

Variables et structures

Notre but est d'utiliser un langage logique pour décrire des situations mathématiques concrètes. Pour cela, on introduit la notion de structure qui permet de relier les formules abstraites à des objets mathématiques réels. Avant de décider si, dans une structure donnée, une formule est vraie ou fausse, il est essentiel de distinguer les variables libres des variables liées.

1. Variables libres, variables liées

1.1. Idée

On considère un langage $\mathcal{L}$ avec des variables $x, y, z, x_1, x_2, \dots$. Pour une formule donnée, une variable peut être libre ou liée. Comprenons la différence par des exemples avant de donner la définition précise. Considérons une formule « $\forall x \ P(x)$ » ou « $\exists x \ P(x)$ », la variable « x » est ici *quantifiée*, c'est-à-dire *liée* à un quantificateur. De même, les deux variables x et y de « $\forall x \ \exists y \ Q(x, y)$ » sont quantifiées. Par contre, dans la formule « $\forall x \ Q(x, y)$ » la variable y n'est pas quantifiée, on dit que cette variable est *libre* (mais la formule est quand même légitime).

Voici une formule (dans une interprétation) :

$$\forall x \in \mathbb{R} \quad \exists y \in \mathbb{R} \quad y > x$$

Comme il n'y a pas de variables libres, on peut attribuer une valeur V/F à cette formule.

Par contre :

$$\forall x \in \mathbb{R} \quad y > x$$

est une formule où y est une variable libre. Cela n'a pas de sens de se demander si cette formule est vraie ou fausse tant qu'une valeur n'est pas attribuée à y .

1.2. Définition

Portée

Définition.

Dans une formule contenant « $\forall x \ P$ » ou « $\exists x \ P$ », la **portée** de $\forall$ (ou $\exists$) est $\mathcal{P}$.

Il est bon de préciser que $\mathcal{P}$ peut être une formule arbitrairement complexe.

Exemple :

$$\underbrace{\forall x \quad \exists y \quad \underbrace{P(x) \rightarrow Q(x, y)}_{\text{portée du } \exists}}_{\text{portée du } \forall}$$

Variable libre ou liée

Définition.

Dans une formule, une variable x est une **variable liée** si toute occurrence de x se trouve dans la portée d'un quantificateur $\forall x$ ou $\exists x$. Sinon on dit que x est une **variable libre**.

Exemples :

- $\forall x \quad P(x) \vee Q(x, y)$: x est liée, y est libre.
- $P(x) \rightarrow Q(x, y)$: x et y sont libres.
- $(\forall x \quad P(x)) \leftrightarrow (\exists y \quad P(y))$: x et y sont liées.
- $(\forall x \quad x = y) \vee (\exists y \quad x = y)$: x et y sont libres !

Il y a une subtilité à bien comprendre, illustrée par ce dernier exemple. Pour qu'une variable x soit libre, il suffit qu'il existe au moins une occurrence de x qui ne soit pas dans la portée d'un quantificateur. Dans notre exemple y est une variable libre pour la partie gauche et x libre pour la partie droite, donc pour la formule totale, x et y sont bien des variables libres.

Formule fermée**Définition.**

Une formule est **fermée** si elle ne contient aucune variable libre.

On dit aussi que c'est une formule **close**, ou bien que c'est une **phrase**.

L'idée est qu'à une formule fermée on va pouvoir attribuer une valeur V/F alors que pour une formule non fermée il reste encore des variables libres auxquelles il faudra affecter une valeur.

Exemples :

- $\forall x \quad (P(x) \wedge (\exists y \quad P(y)))$ est une formule fermée.
- $\exists x \quad ((\exists y \quad P(x) \wedge P(y)) \rightarrow (\forall y \quad Q(x, y)))$ est aussi une formule fermée.
- $\exists x \quad (Q(x, y) \rightarrow (\exists z \quad P(z)))$ n'est pas fermée car y est une variable libre.
- $(\forall x \quad P(x)) \vee (\exists x \quad Q(x)) \vee (\neg R(x))$ n'est pas une formule fermée car x est une variable libre (dans la sous-formule de droite).

1.3. Substitution

Dans un langage $\mathcal{L}$, soit $\mathcal{P}$ une formule, x une variable et t un terme.

Définition.

$\mathcal{P}[t/x]$ désigne la formule où toutes les occurrences libres de la variable x ont été remplacées par le terme t .

Remarque : on rappelle qu'un terme désigne les constantes, les variables et si le langage contient un symbole de fonction f , $f(t_1, \dots, t_n)$ est aussi un terme lorsque $t_1, \dots, t_n$ sont des termes.

Exemple.

Soit $\mathcal{L}$ un langage contenant les constantes a, b , une fonction binaire notée $+$, un prédicat unaire P et un prédicat binaire Q .

Étudions la formule :

$$\mathcal{P} : \quad \forall y \quad P(x) \rightarrow Q(x, y)$$

- $t = a$, $\mathcal{P}[t/x] : \forall y \quad P(a) \rightarrow Q(a, y)$
- $t = x + b$, $\mathcal{P}[t/x] : \forall y \quad P(x + b) \rightarrow Q(x + b, y)$
- $t = b$, $\mathcal{P}[t/y] : \forall y \quad P(x) \rightarrow Q(x, y)$: la formule ne change pas car toutes les occurrences de y sont dans la portée d'un quantificateur.

Remarque : la définition formelle de la substitution se fait par induction en commençant par les formules atomiques, puis l'introduction des connecteurs logiques. Pour les formules avec quantificateurs « $\forall x \mathcal{P}$ » ou « $\exists x \mathcal{P}$ » il n'y a pas de substitution de la variable x à faire car la variable est quantifiée.

On peut aussi faire une **substitution multiple** $\mathcal{P}[t_1/x_1, t_2/x_2, \dots, t_n/x_n]$. On la note $\mathcal{P}[s]$ où s désigne $(t_1/x_1, t_2/x_2, \dots, t_n/x_n)$. Il faut bien comprendre que les substitutions sont simultanées : on remplace toutes les occurrences libres de x_1 par t_1 et *en même temps* toutes les occurrences libres de x_2 par t_2 , etc.

Exemple.

Soit $\mathcal{P}$ la formule « $P(x, y)$ ». Alors :

- $\mathcal{P}[a/x, b/y] : P(a, b)$
- $\mathcal{P}[x + y/x, x/y] : P(x + y, x)$
- $\mathcal{P}[y/x, x/y] : P(y, x)$

Remarque : la formule $\mathcal{P} : \langle P(x) \rightarrow (\forall x Q(x)) \rangle$ est une formule légitime du langage (c'est comme si on avait deux x différents : un rouge et un bleu). La substitution de x par a donne $\mathcal{P}[a/x] : \langle P(a) \rightarrow (\forall x Q(x)) \rangle$ car on ne remplace que les occurrences libres de x . On préférerait écrire comme formule initiale $\mathcal{P}' : \langle P(x) \rightarrow (\forall y Q(y)) \rangle$ qui est plus facile à comprendre.

Capture de variable.

Lors d'une substitution $\mathcal{P}[t/x]$, on suppose toujours qu'aucune variable apparaissant dans t ne tombe dans la portée d'un quantificateur de $\mathcal{P}$. Si le terme t contient des variables qui risqueraient d'être liées par des quantificateurs présents dans $\mathcal{P}$, la substitution pourrait modifier le sens logique de la formule : on parle alors de capture de variables.

Par exemple, soit :

$$\mathcal{P} : \quad \forall y \quad P(x) \rightarrow Q(x, y)$$

La substitution de x par $t = y$ n'est pas légitime, car cela donnerait la formule :

$$\mathcal{P}[y/x] : \quad \forall y \quad P(y) \rightarrow Q(y, y)$$

qui n'a pas du tout le même sens.

Afin d'éviter ce phénomène, on renomme préalablement les variables liées de $\mathcal{P}$ avant d'effectuer la substitution. Pour notre exemple on écrira d'abord $\mathcal{P}$ sous une nouvelle forme :

$$\mathcal{P} : \quad \forall z \quad P(x) \rightarrow Q(x, z)$$

puis on effectue la substitution de x par y :

$$\mathcal{P}[y/x] : \quad \forall z \quad P(y) \rightarrow Q(y, z)$$

2. Structures

Un langage reste une coquille abstraite, il sert de cadre à une grande variété d'objets mathématiques et à leurs propriétés : les entiers, les groupes, les espaces vectoriels, la théorie des ensembles...

2.1. Définition

Définition.

On considère un langage $\mathcal{L}$. Une **structure** $\mathcal{S}$ de ce langage, c'est la donnée :

- d'un ensemble non vide E , appelé **domaine**,
- à chaque symbole de constante c de $\mathcal{L}$, on associe un élément $c^{\mathcal{S}} \in E$,

- à chaque symbole de prédicat P à n places, on associe un sous-ensemble $P^S \subset E^n$, autrement dit on associe une fonction $\phi_P : E^n \rightarrow \{V, F\}$ telle que $\phi_P(a_1, \dots, a_n)$ vaut vrai si et seulement si $(a_1, \dots, a_n) \in P^S$,
- à chaque symbole de fonction f à n places, on associe une fonction $f^S : E^n \rightarrow E$.

On dit aussi que S est une **interprétation** du langage $\mathcal{L}$. À un même langage, on peut associer plusieurs structures différentes.

Une structure est donc la réalisation mathématique concrète d'un langage. Pour définir une structure, il faut définir tous les symboles non logiques : les constantes, les prédicats et les fonctions. On ne touche pas aux symboles logiques : une variable x reste une variable x , sauf que maintenant elle parcourt le domaine E .

2.2. Un premier exemple

On considère un langage $\mathcal{L} = (c, P, f)$ où c est une constante, $P(x, y)$ un prédicat binaire et $f(x)$ une fonction unaire. Comme d'habitude, on ne précise pas explicitement que les variables sont x, y , qu'il y a les connecteurs logiques et qu'il y a aussi le prédicat d'égalité « $x = y$ ».

On va définir une structure qui correspond aux entiers naturels et modélise le successeur. On considère la structure S de domaine $E = \mathbb{N}$ avec l'interprétation du langage $(0, <, x \mapsto x + 1)$, c'est-à-dire :

- pour c la constante de $\mathcal{L}$, on définit $c^S = 0$, c'est l'interprétation du symbole c dans S ,
- pour P le prédicat de $\mathcal{L}$, on définit P^S , par $P^S(x, y) : x < y$,
- pour f la fonction de $\mathcal{L}$, on définit f^S , par $f^S(x) : \text{l'entier successeur de } x, \text{ que l'on note } x + 1$.

Une fois définie la structure, toute formule de $\mathcal{L}$ s'interprète en un énoncé mathématique sur la structure. Voici des exemples :

Formule de $\mathcal{L}$	Formule interprétée dans S	Vrai ou faux dans S ?
$\forall x \exists y P(x, y)$	$\forall x \in \mathbb{N} \exists y \in \mathbb{N} x < y$	C'est vrai
$\forall x P(c, f(x))$	$\forall x \in \mathbb{N} 0 < x + 1$	C'est aussi vrai
$\forall x f(x) = y$	$\forall x \in \mathbb{N} x + 1 = y$	La question n'a pas de sens car y est une variable libre

Pour définir cette structure S , on écrira souvent abusivement $c = 0, P(x, y) : x < y, f(x) = x + 1$ (au lieu de c^S, P^S, f^S).

Il y a évidemment beaucoup d'autres structures possibles pour ce même langage. On peut changer le domaine, la constante, le prédicat, la fonction :

- Structure S_1 de domaine $\mathbb{R}$, avec toujours $(0, <, x \mapsto x + 1)$, sauf que cette fois la formule interprétée $\forall x \in \mathbb{R} 0 < x + 1$ est devenue fausse.
- Structure S_2 de domaine $\mathbb{R}$, avec $(0, \leq, x \mapsto x^2)$, la formule $\forall x P(c, f(x))$ du langage devient $\forall x \in \mathbb{R} 0 \leq x^2$, une formule vraie dans cette interprétation.

2.3. Commentaires

Voici des remarques importantes.

Domaine non vide

Le domaine d'une structure doit être un ensemble non vide. Cela peut sembler un détail, mais c'est assez important. Il semble naturel pour tout le monde que :

Si « $\forall x \in E P(x)$ » est vrai, alors « $\exists x \in E P(x)$ » est aussi vrai.

Mais imaginons un instant que E soit l'ensemble vide, alors on aurait toujours « $\forall x \in E \ P(x)$ » qui est vrai (puisqu'il n'y a rien à vérifier, car pas de x à considérer). Par contre, « $\exists x \in E \ P(x)$ » est toujours faux (car E est vide). Cette situation serait problématique.

On peut aussi argumenter en disant que si E était vide, alors « $\forall x \in E \ x \neq x$ » est vrai et « $\exists x \in E \ x = x$ » est faux, ce qui serait très embêtant.

Bilan : on suppose toujours le domaine E non vide.

Fonctions

Revenons un peu sur les fonctions. Dans le langage, f est juste un symbole de fonction qui doit être suivi de n termes (pour une fonction à n places). Dans la structure, il faut définir l'interprétation de cette fonction, par une bonne vieille fonction : $f^S : E^n \rightarrow E$. C'est une fonction de n variables au sens usuel du terme. Comme déjà dit, dans la pratique on la note simplement $f : E^n \rightarrow E$.

Par exemple, soit $\mathcal{L}$ un langage avec une fonction unaire f et une fonction binaire g .

- Pour un domaine $\mathbb{N}$, on pourrait définir f par $f(x) = x + 1$ ou $f(x) = x^2$ et g par $g(x, y) = x + y$ ou $g(x, y) = xy$. Par contre, on ne peut pas définir f par $f(x) = x - 1$ car $f(0)$ n'est pas dans le domaine $\mathbb{N}$, ni définir g par $g(x, y) = x/y$ qui ne donne pas toujours des entiers.
- En revanche, sur le domaine $] -\infty, 0[$, on pourrait définir $f(x) = x - 1$ et $g(x, y) = -x/y$.

Prédicats

L'interprétation d'un prédicat est peut-être un peu plus difficile à comprendre. Dans le langage un prédicat n -aire est juste un symbole qui doit être suivi de n termes. Pour la structure il faut définir chaque prédicat, c'est-à-dire pour quels objets on obtient la valeur Vrai.

Prenons l'exemple d'un prédicat unaire P dans un langage $\mathcal{L}$. Pour définir une structure S , on définit son domaine E et ensuite il faut définir l'interprétation du prédicat, c'est-à-dire quelles sont toutes les valeurs de E qui rendent $P(x)$ vraie. On peut le faire en définissant un sous-ensemble $P^S \subset E$, qui est donc exactement l'ensemble des $d \in E$, tels que $P(d)$ soit vraie. On peut aussi le définir comme une fonction $\phi_P : E \rightarrow \{V, F\}$ qui renvoie Vrai ou Faux. Le lien entre les deux étant que $\phi_P(d) = V$ si et seulement si $d \in P^S$.

Ainsi, si on veut définir une interprétation de « être un réel plus grand que 7 », on pose $E = \mathbb{R}$ et pour le prédicat, soit on définit $P^S = [7, +\infty[$ l'ensemble des valeurs qui vérifient la propriété, ou bien on définit une fonction :

$$\begin{aligned} \phi_P : E &\longrightarrow \{V, F\} \\ x &\longmapsto \begin{cases} V & \text{si } x \in [7, +\infty[\\ F & \text{sinon.} \end{cases} \end{aligned}$$

Remarque : on a bien insisté sur l'exigence qu'un domaine doit être non vide. Par contre, on n'exige pas qu'un prédicat puisse être réalisé. C'est-à-dire qu'on peut avoir P qui s'évalue toujours à faux dans la structure. Ainsi sur $\mathbb{R}$ on a le droit de définir les prédicats « $x^2 < 0$ » ou bien « $x \neq x$ », même si aucun $x \in \mathbb{R}$ ne va les réaliser.

Pour définir un prédicat binaire $P(x, y)$ dans une structure de domaine E , il faut définir un ensemble $P^S \subset E^2$, qui sera l'ensemble des couples (d_1, d_2) tels que $P(d_1, d_2)$ soit vrai.

Domaine

Lorsque l'on définit une structure, on fixe un domaine E . Les variables sont alors interprétées comme des objets de ce domaine. Par exemple la formule du langage « $\forall x \exists y \ Q(x, y)$ » peut s'interpréter dans une structure où $E = \mathbb{N}$ par « $\forall x \in \mathbb{N} \exists y \in \mathbb{N} \ x < y$ ». Mais alors comment écrire une formule dans laquelle x et y ne parcourent pas exactement le même ensemble, comme par exemple « $\forall \varepsilon > 0 \exists n \in \mathbb{N} \ n > \frac{1}{\varepsilon}$ » ?

Il faut ruser un peu. Soient A et B deux sous-ensembles d'un même ensemble E . On souhaiterait écrire la phrase mathématique :

$$\forall x \in A \quad \exists y \in B \quad Q(x, y)$$

Elle ne peut pas être issue d'une formule du langage, car x et y doivent parcourir le même ensemble. Définissons un prédicat P_A , de sorte que $P_A(x)$ soit vrai exactement si $x \in A$. De même on définit P_B avec $P_B(x)$ vrai si $x \in B$. Considérons la formule du langage :

$$\forall x \quad (P_A(x) \rightarrow \exists y (P_B(y) \wedge Q(x, y)))$$

Elle s'interprète dans notre structure de domaine E (avec P_A et P_B définis comme ci-dessus) par :

$$\forall x \in E \quad (x \in A \rightarrow \exists y \in E (y \in B) \wedge Q(x, y))$$

Si $x \notin A$, alors cet énoncé est vrai (car l'implication est vraie par définition). Si $x \in A$, pour que cet énoncé soit vrai, il faut qu'il existe un $y \in B$ tel que $Q(x, y)$ soit vrai. C'est bien mathématiquement ce qu'on voulait.

2.4. D'autres exemples

Inverse

On souhaite exprimer l'existence d'un inverse (ou d'un opposé). C'est une notion très générale que l'on retrouve dans beaucoup de situations et qu'il est donc intéressant de conceptualiser. Soit $\mathcal{L}$ le langage muni d'une constante e et d'une fonction binaire $f(x, y)$ qu'on notera $x \circ y$. On peut écrire une formule qui exprime l'existence d'un inverse (à droite) :

$$\forall x \quad \exists y \quad x \circ y = e$$

On dit qu'un tel y est un *inverse (à droite)* de x .

Concrètement, voici ce que cela recouvre :

- **Opposé.** On définit la structure $\mathcal{S}$ de domaine $\mathbb{Z}$, avec $e^{\mathcal{S}} = 0$ et $f^{\mathcal{S}}(x, y) = x + y$. La formule du langage devient :

$$\forall x \in \mathbb{Z} \quad \exists y \in \mathbb{Z} \quad x + y = 0$$

Autrement dit, $y = -x$ est ici l'*opposé* de x .

- **Inverse multiplicatif.** Cette fois on définit la structure $\mathcal{S}$ de domaine $\mathbb{R}^*$, avec $e^{\mathcal{S}} = 1$ et $f^{\mathcal{S}}(x, y) = x \times y$. La formule du langage devient :

$$\forall x \in \mathbb{R}^* \quad \exists y \in \mathbb{R}^* \quad x \times y = 1$$

Autrement dit, $y = \frac{1}{x}$ est ici l'*inverse* de x (au sens algébrique).

- **Inverse de matrice.** Soit l'ensemble des matrices inversibles de taille $n \times n$, muni de la multiplication de matrices et $e^{\mathcal{S}} = I$ est la matrice identité. Dans cette structure, la formule exprime $A \times B = I$ c'est-à-dire que $B = A^{-1}$ est l'inverse de la matrice A .
- **Etc.** On pourrait continuer les exemples avec un groupe $(G, \diamond)$ d'élément neutre e , pour exprimer $g \diamond h = e$; ou bien la notion de bijection réciproque d'une fonction pour exprimer $g \circ h = \text{id}$, etc. Tout cela avec le même langage $\mathcal{L}$.

Il y a bien sûr d'autres formules du langage intéressantes pour toutes ces structures. L'associativité :

$$\forall x \quad \forall y \quad \forall z \quad x \circ (y \circ z) = (x \circ y) \circ z$$

L'action de l'élément neutre qui ne change rien :

$$\forall x \quad ((e \circ x) = x) \wedge ((x \circ e) = x)$$

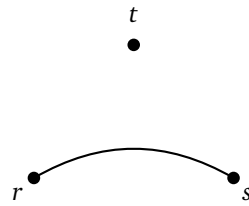
Graphe

Illustrons le fait qu'une même formule peut devenir vraie ou fausse en fonction de la structure choisie. Considérons le langage $\mathcal{L}$ ayant le prédicat $R(x, y)$ avec pour but de modéliser des graphes non orientés. Considérons la formule de $\mathcal{L}$:

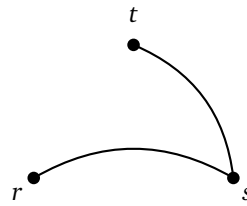
$$\forall x \quad \exists y \quad (x \neq y) \wedge R(x, y)$$

qu'il faut comprendre comme « tout sommet est relié à au moins un autre sommet ».

- On définit S_1 de domaine $E = \{r, s, t\}$ où R^{S_1} est défini par le graphe de gauche G_1 ci-dessous, c'est-à-dire que R est vrai seulement pour les couples (r, s) et (s, r) (qui sont reliés entre eux). Dans cette structure, la formule interprétée est fausse, car t n'est relié à personne.
- Si, en revanche, on définit S_2 de domaine $E = \{r, s, t\}$ où cette fois le prédicat est défini par le graphe de droite G_2 , alors la formule interprétée est vraie.



Graphe G_1



Graphe G_2

2.5. Exemples avancés

Anneaux

On considère un langage $\mathcal{L} = (0, 1, +, -, \times)$, où $0, 1$ sont des constantes, $+, -, \times$ sont des fonctions binaires (en toute rigueur on devrait écrire ces symboles a, b, f, g, h , car ici 0 n'est qu'un symbole du langage, noté ainsi par anticipation). Ce langage est adapté afin de modéliser la notion d'anneau (un ensemble avec addition, soustraction, multiplication, mais pas nécessairement d'inverse).

Voici des exemples de structures :

- **Entiers.** On considère la structure $\mathcal{S}$ de domaine $\mathbb{Z}$. Les constantes sont naturellement interprétées par les entiers 0 et 1 et les fonctions sont interprétées par les opérations usuelles sur les entiers.
- **Polynômes.** Pour cette structure $\mathcal{S}$, le domaine est l'ensemble des polynômes $\mathbb{R}[X]$; cette fois 0 désigne le polynôme nul, 1 le polynôme constant égal à 1 , les opérations sont les additions, soustractions et multiplications de polynômes.
- **Matrices.** Pour cette nouvelle structure, le domaine est l'ensemble $M_n(\mathbb{R})$ des matrices de taille $n \times n$. Naturellement 0 désigne la matrice nulle, 1 désigne la matrice identité I , les opérations sont les opérations sur les matrices.

L'intérêt du langage est donc de pouvoir formaliser des propriétés qui s'appliquent à de nombreuses situations.

Pour un anneau, on exige de plus que les formules suivantes soient vraies, une fois interprétées dans la structure :

- $\forall x \quad x - x = 0$: la soustraction correspond à l'opposé,
- $\forall x \quad 1 \times x = x$: 1 est élément neutre pour la multiplication,
- $\forall x \quad \forall y \quad \forall z \quad x \times (y + z) = x \times y + x \times z$: distributivité,
- etc.

Théorie des ensembles

On considère un langage $\mathcal{L} = (\in)$ où le symbole d'appartenance « $\in$ » désigne un prédicat binaire $P(x, y)$, ici noté $x \in y$.

Avertissement : dans les structures suivantes, les objets vont être des ensembles, et pour le prédicat $x \in y$ il s'agira de savoir si l'ensemble x est bien un *élément* de l'ensemble y . Par exemple $\{1\} \in \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$: c'est bien un ensemble qui est un élément d'un autre ensemble. Le symbole $\in$ ne désignera pas l'appartenance usuelle $1 \in \{1, 2, 3\}$ (puisque 1 et $\{1, 2, 3\}$ sont de nature différente, l'un est un entier, l'autre un ensemble d'entiers), ni l'inclusion $\{1, 2\} \subset \{1, 2, 3\}$ qui désigne autre chose. Construisons une première structure S_1 . Son domaine est $E = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\{\{\emptyset\}\}\}, \dots\}$, chaque élément est un singleton, sauf l'ensemble vide. Le prédicat est défini par les relations suivantes :

$$\emptyset \in \{\emptyset\} \quad \{\emptyset\} \in \{\{\emptyset\}\} \quad \dots$$

Les autres appartenances ne sont pas vérifiées.

Formellement on définit $E = \bigcup_{n \geq 0} E_n$ où :

$$E_0 = \{\emptyset\} \quad \text{et} \quad E_{n+1} = E_n \cup \{\{x\} \mid x \in E_n\}$$

Ainsi E_1 est formé à partir de $\emptyset$ et $\{\emptyset\}$, donc $E_1 = \{\emptyset, \{\emptyset\}\}$, E_2 est formé par E_1 union $\{\{\emptyset\}, \{\{\emptyset\}\}\}$ donc $E_2 = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$ (on ne compte pas $\{\emptyset\}$ deux fois).

Pour définir le prédicat dans cette structure, on donne l'ensemble des $(x, y) \in E^2$ tels que $x \in y$, c'est :

$$\in^S : \quad \{(x, \{x\}) \mid x \in E\}$$

Autrement dit, on a $x \in \{x\}$ et rien d'autre. Dans cette structure, si $x \in y$ alors $y = \{x\}$. On dit que $\{x\}$ est le *successeur* de x .

À quoi peut bien servir une construction si compliquée ? Si on note $\emptyset = 0$, $\{\emptyset\} = 1$, $\{\{\emptyset\}\} = 2$, alors 1 est le successeur de 0, 2 est le successeur de 1, etc. On a construit un modèle pour l'ensemble des entiers naturels $\mathbb{N}$, ayant ici pour seule propriété que tout entier admet un successeur.

Quand on construit une théorie des ensembles, on ajoute des axiomes ou des notions complémentaires. Voici donc des formules que l'on exige vraies quand on les interprète dans la structure :

- Égalité de deux ensembles. Deux ensembles sont égaux si, et seulement si, ils possèdent les mêmes éléments :

$$\forall x \quad \forall y \quad (x = y) \leftrightarrow (\forall z \quad (z \in x \leftrightarrow z \in y))$$

- Axiome de l'ensemble vide :

$$\exists x \quad \forall y \quad \neg(y \in x)$$

C'est-à-dire qu'il existe un ensemble ne contenant aucun élément.

- On définit l'inclusion par :

$$x \subset y \quad \text{signifie par définition} \quad \forall z \quad (z \in x \rightarrow z \in y)$$

- Axiome de l'union d'ensembles :

$$\forall x \quad \exists u \quad \forall z \quad (z \in u) \leftrightarrow (\exists y \quad (y \in x) \wedge (z \in y))$$

Par exemple si $x = \{\{1, 2\}, \{3\}, \{4, 5\}\}$, alors $u = \bigcup_x = \{1, 2, 3, 4, 5\}$ (y pourrait par exemple être $\{1, 2\}$ et alors z serait 1 ou 2).

On peut définir une autre structure liée à la théorie des ensembles. Cette fois on définit $E = \bigcup_{n \geq 0} E_n$ avec :

$$E_0 = \emptyset \quad \text{et} \quad E_{n+1} = \mathcal{P}(E_n)$$

où $\mathcal{P}(X)$ désigne l'ensemble des parties d'un ensemble X . Ainsi :

- $E_0 = \emptyset$
- $E_1 = \{\emptyset\}$, car le vide est la seule partie de E_0 ,
- $E_2 = \{\emptyset, \{\emptyset\}\}$, ce sont les parties de E_1 ,
- $E_3 = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$: le vide, deux singletons, une paire,
- ... E_n contient un nombre d'éléments qui croît très rapidement.

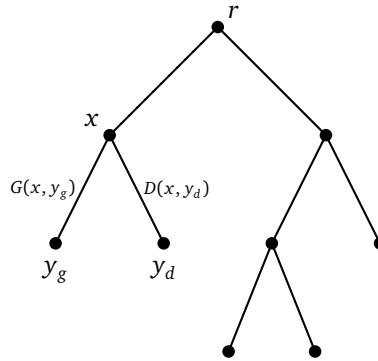
L'appartenance est ici l'appartenance usuelle, par exemple $\{\emptyset\} \in \{\emptyset, \{\emptyset\}\}$, par contre, $\emptyset \notin \{\{\emptyset\}\}$.

Arbres binaires

On souhaite modéliser la notion d'arbre binaire (chaque sommet a 0 ou 2 enfants). Soit un langage $\mathcal{L} = (r, G, D)$ où r est une constante, $G(x, y)$ et $D(x, y)$ sont des prédicats binaires.

On définit une structure $\mathcal{S}$ pour ce langage :

- E est un ensemble de sommets (éventuellement infini),
- $r \in E$ est un sommet distingué, appelé la *racine*,
- $G \subset E^2$ est la relation *enfant gauche*, $G(x, y)$ vaut vrai si y est l'enfant gauche de x ,
- $D \subset E^2$ est la relation *enfant droit*, $D(x, y)$ vaut vrai si y est l'enfant droit de x .



On définit une liste d'*axiomes*, c'est-à-dire des formules qui devront être vraies pour toute structure définissant un arbre binaire.

- Chaque sommet a au plus un enfant gauche et au plus un enfant droit :

$$\forall x \quad \forall y \quad \forall z \quad G(x, y) \wedge G(x, z) \rightarrow y = z$$

$$\forall x \quad \forall y \quad \forall z \quad D(x, y) \wedge D(x, z) \rightarrow y = z$$

- Les enfants gauche et droit sont distincts :

$$\forall x \quad \forall y \quad \forall z \quad G(x, y) \wedge D(x, z) \rightarrow y \neq z$$

- Un sommet n'a qu'un seul parent :

$$\forall x \quad \forall y \quad \forall z \quad G(x, z) \wedge G(y, z) \rightarrow x = y$$

$$\forall x \quad \forall y \quad \forall z \quad D(x, z) \wedge D(y, z) \rightarrow x = y$$

$$\forall x \quad \forall y \quad \forall z \quad G(x, z) \wedge D(y, z) \rightarrow x = y$$

- La racine n'est l'enfant de personne :

$$\forall x \quad \neg G(x, r)$$

$$\forall x \quad \neg D(x, r)$$

- Chaque sommet a soit zéro enfant, soit deux enfants :

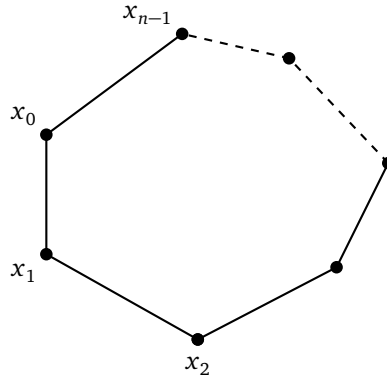
$$\forall x \quad (\exists y \quad G(x, y)) \leftrightarrow (\exists z \quad D(x, z))$$

- Axiome optionnel qui exige que l'arbre soit connexe (en un seul morceau), tout sommet autre que la racine doit avoir un parent :

$$\forall y \quad (y \neq r) \rightarrow (\exists x \quad G(x, y) \vee D(x, y))$$

(Cela assure la connexité dans le cas des arbres finis.)

Il reste un point délicat à exprimer : un arbre n'a pas de cycle (en suivant un chemin, on ne peut pas revenir sur son point de départ).



Cycle dans un graphe

L'absence de cycle n'est pas exprimable par une seule formule, ni même par un nombre fini de formules. Pour tout entier $n \geq 1$ et pour toute suite de relations $R_1, \dots, R_n \in \{G, D\}$, on impose que le chemin correspondant ne puisse pas revenir à son point de départ. Cela s'exprime par la formule :

$$\forall x_0 \quad \forall x_1 \quad \cdots \quad \forall x_{n-1} \quad \neg \left(R_1(x_0, x_1) \wedge R_2(x_1, x_2) \wedge \cdots \wedge R_n(x_{n-1}, x_0) \right)$$

Exercices

Variables libres, variables liées

Exercice 1. Pour chacune des formules suivantes, indiquer la portée de chaque quantificateur, lister les variables libres et les variables liées parmi x, y, z . La formule est-elle une formule fermée ?

1. $\forall x \quad P(x) \rightarrow Q(a, x)$
2. $\exists z \quad Q(x, y) \wedge Q(y, z) \wedge Q(z, x)$
3. $\forall x \quad (T(x, y, c) \rightarrow (\forall z \quad T(a, y, z)))$
4. $(\forall x \quad P(x)) \wedge (\exists y \quad P(y)) \wedge (x \neq y)$
5. $((\exists x \quad P(x)) \vee (\exists y \quad Q(x, y))) \rightarrow (\forall z \quad P(z))$
6. $(\exists x \quad \exists y \quad P(x) \wedge P(y) \wedge (x \neq y)) \rightarrow P(z)$
7. $\forall x \quad (Q(x, x) \rightarrow (\exists x \quad P(x)))$

Exercice 2. Pour chacune des formules suivantes données dans une structure, déterminer les variables libres et les variables liées (parmi x, y). Si la formule est fermée dire si la phrase est vraie ou fausse. Si la formule n'est pas fermée dire pour quelles valeurs des variables libres la formule devient vraie.

1. $\forall x \in \mathbb{N} \quad x^2 \geq y$
2. $\forall x \in \mathbb{N} \quad \exists y \in \mathbb{N} \quad x \geq y + 1$
3. $\forall y \in \mathbb{N} \quad \exists x \in \mathbb{N} \quad x \geq y + 1$
4. $\exists y \in \mathbb{R} \quad x \neq y$
5. $\forall y \in \mathbb{R} \quad x \neq y$
6. $(\forall x \in \mathbb{R} \quad x^2 \geq 0) \rightarrow (y = x^2 \rightarrow y \geq 0)$

Exercice 3. Réécrire les formules après chaque substitution indiquée.

1. $\mathcal{P}_1 : \forall x \quad (P(x) \wedge Q(x, y)); \mathcal{P}_1[b/y], \text{ puis } \mathcal{P}_1[a/x], \text{ puis } \mathcal{P}_1[x/y].$
2. $\mathcal{P}_2 : P(x) \rightarrow (\exists y \quad Q(x, y)); \mathcal{P}_2[x + a/x], \text{ puis } \mathcal{P}_2[y + b/y].$
3. $\mathcal{P}_3 : \exists z \quad (x \neq y \rightarrow T(x, y, z)); \mathcal{P}_3[a/x, b/y, c/z].$
4. $\mathcal{P}_4 : (\forall x \quad P(x)) \wedge (\exists y \quad Q(x, y)) \wedge (\forall z \quad \exists x \quad T(x, y, z)); \mathcal{P}_4[y/x, x/y, c/z].$

Structures

Exercice 4. On considère le langage $\mathcal{L} = (a, f)$ avec une constante a et une fonction binaire $f(x, y)$. Pour chaque formule et pour chaque structure, donner l'interprétation de la formule et dire si cette formule interprétée est vraie ou fausse.

Formules :

- $\mathcal{P}_1 : \forall x \exists y f(x, y) = a$
- $\mathcal{P}_2 : \forall x \exists y f(x, x) = f(f(y, y), y)$
- $\mathcal{P}_3 : \forall x \forall y f(x, y) = a \rightarrow x = a$

Structures :

1. $\mathcal{S}_i : E = \mathbb{Z}, a = 0, f(x, y) = x + y$
2. $\mathcal{S}_{ii} : E = \mathbb{R}, a = 0, f(x, y) = x - y$
3. $\mathcal{S}_{iii} : E =]0, +\infty[, a = 1, f(x, y) = xy$

Exercice 5. On considère le langage $\mathcal{L} = (a, b, P, Q)$ où a, b sont des constantes et $P(x)$ et $Q(x, y)$ des prédicats (les variables sont x, y). Pour chaque formule et pour chaque structure, donner l'interprétation de la formule et dire si cette formule interprétée est vraie ou fausse.

Formules :

- $\mathcal{P}_1 : \forall x Q(x, a)$
- $\mathcal{P}_2 : \exists x \forall y P(x) \wedge Q(x, y)$
- $\mathcal{P}_3 : (\forall x P(x)) \rightarrow (\exists y Q(a, y))$

Structures :

1. $\mathcal{S}_i : E = \mathbb{N}, a = 0, b = 1, P(x) : \ll x \neq 1 \gg, Q(x, y) : \ll x > y \gg$
2. $\mathcal{S}_{ii} : E = M_2$ l'ensemble des matrices 2×2 à coefficients réels, a la matrice nulle, b la matrice identité, $P(x) : \ll \det(x) \neq 0 \gg$ (le déterminant est non nul), $Q(x, y) : \ll xy = I \gg$ (le produit vaut l'identité)

Exercice 6. Soit $\mathcal{L}$ un langage avec un prédicat $R(x, y)$. On considère $E = \{r, s, t, u\}$. Pour chacune des formules suivantes, définir une interprétation du prédicat $R(x, y)$ sous la forme d'un graphe non orienté afin que la formule soit vraie dans cette structure. (Il peut y avoir plusieurs solutions possibles.)

1. $\forall x \forall y (x \neq y) \rightarrow R(x, y)$
2. $(\forall x R(x, x)) \wedge (\forall x \exists y \neg R(x, y))$
3. $\forall x \forall y \forall z (R(x, y) \wedge R(x, z)) \rightarrow (y = z)$
4. $\neg (\forall x \forall y \forall z (R(x, z) \wedge R(z, y)) \rightarrow R(x, y))$