Chapitre 7

Ensembles et graphes

Les ensembles et les graphes sont des formalisme si « évidents » qu'on oublie parfois qu'il existe une définition formelle de ces concepts. Dans l'usage courant des ensembles il est très rarement nécessaire de faire référence aux définitions formelles, l'intuition suffit en général. Il est cependant intéressant de voir comment cette théorie est construite et comment des notions telles que l'ensemble vide, l'ensemble des ensembles, les relations et fonctions, etc. sont traitées formellement.

La théorie des graphes et des ensembles s'avère également nécessaire pour modéliser des situations qui sont strictement impossibles de modéliser directement en logique des prédicats. On peut montrer que la notion cardinalité (nombre d'éléments d'un ensemble) n'est pas directement représentable en logique des prédicats. Par conséquent, si on définit un langage logique composé des prédicats unaires *Homme* et *Femme*, il est impossible d'écrire une formule pour représenter le fait qu'il y a plus de femmes que d'hommes. De même, si l'on représente les arcs d'un graphe par des atomes de la forme $Arc(x, y)$, il est impossible d'écrire une formule pour représenter la connexité du graphe. C'est dans ce genre de situations qu'il faut recourir à la théorie des ensembles.

7.1 Théorie des ensembles

La théorie actuelle des ensembles est due à Zermello et Fraenkel (ZF), elle formalise les travaux de Cantor (1874). Elle contient des axiomes qui permettent de définir des ensembles de différentes manières, et de les comparer. Les prédicats de base du langage sont :

$\in$: appartenance à un ensemble

$=$: égalité de deux ensembles

Comme son nom l'indique, la théorie des ensembles ne traite que d'ensembles, tout objet considéré est un ensemble. Il n'y a donc pas de notion d'objet élémentaire qui ne serait pas un ensemble. La théorie contient les axiomes habituels de l'égalité : $\forall x(x = x)$, symétrie, transitivité, congruence (voir le chapitre sur la modélisation logique).

Les axiomes de la théorie des ensembles servent essentiellement à définir les différentes manières de construire des ensembles (paire, union, parties, compréhension, remplacement) et la manière de comparer les ensembles. Il y a également des axiomes plus techniques (infini, choix, fondation) qui s'avèrent nécessaires pour les travaux fondamentaux en mathématiques.

Axiome de la paire

Étant donné deux ensembles x et y il existe toujours un ensemble z, noté $\{x, y\}$, qui contient exactement x et y. Formellement :

$$\forall x \forall y \exists z (\forall u (u \in z \Leftrightarrow u = x \vee u = y))$$

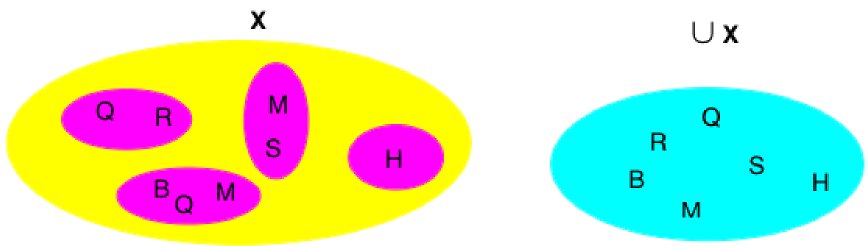
Si $x = y$, ce qui n'est pas interdit, z ne contient qu'un élément. On note $\{x\}$.

Axiome de la réunion

On peut construire à partir d'un ensemble e l'ensemble $\bigcup e$ qui contient tous les éléments des ensembles contenus dans e et eux seuls. Formellement

$$\forall e \exists z (\forall u (u \in z \Leftrightarrow \exists v (v \in e \wedge u \in v)))$$

Par exemple



En combinant les axiomes de la paire et de la réunion, à partir de n ensembles $x_1, \ldots, x_n$, on peut toujours construire un ensemble qui contient exactement ces ensembles. On le note $\{x_1, \ldots, x_n\}$. La construction procède de la manière suivante :

1. $x_1, x_2 \longrightarrow z_{12} = \{x_{1,} x_2\}$ (axiome de la paire)
2. $x_3, x_3 \longrightarrow z_3 = \{x_3\}$ (axiome de la paire)
3. $z_{12}, z_3 \longrightarrow w = \{z_{12}, z_3\}$ (axiome de la paire)
4. $w \longrightarrow z_{123} = \bigcup w$ contient x_1, x_2, x_3 (axiome de la réunion)
5. et ainsi de suite avec $x_4, \ldots, x_n$.

On peut ensuite construire des ensembles arbitrairement complexes en terme d'imbrication. Par exemple

$$x = \{a, b, \{t, u\}, h\}$$
$$y = \{\{\{a\}\}\}$$

on aura alors

$$\{\{a\}\} \in y$$

mais

$$a \notin y$$

Axiome d'extensionnalité

Cet axiome définit la notion d'égalité entre deux ensembles en disant que deux ensembles ont les mêmes éléments alors ils sont égaux.

$$\forall x \forall y (\forall z (z \in x \Leftrightarrow z \in y) \Rightarrow x = y)$$

Cet axiome nous permet d'écrire :

$$\{b, a, c\} = \{a, b, c\} = \{a, c, b, a\}$$

Si on utilise la notation$x \subseteq y$ pour abréger la formule $\forall z (z \in x \Rightarrow z \in y)$ on voit que

$$(x \subseteq y \wedge y \subseteq x) \Rightarrow x = y$$

Axiome de l'ensemble des parties

Pour tout ensemble x, il existe un ensemble dont les éléments sont tous les sous-ensembles de x. Cet ensemble se note habituellement $\mathcal{P}(x)$ ou 2^x.

Par exemple

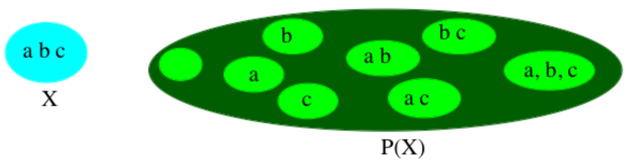


Schéma d'axiomes de compréhension ou de séparation

Pour tout ensemble x et toute formule $P(w)$ avec une variable libre exprimée dans un langage logique du premier ordre, il existe un ensemble dont les éléments sont les éléments de x vérifiant P. Notation :

$$\{w \in x | P[w]\}$$

Il s'agit d'un schéma d'axiome et non d'un seul axiome . Il y a un axiomes pour chaque formule P possible.

Exemple 7.1. Quelques ensembles de nombres entiers définis par compréhension :

- l'ensemble des nombres entiers qui sont des multiples de 7 : $\{w \in \mathbb{N} | \exists k(k \in \mathbb{N} \wedge k \times 7 = w)\}$ (on utilise le langage de la théorie des nombres)
- l'ensemble des étudiants qui ont moins de 22 ans : $\{z \in \text{Etudiants } | \text{age}(z) < 22\}$

Conséquences de cet axiome

L'existence d'un ensemble vide est une conséquence directe de l'axiome. Soit $V = \{z \in \{1, 2\} | \neg(z = z)\}$. Comme aucun z ne peut satisfaire cette formule il n'y a aucun élément dans V. Par l'axiome d'extensionalité V est unique, on le note $\varnothing$.

À l'autre extrémité du spectre, et contrairement à ce que l'on pourrait penser, il n'existe pas d'ensemble universel qui contient tous les ensembles.

Démonstration. Supposons qu'un tel ensemble U existe et définissions l'ensemble $B = \{x \in U \mid x \notin x\}$. Supposons que $B \in U$. On a alors deux possibilités : soit $B \in B$ et dans ce cas, par définition de B, on en déduit que $B \notin B$; soit $B \notin B$ et dans ce cas, toujours par définition de B on en déduit que $B \in B$. Donc, quelle que soit l'option prise on arrive à une proposition absurde. Ce qui démontre $\neg(B \in U)$. On a donc trouvé un ensemble B qui n'est pas dans l'ensemble supposé universel U. □

Intersection et différence

Il n'y a pas besoin d'axiomes pour définir les notions usuelles d'intersection et de différence de deux ensembles. Ces deux opérations sont en fait des abréviations pour simplifier l'écriture des formules. On les définit comme suit :

$$X \cap Y \stackrel{def}{=} \{z \in X | z \in Y\}$$
$$X \setminus Y \stackrel{def}{=} \{z \in X | z \notin Y\}$$

Schéma d'axiomes de remplacement

Définition 7.1. Un formule logique F possédant deux variables libres, disons x et y, est *fonctionnelle* si pour une valeur de x il n'y a qu'une valeur possible de y qui rend la formule vraie. Si la valeur de x est fixée il n'y a qu'une seule valeur possible de y pour rendre la formule vraie. Cette valeur est appelée l'*image* de x par F. Formellement F est fonctionnelle si $\forall x \forall y \forall z(F(x, y) \wedge F(x, z) \Rightarrow y = z)$. Par exemple, la formule (sur les entiers)

$$x + y = 5$$

est fonctionnelle. Par contre, la formule

$$x + y < 5$$

ne l'est pas. Pour $x = 3$ on peut choisir $y = 0$ ou 1 ou 2.

Le schémma d'axiomes de remplacement dit que si $F(x, y)$ est fonctionnelle alors pour tout ensemble z il existe un ensemble appelé $F(z)$ qui contient toutes les images des éléments de z. Formellement, pour toute formule fonctionnelle $F(x, y)$ on a un axiome

$$\forall z \exists u (\forall w (w \in u \Leftrightarrow \exists v (v \in z \wedge F(v, w))))$$

Exemple 7.2. Si F est la formule

$$y = 3x + 2$$

et

$$A = \{3, 6, 9\}$$

on aura

$$F(A) = \{11, 20, 29\}$$

7.2 Constructions pour la modélisation

La théorie des ensembles s'avère très efficace pour la modélisation conceptuelle (la représentation de concepts du monde et de leurs relations), pour autant qu'on l'équipe d'un certain nombre de constructions.

Paires ordonnées

Une paire ordonnée (a, b) est composée de deux éléments dont on distingue un premier (a) et un second (b). Techniquement, (a, b) est l'ensemble $\{\{a\}, \{a, b\}\}$. L'égalité de deux paire ordonnées est définie par

$$(a, b) = (a', b') \text{ si et seulement si } a = a' \text{ et } b = b'$$

Démonstration. Si $(a, b) = (a', b')$ alors, par l'axiome d'extensionalité $\{\{a\}, \{a, b\}\}$ et $\{\{a'\}, \{a', b'\}\}$ ont les mêmes éléments. Donc soit $\{a\} = \{a'\}$ et $\{a, b\} = \{a', b'\}$, soit $\{a\} = \{a', b'\}$ et $\{a, b\} = \{a'\}$. Si $a \neq b$ on a forcément $\{a\} = \{a'\}$ et $\{a, b\} = \{a', b'\}$ et donc $a = a'$ et $b = b'$. Si $a = b$ on aura forcément $a' = b'$ et donc $a = a'$ et $b = b'$. □

N-uplets et produit cartésien

On définit le triplet (a, b, c) comme $(a, (b, c))$ et le n-uplet $(a_1, a_2, \ldots a_n)$ comme $(a_1, (a_2, (\ldots, a_n) \ldots$

Le produit cartésien $E_1 \times E_2 \times \cdots \times E_n$ de n ensembles est formé de tous les n-uplets possibles $(a_1, \ldots, a_n)$ tels que $a_1 \in E_1, \ldots, a_n \in E_n$.

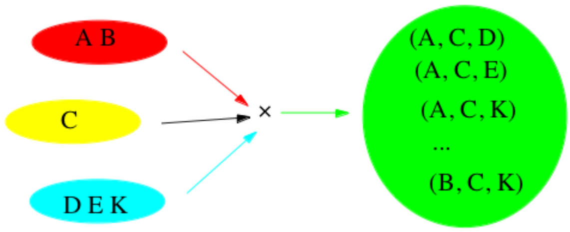


FIGURE 7.1 –

Exemple

Exemple 7.3. Dans un restaurant on a les ensembles

$$Viande = \{poulet, porc, boeuf\}$$
$$Sauce = \{citronelle, aigredoux, curry\}$$
$$Force = \{doux, normal, fort\}$$

La carte du restaurant est le produit cartésien

$$Carte = Viande \times Sauce \times Force$$
$$= \{(poulet, cirtonelle, doux), (poulet, citronelle, normal), \ldots, (boeuf, curry, fort)\}$$

Relations et Applications

Définition 7.2. Une *relation* n-aire sur $A_1, \cdots, A_n$ est un sous-ensemble de $A_1 \times \ldots \times A_n$.

Une relation est donc un ensemble de n-uplets sur $A_1, \cdots, A_n$. Chaque n-uplet peut être considéré comme représentant un lien entre un objet de A_1, un objet de A_2, ... et un objet de A_n. Par exemple, chaque triplet d'une relation MARIAGE associe une personne, une personne et une date (de mariage).

Définition 7.3. Une *application* ou *fonction* F d'un ensemble de départ A dans un ensemble d'arrivée B , notée $F : A \rightarrow B$ est une relation qui associe à chaque élément de l'ensemble de départ un et un seul élément de l'ensemble d'arrivée. Par exemple, la fonction AGE associe une personne à un et un seul nombre entier. Formellement, F doit satisfaire

1. $\forall x (x \in A \Rightarrow \exists y\, (x, y) \in F)$
2. $\forall x \forall y_1 \forall y_2 ((x, y_1) \in F \wedge (x, y_2) \in F \Rightarrow y_1 = y_2)$

Si $(x, y) \in F$ on écrit $f(x) = y$.

Définition 7.4. L'*image* de F est l'ensemble des y pour lesquels il existe un x tel que $(x, y) \in F$

Une fonction est *injective* si tous les éléments de l'ensemble de départ possèdent des images différentes, elle est *surjective* si son image couvre tout l'ensemble d'arrivée (pour chaque élément y de l'ensemble d'arrivée on a un x tel que $(x, y) \in F$. Une fonction injective et surjective est dite bijective (ou 1 à 1).

La notion de bijection permet de définir la cardinalité ou taille des ensembles.

Définition 7.5. On dit que deux ensembles (finis ou non) ont la *même cardinalité* s'il existe une bijection entre eux. Un ensemble fini E est de cardinalité n si et seulement si il existe une bijection entre E et l'ensemble d'entiers $\{1, \ldots, n\}$. On écrit $|E| = n$.

Une *fonction partielle* est une relation qui ne remplit que la condition 2 de la définition de fonction. Elle n'est donc pas définie pour tous les éléments de l'ensemble de départ.

Le *domaine* d'une fonction partielle F est l'ensemble des $x \in A$ pour lesquels il existe un y tel que $(x, y) \in F$ (les éléments qui ont une image). Pour une fonction, le domaine et l'ensemble de départ sont les mêmes.

Remarque sur les ensembles infinis

La définition de cardinalité (7.5) s'applique aussi bien aux ensembles finis qu'infinis. Par exemple, La fonction $f : x \mapsto 2x$ établit une bijection entre l'ensemble des entiers et l'ensemble des entiers pairs. Ces deux ensembles sont donc considérés comme de même taille (infinie).

Par contre on ne peut pas établir de bijection entre un ensemble E et l'ensemble $\mathcal{P}(E)$ de ses parties. Supposons qu'une telle bijection, appelons-la j, existe entre E et $\mathcal{P}(E)$. Définissons l'ensemble $A = \{x \in E : x \notin j(x)\}$. Comme j est une bijection et A est un sous-ensemble de E, il y a un et un seul y tel que $A = j(y)$. Mais cet y ne peut pas exister. En effet,

- soit $y \in j(y) = A$, ce qui entraîne, par définition de A, $y \notin j(y)$, contradiction
- soit $y \notin j(y) = A$, ce qui entraîne, par définition de A, $y \in A$, contradiction

$\square$

La conséquence de ce fait est l'existence d'une hiérarchie dans les infinis (il y a des ensembles « plus infinis » que d'autres !) Par exemple, $\mathcal{P}(\mathbb{N})$, l'ensemble des ensembles d'entiers), est « plus grand » que $\mathbb{N}$. De même $\mathcal{P}(\mathcal{P}((\mathbb{N}))$ et plus grand que $\mathcal{P}(\mathbb{N})$, et ainsi de suite.

7.3 Modélisation avec les ensembles

7.3.1 Modèles conceptuels et ensembles

La modélisation conceptuelle d'un domaine général (l'architecture, la biologie, etc.) ou spécifique (la gestion du personnel de l'entreprise X) consiste à décrire les concepts qui existent dans ce domaine, à les classer et à décrire leurs relations. Il y a une correspondance naturelle entre concepts et ensembles du fait qu'un concept est caractérisé par son extension qui est l'ensemble des individus qui appartiennent au concept. L'ensemble de toutes les locomotives existantes forme l'extension du concept *locomotive*.

D'autre part la *compréhension* d'un concept est la définition du concept par rapport aux autres (ses relations avec les autres). En termes de théorie des ensembles on représentera la compréhension par

– des relations entre les ensembles représentant les concepts
– des inclusions entre ces ensembles

Une inclusion $A \subseteq B$ signifie que tout objet qui fait partie du concept A fait aussi partie du concept B, le concept B est donc plus général que A.

Dans le domaine des système d'information de nombreux modèles conceptuels (ou sémantiques) ont été proposés, en particulier les modèles inspirés du modèle Entity-Relationship de Chen [1] ou du modèle Z d'Abrial [2].

Exemple 7.4. On veut représenter la notion de bâtiment et les notions qui lui sont liées.

$\mathsf{Batiment} \subseteq \mathsf{Construction}$: un bâtiment est une construction

$\mathsf{appartient} : \mathsf{Mur} \rightarrow \mathsf{Batiment}$: un mur appartient à un bâtiment

$\mathsf{toit} : \mathsf{Batiment} \rightarrow \mathsf{Toit}$ (bijection) : un bâtiment possède un et un seul toit et chaque toit est sur un et un seul bâtiment.

$\mathsf{Ouverture} = \mathsf{Porte} \cup \mathsf{Fenetre}$: une ouverture est soit une porte soit une fenêtre

$\mathsf{dans} : \mathsf{Ouverture} \rightarrow \mathsf{Mur}$: toute ouverture est dans un mur.

$\mathsf{adr} \subseteq \mathsf{Batiment} \times \mathsf{Adresse}$: la relation adr associe un bâtiment à une ou des adresses

$\mathsf{rue} : \mathsf{Adresse} \rightarrow \mathsf{Rue}$: une adresse est associée à une rue

$\mathsf{numero} : \mathsf{Adresse} \rightarrow \mathbb{N}$: une adresse possède un numéro qui est un entier

Ce type de modélisation est en général beaucoup plus compacte que l'équivalent avec des formules de logique des prédicats de base. De plus, il est relativement aisé à représenter graphiquement, par exemple avec des diagrammes entité-association ou des diagrammes de classe UML.

1. Chen, P. The Entity-Relationship Model-Toward a Unified View of Data. ACM Transactions on Database Systems 1/1/1976 pp. 9–36.
2. Abrial, J.-R. Data Semantics in Data Base Management (Kimbie, Koffeman, eds, North-Holland, 1974, pp. 1-59).

7.3.2 Une représentation graphique : les diagrammes de classe UML

Concept Un diagramme de classe [C] représente un concept **C** et l'ensemble C des objets (l'extension) du concept.

Associations Une flèche [C] -|> [D] représente la relation de sous-classe, donc d'inclusion $C \subseteq D$

Un association simple [C]–A–[D] représente une relation binaire entre C et D ($A \subseteq C \times D$)

On peut placer des contraintes de multiplicité aux extrémités de l'association [C]–A–(m..n)[D] signifie

$\forall x \in C(m \leq |\{y \in D : (x, y) \in A\}| \leq n)$, un objet x de **C** est associé à au moins m et au plus n objets de **D**. On représente l'absence de contrainte de borne supérieure par le symbole *.

Si on ajoute une contrainte « ordered » / « set »

Attributs Un diagramme de classe peut contenir des définitions d'attributs de la forme A :T (multiplicité)

Un attribut de multiplicité 1 correspond à une fonction $A : C \to T$. Chaque objet de C prend une valeur du type T pour l'attribut A.

Un attribut de multiplicité (m,n) correspond à une fonction $A : C \to P(T)$ telle que $m \leq |A(x)| \leq n$. Chaque objet de C prend comme valeur pour A un sous ensemble de au moins m et au plus n éléments de T.

7.4 Les diagrammes de classe UML

Depuis la fin des années 1990 le langage UML s'est imposé auprès des praticiens du génie logiciel et des systèmes d'information comme le langage de prédilection pour la modélisation conceptuelle. Les diagrammes de classe servent à décrire les différents types d'entités existant dans le système (informatique ou non) que l'on veut décrire

- Une classe représente un ensemble d'objets ayant des propriétés communes. Les objets faisant partie d'une classe sont appelés instances de la classe. « Un objet d'une classe doit contenir des valeurs pour chaque attribut en accord avec le type et la multiplicité de l'attribut. ». Un attribut est donc une fonction de C vers T, P(T), T*, P(T)|n, T<n, T*norep, ...
- Une association représente des liens entre objets de plusieurs classes. Une association peut posséder deux ou plus « extrémités », ce sont les classes liées par cette association. Un élément d'une association à n extrémités est un n-uplet d'objets de ces classes

– Une classes-association est une association dont les éléments possèdent des caractéristiques propres

Signification des diagrammes

Pour une classe C on notera C l'ensemble des instances de C

C1 (n1,m1) r1<– A –> r2 (n2,m2) C2

pour tout a dans A, r1(a) est dans C1 et r2(a) est dans C2

pour tout x1 dans C1 il existe au moins n1 éléments a de A tels que r1(a) = x1 (x1 participe à au moins n1 liens)

etc.

Pour une agrégation

Pour une composition

– Elle décrit les attributs que possèdent les entités de ce type

7.5 Graphes

Graphes simples

Un graphe représente un ensemble d'objets reliés entre eux.

Définition 7.6. Un graphe simple est une paire (V, E) où

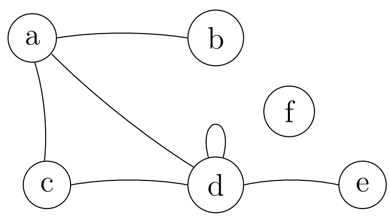
- V (vertex) est un ensemble (les sommets)
- E (edges) est un ensemble de paires (non ordonnées) de sommets (les arêtes)

Exemple

Le graphe (V, E) avec

- $V = \{a, b, c, d, e, f\}$
- $E = \{\{a, b\}, \{a, c\}, \{c, d\}, \{d, a\}, \{d\}, \{d, e\}\}$

Sous forme graphique



Graphe orienté

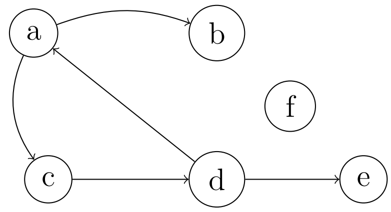
Un graphe orienté représente un ensemble d'objets reliés entre eux par des arcs qui ont un sens de parcours.

Un graphe orienté simple est une paire (V, E) où

– V (vertex) est un ensemble (les sommets)
– E (edges) est un ensemble de paires ordonnées de sommets (les arcs)

Exemple

Le graphe orienté (V, E) avec
– $V = \{a, b, c, d, e, f\}$
– $E = \{(a, b), (a, c), (c, d), (d, a), (d, e)\}$
Sous forme graphique



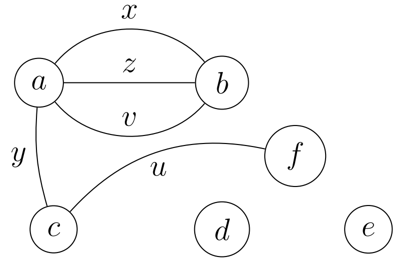
Graphes généraux

Un graphe simple a au plus une arête entre deux sommets. Un graphe général peut en avoir plusieurs. Dans ce cas il faut une autre définition

Un graphe simple est un triplet (V, E, g) où
– V (vertex) est un ensemble (les sommets)
– E (edges) est un ensemble (les arêtes)
– g est une fonction (d'incidence) $E \to P(V)$

Exemple

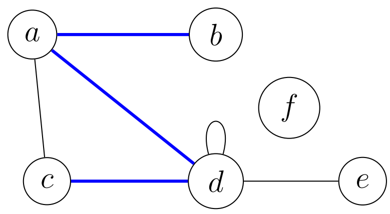
$V = \{a, b, c, d, e, f\}$
$E = \{x, y, z, u, v\}$
$g(x) = \{a, b\}, g(y) = \{a, c\}, g(z) = \{a, b\}, g(u) = \{c, f\}, g(v) = \{b, a\}$



Chemins et cycles

Un chemin dans un graphe est une séquence d'arêtes $(a_1, a_2, \ldots, a_n)$ telle que deux arêtes successives ont un sommet en commun.

Exemple. Le chemin $C = (\{b, a\}, \{a, d\}, \{d, c\})$



Un chemin est *simple* s'il ne passe pas deux fois par le même sommet.

Un chemin simple est un *cycle* si la première et la dernière arête ont un sommet commun (à eux seuls).

Chemins et circuits

Un chemin orienté (dans un graphe orienté) est une séquence d'arc $(a_1, a_2, \ldots, a_n)$ telle que deux arc successifs, le sommet de destination du premier est égal au sommet d'origine du second.

Un circuit est un chemin orienté qui commence et se termine sur le même sommet.

Graphes étiquetés

On peut ajouter de l'information à un graphe en associant une valeur à chaque sommet et/ou à chaque arête.

Pour un graphe $G = (V, E)$ l'étiquetage des sommets (resp. arêtes) est une fonction e de V (resp. E) dans l'ensemble des étiquettes (p.ex. l'ensemble des nombres entiers, des réels, des couleurs, etc.)

Modélisation avec les graphes

De nombreux problèmes peuvent se ramener à des problèmes sur les graphes

trouver une route	trouver un chemin
trouve la route la plus courte	chemin à coût minimal
durée d'un projet	chemin à coût maximal
réseau le moins cher	arbre de recouvrement min.
horaire d'une école	coloration d'un graphe
...	...

Connexité

On dit qu'un graphe est *connexe* s'il existe toujours un chemin entre n'importe quelle paire de ses sommets.

Une *composante connexe* d'un graphe quelconque est un sous-graphe qui est connexe et n'est contenu dans aucun autre sous-graphe connexe.

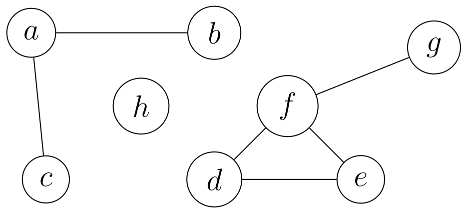


FIGURE 7.2 – Un graphe avec trois composantes connexes : $\{a, b, c\}$, $\{h\}$, $\{d, e, f, g\}$

Définition 7.7. Un *arbre* est un graphe connexe sans cycle.

Propriétés
- entre deux sommets y a un et un seul chemin
- (V, E) est un arbre $\Leftrightarrow$ il est connexe et $|V| = |E| + 1$
- tout graphe connexe contient (au moins) un arbre qui comprend tous ses sommets (arbre de recouvrement)

On peut choisir n'importe quel sommet comme racine de l'arbre. Le niveau d'un sommet est sa distance à la racine.

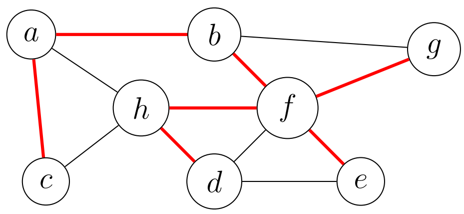


FIGURE 7.3 – Un graphe avec un arbre de recouvrement (arêtes rouges)

Modélisation avec les arbres

On retrouve les arbres dans de très nombreuses situations
- structures de choix (arbres de décision)
- structures emboitées (composant – composé, décomposition en sous-éléments)
- descendance (généalogie)
- structures de classement

Graphes orientés acycliques

(= DAG : Directed acyclic graph)

Une sorte de généralisation des arbres, elle apparaît dans des problèmes et structures comme :

- les hétérarchies (hiérarchies non strictes)
- les situations où il y a un ordre partiel (certains objets ne sont pas comparables)
- les problèmes d'ordonnancement
- ...