la gestion de la structure de données de type `graphe`.

- La fonction `initialiserGraphe` est très simple, il s'agit uniquement d'affectations élémentaires.

```

void initialiserGraphe(GRAPHE *g)
{
    g->nbS=0;
    g->nbA=0;
    g->maxS=0;
    g->premierSommet=NULL;
    g->dernierSommet=NULL;
}

```

- La fonction `ajouterSommet` ajoute un élément de type `sommet` comme dernier élément de la liste chaînée contenant les sommets du graphe d'adresse `*g`. Le champ `maxS` de `g` (contenant le plus grand label attribué en cours d'exécution à un sommet du graphe) est incrémenté d'une unité et cette valeur est aussi le label du sommet à ajouter. Il est de plus nécessaire de mettre à jour les champs `dernierSommet`, `nbS` et éventuellement `premierSommet` (s'il s'agit du premier sommet créé dans le graphe) de `g`. On notera l'intérêt d'avoir à sa disposition le champ `dernierSommet` permettant un accès immédiat à l'extrémité de la liste chaînée.

```

int ajouterSommet(GRAPHE *g, int info)
{
    SOMMET *pointeur;
    g->maxS++;
    pointeur=(SOMMET *) malloc(sizeof(SOMMET));
    if (pointeur == NULL)
    {
        printf("Erreur! Memoire insuffisante pour creer un sommet\n");
        return -1;
    }
    else
    {
        pointeur->label=g->maxS;
        pointeur->info=info;
        pointeur->suivant=NULL;
        pointeur->adj=NULL;
        if (g->nbS == 0)
        {
            g->premierSommet=pointeur;
            g->dernierSommet=pointeur;
        }
        else
        {
            g->dernierSommet->suivant=pointeur;
            g->dernierSommet=pointeur;
        }
    }
}

```

```

    }
    g->nbS++;
    return pointeur->label;
}
}

```

.....

- La fonction `ajouterArc` détermine tout d'abord si le graphe contient bien les sommets de label donnés par `a` et par `b`. Si l'un d'eux n'existe pas, la fonction renvoie un code d'erreur égal respectivement à `-1` ou `-2`. Sinon, deux situations peuvent se présenter. L'arc existe, le champ `info` est simplement mis à jour. L'arc n'existe pas. Il est créé au sein de la liste chaînée des sommets adjacents au sommet de label `a`. Il est positionné dans cette liste de telle sorte que les labels des sommets présents sont ordonnés par ordre croissant.

```

int ajouterArc(GRAPHE *g, int a, int b, int info)
{
    SOMMET *psommet, *psommet2;
    ELTADJ *padj, *precedent;
    psommet=g->premierSommet;

    /* on parcourt les sommets jusqu'a trouver a */
    while (psommet != NULL)
    {
        if (psommet->label == a) break;
        psommet=psommet->suivant;
    }
    if (psommet == NULL)
    {
        printf("Erreur! Creation d'un arc dont l'origine n'existe pas\n");
        return -1;
    }
    else /* on a trouver a */
    {
        padj=psommet->adj;

        /* on parcourt les sommets pour trouver b */
        psommet2=g->premierSommet;
        while (psommet2 != NULL)
        {
            if (psommet2->label == b) break;
            psommet2=psommet2->suivant;
        }
        if (psommet2 == NULL)
        {
            printf("Erreur! Creation d'un arc dont l'extremite n'existe pas\n");
            return -2;
        }
        else /* on a trouver a et b */
        {
            if (padj == NULL) /* la liste d'adjacence est vide */
            {
                padj=(ELTADJ *) malloc(sizeof(ELTADJ));
                if (padj == NULL)
                {
                    printf("Erreur! Memoire insuffisante pour creer un sommet\n");
                    return -3;
                }
            }
            else
            {
                psommet->adj=padj; /* premier element de la liste d'adjacence */
                padj->suivant=NULL;
            }
        }
    }
    else /* la liste d'adjacence est non vide, on la parcourt pour voir si b s'y trouve */

```

```

{
  if (padj->dest > b)
  {
    padj=(ELTADJ *) malloc(sizeof(ELTADJ));
    if (padj == NULL)
    {
      printf("Erreur! Memoire insuffisante pour creer un sommet\n");
      return -3;
    }
    else
    {
      padj->suivant=psommet->adj;
      psommet->adj=padj;
    }
  }
  else
  {
    while (padj != NULL)
    {
      if (padj->dest == b) {padj->info=info; break;} /* l'arc existe, update info */
      if (padj->dest > b) {padj=NULL; break;} /* on depasse b sans le trouver */
      precedent=padj;
      padj=padj->suivant;
    }
    if (padj == NULL) /* l'arc n'existe pas, il faut le creer */
    {
      padj=(ELTADJ *) malloc(sizeof(ELTADJ));
      if (padj == NULL)
      {
        printf("Erreur! Memoire insuffisante pour creer un sommet\n");
        return -3;
      }
      else
      if (precedent->suivant==NULL) /* element ajouter a la fin */
      {
        precedent->suivant=padj;
        padj->suivant=NULL;
      }
      else /* element ajouter "au milieu" pour garder ordre */
      {
        padj->suivant=precedent->suivant;
        precedent->suivant=padj;
      }
    }
  }
  }
  padj->dest=b;
  padj->info=info;
  g->nbA++;
}
return 0;
}
}

```

-
- La fonction `supprimerSommet` teste si le sommet de label `a` existe. Si tel est le cas, ce sommet est supprimé de la liste chaînée constituée des sommets du graphe (ce qui nécessite la mise à jour de pointeurs de cette liste). On supprime aussi la liste chaînée des sommets adjacents à `a` et les occurrences du sommet `a` dans les liste d'adjacence des autres sommets (avec les mises à jour ad hoc des listes chaînées).

```

int supprimerSommet(GRAPHE *g, int a)
{
  SOMMET *psommet, *precedent;
  ELTADJ *padj, *suivant, *precedent_adj;
  int flag_premier_sommet, flag_premier_arc;
  if (g->premierSommet == NULL)

```

```

{
    printf("Erreur! Graphe vide, suppression impossible\n");
    return -1;
}
else
{
    psommet=g->premierSommet;
    flag_premier_sommet=1;
    while (psommet != NULL)
    {
        if (psommet->label == a) break;
        else
        {
            flag_premier_sommet=0;
            precedent=psommet;
            psommet=psommet->suisvant;
        }
    }
    if (psommet == NULL)
    {
        printf("Erreur! Le sommet a supprimer n'existe pas\n");
        return -1;
    }
    else
    {
        if (psommet->suisvant == NULL) g->dernierSommet=precedent;

        if (flag_premier_sommet == 1) g->premierSommet=psommet->suisvant;
        else precedent->suisvant=psommet->suisvant;
        padj=psommet->adj;
        free(psommet);
        g->nbS--;
        while (padj != NULL)
        {
            suivant=padj->suisvant;
            free(padj);
            g->nbA--;
            padj=suivant;
        }
    }

    /* il faut aussi supprimer les arcs ayant le sommet a supprimer comme extremite */
    psommet=g->premierSommet;
    while (psommet != NULL)
    {
        padj=psommet->adj;
        while (padj !=NULL)
        {
            flag_premier_arc=1;
            if (padj->dest == a) break;
            else
            {
                flag_premier_arc=0;
                precedent_adj=padj;
                padj=padj->suisvant;
            }
        }
        if (padj != NULL)
        {
            if (flag_premier_arc == 1) psommet->adj=padj->suisvant;
            else precedent_adj->suisvant=padj->suisvant;
            free(padj);
            g->nbA--;
        }
        psommet=psommet->suisvant;
    }
    return 0;
}
}
}

```

.....

- La fonction `supprimerArc` supprime, si possible, de la liste d’adjacence du sommet de label `a` l’arc joignant les sommets de label `a` et `b`. Une erreur est renvoyée sinon.

```

int supprimerArc(GRAPHE *g, int a, int b)
{
    SOMMET *psommet;
    ELTADJ *padj, *precedent_adj;
    int flag_premier_arc;
    if (g->premierSommet == NULL)
    {
        printf("Erreur! Graphe vide, suppression impossible\n");
        return -1;
    }
    else
    {
        psommet=g->premierSommet;
        while (psommet != NULL)
        {
            if (psommet->label == a) break;
            else psommet=psommet->suivant;
        }
        if (psommet == NULL)
        {
            printf("Erreur! L'extremite de l'arc a supprimer n'existe pas !!!!\n");
            return -1;
        }
        else
        {
            padj=psommet->adj;
            flag_premier_arc=1;
            while (padj !=NULL)
            {
                if (padj->dest == b) break;
                else
                {
                    flag_premier_arc=0;
                    precedent_adj=padj;
                    padj=padj->suivant;
                }
            }
            if (padj != NULL)
            {
                if (flag_premier_arc == 1) psommet->adj=padj->suivant;
                else precedent_adj->suivant=padj->suivant;
                free(padj);
                g->nbA--;
            }
            else
            {
                printf("Erreur! L'extremite de l'arc a supprimer n'existe pas !!!!\n");
                return -1;
            }
        }
    }
    return 0;
}
}

```

-
- La fonction `supprimerGraphe` libère l’ensemble de la mémoire utilisée par un graphe et rend simplement un graphe “vide”.

```

void supprimerGraphe(GRAPHE *g)
{
    SOMMET *psommet,*temps;
    ELTADJ *padj,*tempadj;
    psommet=g->premierSommet;
    while (psommet !=NULL)
    {
        padj=psommet->adj;

```

```

    while (padj !=NULL)
    {
        tempadj=padj;
        padj=padj->suisvant;
        free(tempadj);
    }
    temps=psommet;
    psommet=psommet->suisvant;
    free(temps);
}
initialiserGraphe(g);
}

```

-
- La fonction `afficherGraphe` affiche tout d'abord plusieurs informations de base sur le graphe d'adresse `*g` (nombre de sommets, nombre d'arêtes, ...). Ensuite la liste chaînée des sommets est parcourue et pour chacun de ces sommets, on parcourt l'entièreté de la liste d'adjacence correspondante.

```

void afficherGraphe(GRAPHE *g)
{
    SOMMET *psommet;
    ELTADJ *padj;
    if (g->premierSommet == NULL) printf("graphe vide\n");
    else
    {
        printf("nbS=%d , nbA=%d, label max.=%d\n", g->nbS, g->nbA, g->maxS);
        psommet=g->premierSommet;
        do
        {
            printf("\n");
            printf("Sommet de label: %d , info: %d\n", psommet->label, psommet->info);
            if (psommet->adj == NULL) printf(" -> ce sommet n'a aucun arc sortant\n ");
            else
            {
                padj=psommet->adj;
                do
                {
                    printf(" -> arc de %d vers %d avec l'info. %d \n", psommet->label, padj->dest, padj->info);
                    padj=padj->suisvant;
                }
                while (padj != NULL);
            }
            printf("\n");
            psommet=psommet->suisvant;
        }
        while (psommet != NULL);
    }
}

```

-
- La fonction `lireFichier` ouvre en lecture un fichier texte. Ce fichier, s'il satisfait les conventions énoncées précédemment (cf. exemple V.2.1), est utilisé pour construire le graphe correspondant.

```

int lireFichier(char *nomf, GRAPHE *g)
{
    FILE *fp;
    char ligne[MAX+1];
    int temp,i,j,nbS1,nbLigne,sommet,nbElt,creerArc;

    initialiserGraphe(g);
    fp=fopen(nomf,"r"); /* ouvre un fichier en lecture */
    nbLigne=0; /* compte les lignes du fichier */
    sommet=0; /* label du sommet en cours */
    nbS1=0; /* compte les sommets de la 1ere ligne */
    while (fgets(ligne,MAX,fp) != NULL)

```

```

{
    nbLigne++; /* compte le nombre de lignes du fichier */

    if (ligne[0] != '\n') /* on passe les lignes vides */
    {
        i=0;
        if (nbS1 == 0) /* compte les sommets de la 1ere ligne */
        {
            nbS1=1;
            while (ligne[i] != '\n')
            {
                if (ligne[i] == ',') nbS1++;
                i++;
            }
            for (j=1; j<=nbS1; j++)
            {
                ajouterSommet(g,0);
            }
            i=0; /* on relit la 1ere ligne */
        }

        sommet++; /* origine des arcs */
        nbElt=0; /* controle le nombre d'arcs crees */
        while (ligne[i] != '\n')
        {
            temp=0; /* va contenir l'extremite */
            creerArc=1;
            while (ligne[i] != ',' && ligne[i] != '\n')
            {
                while (ligne[i]== ' ' || ligne[i]=='\t') {i++;}
                if ((ligne[i]>'9' || ligne[i]<'0') && ligne[i]!='x')
                {
                    printf("Erreur à la ligne %d !\n",nbLigne);
                    supprimerGraphe(g);
                    return -1; /* pas des chiffres ! */
                }
                if (ligne[i]=='x') creerArc=0;
                temp=10*temp+ligne[i]-'0';
                i++;
                while (ligne[i]== ' ' || ligne[i]=='\t') {i++;}
            }
            if (ligne[i] == ',') i++;
            nbElt++;
            if (nbElt<=nbS1 && creerArc==1) ajouterArc(g,sommet,nbElt,temp); /* ligne pas trop longue */
        }
        if (nbElt != nbS1) /* pas le bon nombre de champs sur ligne */
        {
            printf("Erreur à la ligne %d !\n",nbLigne);
            supprimerGraphe(g);
            return -1; /* pas le bon nombre de champs */
        }
    }
}
fclose(fp);
return 0;
}

```

Bibliographie

- [1] J.-P. Allouche, J. Shallit, *Automatic sequences, Theory, Applications, Generalizations*, Cambridge Univ. Press, Cambridge (2004).
- [2] C. Berge, *Graphes et hypergraphes*, Dunod, Paris, (1970).
- [3] N. Biggs, *Algebraic Graph Theory*, 2nd Edition, Cambridge Math. Library, (1993).
- [4] B. Bollobás, *Graph Theory, An Introductory Course*, Graduate Text in Math. **63**, Springer, (1979).
- [5] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, North-Holland, New-York, (1976).
- [6] V. Bouchitté, *Graphes et algorithmique des graphes*, notes de cours, École normale supérieure de Lyon.
- [7] C.P. Bruter, *Les matroïdes, nouvel outil mathématique*, Initiation aux nouveautés de la science **21**, Dunod, Paris, (1970).
- [8] R. Diestel, *Graph Theory*, 3rd Edition, Graduate Text in Math. **173**, Springer, (2005).
- [9] F.R. Gantmacher, *The theory of matrices*, Chelsea, (1960).
- [10] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, Cambridge University Press, (1994).
- [11] C. Godsil, G. Royle, *Algebraic Graph Theory*, Graduate Text in Math. **207**, Springer, (2001).
- [12] T. Harju, *Lecture Notes on Graph Theory*, University of Turku, Spring 2002.
- [13] B.W. Kernighan, D.M. Ritchie, *The C Programming Language*, 2nd Ed., Prentice Hall, (1988).
- [14] D. Lind, B. Marcus, *Symbolic Dynamics and Coding, An Introduction*, Cambridge University Press, (1995).
- [15] P. Lopez, Cours de **GRAPHES**, notes de cours (2005).
www.laas.fr/~lopez/cours/GRAPHES/graphes.html.
- [16] K. Loudon, *Maîtrise des algorithmes en C*, Ed. française, O'Reilly, Paris, (2000)
- [17] C. Meyer, A. Langville, Updating the PageRank Vector, SIAM Annual Meeting, New Orleans, 2005.
- [18] C. Meyer, A. Langville, Google's PageRank and Beyond: The Science of Search Engine Rankings, Princeton University Press, (2006).
- [19] Ministère de l'éducation nationale, *Introduction d'éléments de la théorie des graphes*, accompagnent de la mise en oeuvre des programmes, (2001).
- [20] O. Ore, *Graphs and their uses*, version révisée par R. J. Wilson, New Mathematical Library **34**, Math. Association of America.
- [21] J. Oxley, On the interplay between graphs and matroids, Surveys in combinatorics, 2001 (Sussex), 199–239, *London Math. Soc. Lecture Note Ser.* **288**, Cambridge Univ. Press, (2001).
- [22] L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report, Stanford University, 1998.
- [23] J. Miles Prystowsky, L. Gill, *Calculating Web Page Authority Using the PageRank Algorithm*, <http://online.redwoods.cc.ca.us/instruct/darnold/LAPROJ/>.
- [24] S. P. Radziszowski, Small Ramsey Number, Dynamic surveys, *The Electronic Journal of Combinatorics*, <http://www.combinatorics.org/Surveys/ds1.pdf>

- [25] V. Rosta, *Ramsey Theory Applications*, Dynamic surveys, *The Electronic Journal of Combinatorics*, www.combinatorics.org/Surveys/ds13.pdf
- [26] E. Seneta, *Non-Negative Matrices, An Introduction to Theory and Applications*, George Allen and Unwin Ltd, London, (1973).
- [27] W.T. Tutte, *Graph Theory*, Cambridge Mathematical Library, Encyclopedia of mathematics and its applications **21**, (1984).
- [28] D.J.A. Welsh, *Matroid Theory*, Academic Press, London, (1976).
- [29] R. Wilson, Four Colours Suffice, or how to colour a map, *European Math. Society Newsletter* **46** December 2002, 15–19.

Liste des figures

I.1 Un arc reliant deux sommets, une boucle.	6
I.2 Un exemple de graphe.	6
I.3 Un exemple de multi-graphe.	7
I.4 Un exemple de graphe simple.	8
I.5 Un graphe (dirigé) simple, un graphe et un multi-graphe.	9
I.6 Un graphe (non dirigé) simple, un graphe et un multi-graphe.	9
I.7 Des graphes non orientés 3 et 4-réguliers.	9
I.8 Un multi-graphe orienté 2-régulier.	10
I.9 Un graphe biparti (non complet).	10
I.10 Graphe étiqueté par les distances entre villes (itinéraire “express” calculé par www.mappy.fr).	11
I.11 Un exemple d’hyper-graphe.	11
I.12 Les sept ponts de Königsberg.	13
I.13 Deux représentations d’un cube.	13
I.14 La carte des USA.	14
I.15 Graphe d’incompatibilité.	14
I.16 Graphe de Cayley de \mathcal{S}_3 .	15
I.17 Un autre graphe de Cayley de \mathcal{S}_3 .	16
I.18 Sous-graphe couvrant.	16
I.19 Orientation d’un graphe.	17
I.20 Un graphe planaire.	18
I.21 Un graphe non planaire.	18
I.22 $K_{3,3}$ sur un tore.	18
I.23 Graphe d’intervalles.	19
I.24 Problème d’affectation.	19
I.25 Une application du tri topologique.	20
I.26 Le graphe de De Bruijn d’ordre 3 de $ababab\dots$.	20
I.27 Un problème de flot maximum.	21
I.28 Chemin critique dans la planification de tâches.	22
I.29 Une piste (ou chemin élémentaire) et un chemin simple	23

I.30	Un circuit, un circuit élémentaire (ou piste fermée) et un circuit simple.	23
I.31	Un graphe orienté.	25
I.32	Un digraphe simple pondéré.	26
I.33	Illustration de l'algorithme de Dijkstra.	28
I.34	Illustration de l'algorithme de Dijkstra.	28
I.35	Construction d'un circuit eulérien.	30
I.36	Une maison à tracer d'un seul trait.	31
I.37	Un graphe simple (connexe).	33
I.38	digraphe et graphe des composantes.	35
I.39	Un graphe et deux sous-graphes.	37
I.40	Des sommets indépendants.	38
I.41	Un graphe et ses points d'articulation.	39
I.42	Un graphe au moins 2-connexe.	39
I.43	Un graphe 2-connexe.	39
I.44	Un graphe et ses composantes 2-connexes.	40
I.45	Un graphe et ses arêtes de coupure.	40
I.46	Illustration de la proposition I.6.6	41
I.47	Mise à sens unique	42
I.48	Un graphe et une coupure.	42
I.49	Un multi-graphe tel que $\lambda(G) = 2$ et $\kappa(G) = 1$.	42
I.50	Un multi-graphe tel que $\lambda(G) = k$ et $\kappa(G) = 1$.	43
I.51	Un graphe simple tel que $\lambda(G) = 4$ et $\kappa(G) = 1$.	43
I.52	Une clique de taille 4, $\omega(G) = 4$.	44
I.53	Une illustration du théorème de Menger : 3 chemins indépendants joignent u et v , 3 sommets séparent u et v .	45
I.54	Une illustration de la preuve corollaire de Menger.	46
I.55	Un arbre.	50
I.56	Un arbre pointé.	51
I.57	Un arbre à parcourir.	52
I.58	Exemple de parcours en profondeur d'un graphe.	53
I.59	Homomorphisme de G dans H .	53
I.60	Homomorphisme de G dans H .	53
I.61	Deux graphes isomorphes.	54
I.62	Un arbre lexicographique infini.	55
I.63	Un arbre infini pondéré représentant le dictionnaire de A_L .	56
I.64	Un arbre infini pondéré représentant le dictionnaire de A_M .	56

I.65	Squelette d'un dodécaèdre et circuit hamiltonien.	57
I.66	Déplacement d'un cavalier.	58
I.67	Une illustration de la preuve de la proposition I.11.3.	58
I.68	Un graphe non hamiltonien.	59
I.69	Le graphe de Petersen.	59
I.70	Théorème d'Ore, circuit dans G .	62
I.71	Construction de la fermeture d'un graphe.	63
I.72	Deux circuits hamiltoniens disjoints de K_5 .	66
I.73	Deux autres circuits hamiltoniens disjoints de K_5 .	66
I.74	Une tentative de partition de K_7 .	67
I.75	Chemins hamiltoniens disjoints de K_6 et K_8 .	68
II.1	Un graphe G et sa matrice d'adjacence.	70
II.2	Deux graphes cospectraux.	71
II.3	Disposition des valeurs propres d'une matrice irréductible.	75
II.4	Une illustration des graphes G et G' .	75
II.5	Disposition des valeurs propres d'une matrice primitive.	77
II.6	Un graphe avec une matrice d'adjacence primitive.	78
II.7	Un graphe avec une matrice d'adjacence irréductible.	78
II.8	Trois graphes f. connexes.	83
II.9	Deux composantes f. connexes.	86
II.10	Trois composantes f. connexes.	86
II.11	Recherche d'un sous-arbre couvrant.	96
II.12	Contraction d'un graphe G en $G \cdot e$.	97
II.13	Applications successives de la formule de Cayley.	97
II.14	Nombre de sous-arbres couvrants.	98
II.15	Un arbre encodé par $(1, 4, 3, 1, 6, 6)$.	99
II.16	Un arbre encodé par $(1, 1, 1)$.	99
II.17	Deux arbres de C_i pour lesquels $d_i = 2$ et $d_i = 3$.	100
II.18	Un sous-arbre couvrant pointé et orienté.	103
II.19	Graphes correspondant aux matrices $D_1^{(3)}, D_2^{(3)}, D_3^{(3)}$ et $D_4^{(3)}$.	105
II.20	Un arbre pointé parcouru en largeur (avec un arc supplémentaire).	106
II.21	Un cycle et des colonnes dont la somme est nulle.	106
II.22	Nombre de sous-arbres couvrants pour G et G' .	107
II.23	Les sous-arbres couvrants du graphe G .	108
II.24	Application de l'algorithme de Prim.	109
III.1	Théorème de Jordan	111

III.2	Un graphe planaire et ses faces.	112
III.3	Rotation sur la sphère	113
III.4	Choix de la face infinie.	113
III.5	Suppression d'arêtes.	114
III.6	Subdivision d'une arête.	115
III.7	Graphes homéomorphes.	115
III.8	Graphes homéomorphes.	116
III.9	Si G n'était pas 2-connexe.	117
III.10	H_1 , H_2 et l'arête $\{u, v\}$.	118
III.11	Lemme III.4.5	119
III.12	Si on dispose d'au plus une arête.	119
III.13	G contient un sous-graphe homéomorphe à $K_{3,3}$.	120
III.14	Illustration du lemme III.4.7.	120
III.15	Illustration du lemme III.4.7, cas (a).	121
III.16	Illustration du lemme III.4.7, cas (b) et (c).	121
IV.1	Une illustration du lemme IV.1.3.	123
IV.2	Si H n'est pas simple.	124
IV.3	Projection stéréographique.	125
IV.4	Un graphe et son dual.	125
IV.5	Francis Guthrie.	126
IV.6	Suppression des sommets de degré 2.	126
IV.7	Suppression des sommets de degré ≥ 4 .	127
IV.8	Réduction d'une face délimitée par deux arêtes.	128
IV.9	Réduction d'une face délimitée par deux arêtes.	129
IV.10	Configurations pour la réduction d'une face rectangulaire.	129
IV.11	Réductions d'une face rectangulaire.	129
IV.12	Satisfaction de R.2 .	130
IV.13	Réductions d'une face pentagonale.	130
IV.14	Cinq couleurs effectives.	131
IV.15	Une surface de genre 2.	131
IV.16	Illustration de $m_{k,G}$.	133
IV.17	$\mathcal{R}(3, 3) > 3, 4, 5$.	136
IV.18	Un graphe et un coloriage de K_n .	138
V.1	Un pointeur et une variable.	142
V.2	L'échange de deux variables.	144
V.3	Création d'une liste chaînée.	147

V.4 Implémentation d'un graphe.	149
V.5 Le type <code>eltadj</code> .	149
V.6 Le type <code>sommet</code> .	149
V.7 Le type <code>graphe</code> .	150

Index

Notations

(v_i, v_j) (arc)	5
$A(G)$ (matrice d'adjacence) ...	69, 72
$Aut(G)$ (groupe des automorphismes)	54
$D(G)$ (matrice de demi-degré entrant)	102
$G - e$ (sous-graphe)	37
$G - v$ (sous-graphe)	37
K_n (graphe complet)	9
$K_{m,n}$ (graphe biparti complet)	10
$\alpha(G)$ (sommets indépendants)	37
$\chi(G)$ (nombre chromatique)	123
$\kappa(H)$ (taille min. ens. articulation)	39
$\lambda(H)$ (taille min. coupe)	42
\leftrightarrow (sommets f. connectés)	24
\mathcal{A}_G (algèbre des polynômes de $A(G)$)	92
$\mathcal{C}_S(G)$ (graphe de Cayley)	15
$\mathcal{F}(G_0)$ (fermeture)	62
$\nu(v)$ (ens. des voisins)	8
$\omega(G)$ (clique maximale)	43
$\omega(v)$ (arcs adjacents)	6
$\omega^+(v)$ (arcs sortants)	6
$\omega^-(v)$ (arcs entrants)	6
\sim (sommets connectés)	23
$\tau(G)$ (nombre de sous-arbres couvrants)	96
$\{v_i, v_j\}$ (arête)	8
$d^+(v)$ (demi-degré sortant)	6
$d^-(v)$ (demi-degré entrant)	6
p -graphe	7
$\text{pred}(v)$ (prédécesseur)	6
$\text{succ}(v)$ (successeur)	6
$\text{diam}(G)$ (diamètre)	24

$d(a, b)$ (distance)	24
$\text{pred}^*(v)$	34
$\text{succ}^*(v)$	34

A

adjacent	
arc	6
sommets	8
algèbre d'adjacence	92
algorithme	
Dijkstra	26
Fleury	31
Kruskal	110
Prim	108
sous-arbre couvrant	95
test connexité	32
test cycle	47
test cycle (2)	48
ancêtre	51
Appel K.	131
arête	5
adjacente	8
incidente	8
arête	
coupure	40
subdivision	115
arbre	49
n -aire	49, 51
ancêtre	51
complet	51
descendant	51
feuille	51
fils	51
lexicographique	55
orienté	102
père	51
parcours	

- en profondeur 51
 - en largeur 52
 - infixe 52
 - préfixe 51
 - suffixe 51
- pointé 50
 - orienté 102
- régulier 56
- racine 50
- arc 5
 - adjacent 6
 - destination 5
 - entrant 6
 - extrémité 5
 - incident 6
 - origine 5
 - sortant 6
- articulation
 - ensemble 38
 - point 38
- B**
- Bott-Mayberry 107
- boucle 5
- C**
- Cayley 15, 98
- chemin 22, 24
 - élémentaire 22
 - eulérien 29
 - fermé 22
 - hamiltonien 57
 - ouvert 22
 - simple 22
- Chvátal (théorème) 64
- circuit 22
 - eulérien 29
 - hamiltonien 57
- Clarke 102
- clique 43
- collapse 35
- coloriage 123
 - propre 123
- component graph 35
- composante
 - connexe 23
 - f. connexe 24
 - fortement connexe 24
 - s. connexe 24
 - simplement connexe 24
- condensé 35
- contraction 96
- coupe 41
- coupure 41
- coupure (arête) 40
- cycle 22
- D**
- De Bruijn (graphe) 20
- degré 6, 8
- demi-degré
 - entrant 6
 - sortant 6
- descendant 51
- destination 5
- diamètre 24
- digraphe 8
- Dijkstra (algorithme) 26
- Dirac (théorème) 60
- distance 24
- E**
- edge 5
- ensemble
 - articulation 38
- Erdős 66, 137
- etiquette 10
- Euler
 - formule 113
- extrémité 5
- F**
- face 111
 - adjacente 112
 - infinie 112
- fermeture 62
- feuille 51
- fil 51
- Fleury
 - algorithme 31
- forêt 49
- formule
 - Euler 113
- frontière 112
- G**

- graphe..... 5
 2-connexe..... 38
 k -connexe..... 39
 k -connexe (pour les arêtes)..... 42
 étiqueté..... 10
 asymétrique..... 54
 biparti..... 10
 biparti complet..... 10
 Cayley..... 15
 comple..... 9
 connexe..... 23
 cospectraux..... 71
 De Bruijn..... 20
 des composantes..... 35
 diamètre..... 24
 dirigé..... 5, 8
 eulérien..... 29
 f. connexe..... 24
 fermeture..... 62
 fini..... 5
 fortement connexe..... 24
 hamiltonien..... 57
 isomorphe..... 54
 non dirigé..... 8
 non orienté..... 8
 orienté..... 5
 parcours en profondeur..... 52
 Petersen..... 59
 planaire..... 111
 planaire topologique..... 111
 pondéré..... 10
 primitif..... 74
 régulier..... 9
 s. connexe..... 24
 simple..... 7
 simplement connexe..... 24
 sous-graphe..... 37
 couvrant..... 38
 induit..... 37
 propre..... 37
 strict..... 7
 graphe d'intervalles..... 19
 Guthrie F..... 125
- H**
- Haken W..... 131
 handshaking lemma..... 8
 Heawood (formule)..... 131
- Hoffman (théorème)..... 94
 homéomorphisme..... 115
 homomorphisme..... 53
 hyper-arête..... 11
 hyper-graphe..... 11
 fini..... 11
- I**
- incident
 arête..... 8
 arc..... 6
 isomorphisme..... 54
- K**
- Kruskal (algorithme)..... 110
- L**
- label..... 10
 liste
 adjacence..... 148
- M**
- matrice
 acyclique..... 80
 adjacence..... 69, 72
 cyclique..... 80
 demi-degré entrant..... 102
 incidence..... 70
 irréductible..... 73
 primitive..... 73
 matroïde..... 11
 normal..... 12
 Mayberry, Bott..... 107
 Menger
 théorème..... 44
 multi-ensemble..... 7
 multi-graphe..... 7
 fini..... 7
- N**
- noeud..... 5
 nombre chromatique..... 123
- O**
- Ore (théorème)..... 62
 origine..... 5
- P**
- période..... 79

- père 51
- parcours
- en largeur 52
 - en profondeur 51, 52
 - infixe 52
 - préfixe 51
 - suffixe 51
- passerelle 29
- Perron
- théorème 77
 - valeur propre 74
- Perron-Frobenius (théorème) 74
- Petersen (graphe) 59
- piste 22
- poids 10
- point d'articulation 38
- polynôme chromatique 132
- Prüfer (codage) 98
- prédécesseur 6
- Prim (algorithme) 108
- R**
- racine 50
- Robbins (théorème de) 42
- rooted tree 50
- routage 28
- routeur 29
- S**
- sommet 5
- adjacent 6, 8
 - connecté 23
 - distance 24
 - fortement connecté 24
 - indépendant 37
 - voisin 6, 8
- sous-graphe 37
- clique 43
 - couvrant 38
 - propre 37
- sous-graphe induit 37
- SPF (Shortest Path First) 29
- squelette 13, 112
- Steinitz (théorème) 112
- subdivision 115
- successeur 6
- Szekeres 137
- T**
- théorème
- Chvátal 64
 - Chvátal-Erdős 66
 - Dirac 60
 - exclusion 116
 - Hoffman 94
 - Menger 44
 - Ore 62
 - quatre couleurs 131
 - Robbins 42
- tri topologique 48
- triangle 70
- TSP 12
- V**
- vertex 5
- voyageur de commerce 12