

Calcul des prédicats

Le **calcul des prédicats**, ou **logique du premier ordre**, est une **théorie formelle** (au sens où il s'agit de manipuler des *formules*) qui modélise le raisonnement mathématique.

Des exemples de théories mathématiques modélisables en logique du premier ordre sont les **entiers naturels** (via les **axiomes de Peano**) ou les mathématiques entières (via les **axiomes de Zermelo-Fraenkel**). Il existe cependant des limitations bien connues à la possibilité de réaliser ces modélisations, comme l'a démontré Gödel avec ses **théorèmes d'incomplétude**.

Définitions

On fait comme d'habitude une distinction entre **syntaxe** et **sémantique**. Si la première est à peine plus complexe que la syntaxe du **Calcul des propositions**, la deuxième est beaucoup plus riche, car n'importe quelle structure mathématique peut faire de modèle à la logique du premier ordre.

Syntaxe

La logique du premier ordre a été conçue pour décrire n'importe quelle structure mathématique, comme par exemple les nombres naturels. Avant de pouvoir énoncer des propriétés (i.e., des théorèmes) d'une structure mathématique, nous devons définir quel est son *langage*, i.e. quelles sont les façons légitimes de combiner les objets de la structure.

Un **langage**, ou **signature**, est la donnée des éléments suivants (tous en quantité pas nécessairement finie):

- Des **symboles de constante**, par exemple: $0, 1, 2, \dots$
- Des **symboles de fonction**, par exemple: f, g, h, \dots . À chaque symbole est associé un entier strictement positif appelé **arité de la fonction**; ce nombre indique le nombre d'arguments pris par la fonction.
- Des **prédicats** (ou **symboles de relation**), par exemple: P, Q, R, \dots . À chaque symbole est aussi associé une **arité**, qui peut être dans ce cas positive ou nulle.

Par exemple, le langage de l'arithmétique est constitué de:

- les constantes $0, 1, 2, \dots$;
- les symboles de fonction $+, -, \times, /$ (tous de arité 2);
- les relations $=, \leq$ (tous de arité 2).

On peut combiner les symboles d'un langage pour former des **termes**. Pour cela, nous nous donnons un ensemble **dénombrable** de **variables** x, y, z, \dots alors un **terme** est

- soit une variable,
- soit un symbole de constante,
- soit l'application d'un symbole de fonction d'arité n à n termes.

Par exemple

$$0 + 1, \quad x - 3, \quad (y + 5) \times 6$$

sont des termes, mais

$$-5, \quad x = y, \quad ++3, xy$$

n'en sont pas. Un terme qui ne contient pas de variables est dit **clos**.

Remarque: ni les parenthèses ni le point ne font pas partie du langage. Ils sont simplement là pour rendre les termes non-ambigus, i.e. pour indiquer dans quel ordre les fonctions ont été appliquées pour obtenir le terme. Voir [du bon usage des parenthèses](#).

Une **formule atomique** est un prédicat du langage de arité n appliqué à n termes. Par exemple

$$x = y, \quad (2 + y) = x, \quad (0 + 1) = (3 - 2)$$

sont des formules atomiques, mais

$$2 + 3, \quad 2 + (y = x), \quad y = x = z, 0 \leq 1 \leq 2$$

n'en sont pas (même si certaines d'entre elles sont des raccourcis communément utilisés).

Finalement nous sommes prêts à définir la syntaxe de la logique du premier ordre. Nous nous donnons les **connecteurs logiques suivants**

- Les connecteurs du **Calcul des propositions**: $\wedge, \vee, \neg, \rightarrow, \dots$;
- Les **quantificateurs**
 - \forall , qui se lit *pour tout*,
 - \exists , qui se lit *il existe*.

Une **formule du premier ordre** (ou **formule bien formée**) est:

- soit une formule atomique,
- soit, si ϕ et ψ sont deux formules du premier ordre, l'une des expressions suivantes:
 - $\square \phi \wedge \psi$, ou $\phi \vee \psi$, ou $\neg \phi$, ou $\phi \rightarrow \psi$, etc.,
 - $\square \exists x. \phi$, ou $\forall x. \phi$, pour une variable quelconque x .

Voici des exemples de formules:

$$\forall x. (x \leq 0 \vee x = 0), \quad (\exists x. x + 2 = 0) \wedge (\exists y. z = 2)$$

et des non-exemples:

$$\forall x \leq 0. x = y, \quad \exists x. x, \quad (\exists x. x = y) + 1$$

(même si la première est un raccourci courant).

Variables libres et liées

Une variable est dite **libre** si elle n'est quantifiée par aucun quantificateur (dans son sous-arbre syntaxique). Elle est dite **liée** sinon. Par exemple, dans le prédicat

$$\forall x. (P(x) \vee Q(y))$$

la variable x est liée, tandis que y est libre. De façon moins évidente, dans le prédicat

$$(\forall x. P(x)) \vee Q(x)$$

la première occurrence de la variable x est liée, tandis que la deuxième est libre (en effet, le quantificateur \forall ne porte pas sur elle, car il est dans un sous-arbre différent).

Une formule sans variables libres est dite **close**. Une formule ϕ de variables libres x, y, \dots est parfois notée $\phi(x, y, \dots)$ lorsque l'on veut indiquer clairement les noms de ses variables libres. Ceci est intéressant, par exemple, lorsque l'on forme un prédicat plus long à partir de ϕ , comme par exemple

$$\forall x. \exists y. \phi(x, y).$$

Priorité des opérateurs

Les règles de priorité des opérateurs sont les mêmes que pour le [Calcul des propositions](#). Les quantificateurs ont par convention la priorité la plus basse, ainsi

$$\forall x. P(x) \vee Q(x)$$

est à lire comme

$$\forall x. (P(x) \vee Q(x))$$

et non pas comme

$$(\forall x. P(x)) \vee Q(x)$$

On omet aussi les parenthèses lorsque plusieurs quantificateurs se suivent. Ainsi, la seule façon d'interpréter

$$\forall x. \exists y. P(x, y)$$

est

$$\forall x. (\exists y. P(x, y))$$

Autres conventions

En dehors du cadre de la logique, le calcul des prédicats est utilisé pour exprimer de façon succincte des énoncés mathématiques. Pour avoir plus de souplesse et de maniabilité, l'on accepte dans ce cas d'autres raccourcis de notation qui ne sont pas standard dans la théorie de la logique du premier ordre.

Les raccourcis les plus courants sont

- $\forall x, y. P(x, y)$ pour $\forall x. \forall y. P(x, y)$;
- $\forall P(x). Q(x)$, pour $\forall x. P(x) \rightarrow Q(x)$;
- $\exists P(x). Q(x)$, pour $\exists x. P(x) \wedge Q(x)$.

Ces raccourcis peuvent être utilisés, par exemple, pour écrire

$$\forall x, y \in \mathbb{N}. x + y \geq 0.$$

Une autre convention courant consiste à utiliser le symbole \Rightarrow pour l'implication, à la place du symbole \rightarrow qui a déjà trop d'utilisations en algèbre. Dans la suite de ce cours, nous allons rigoureusement éviter d'utiliser ces raccourcis.

Sémantique

Comme en [Calcul des propositions](#), donner une sémantique au calcul des prédicats revient à attacher une *interprétation* à chaque symbole, et ensuite à chaque prédicat. Les choses sont rendues plus complexes par le fait que le calcul des prédicats permet de modéliser toutes les mathématiques, et que donc le nombre d'interprétations possibles est infini (alors que, rappelons-le, dans une proposition avec n formules atomique a seulement 2^n modèles distincts).

La première chose à fixer pour interpréter le calcul des prédicats est un **domaine du discours**, en général un ensemble (par exemple les entiers), dans lequel on va interpréter tous les **termes**.

Une fois le **domaine du discours** fixé, on définit un **modèle** (ou **interprétation**, ou **structure du premier ordre**) comme l'assignation

- d'un élément du domaine à chaque *symbole de constante* ;
- d'une *fonction* du domaine dans lui-même à chaque *symbole de fonction* ;
- d'une *relation* sur le domaine à chaque *symbole de relation*.

Par exemple, si le langage est celui de l'arithmétique, un modèle possible sera l'assignation

- des nombres **zéro**, **un**, ... aux constantes 0, 1, ...
- des opérations d'addition, soustraction, ... aux symboles +, -, ...
- des relations d'égalité, d'ordre, ... aux symboles =, ≤, ...

Remarque : Tout ceci peut paraître une inutile lapalissade, car on donne à chaque symbole la signification qu'on lui attribue déjà ordinairement. Gardez à l'esprit, toutefois, qu'un même symbole peut être interprété de plusieurs façons différentes : les chiffres peuvent représenter des nombres en *base deux*, les opérations arithmétiques peuvent être faites *modulo un entier*, etc. La théorie du calcul des prédicats s'intéresse aux **metathéorèmes** qu'on peut démontrer *indépendamment de la façon d'interpréter* le langage ; il est donc naturel de monter à ce niveau d'abstraction, afin de pouvoir attacher des modèles exotiques au calcul des prédicats.

Une fois le modèle fixé, on peut passer à définir l'interprétation des prédicats. On commence par se donner une **affectation des variables** μ , c'est à dire une fonction $\mu(x)$ qui à chaque variable x, y, \dots associe un élément du domaine.

On peut maintenant définir l'interprétation d'un terme. L'interprétation d'un terme t , **sous l'affectation** μ est définie *inductivement* par

- si t est une constante, l'élément du domaine qui est lui est associé ;
- si t est une variable x , l'élément du domaine $\mu(x)$;
- si t est de la forme $f(t_1, t_2, \dots)$, avec f un symbole de fonction et t_1, t_2, \dots des termes, l'interprétation de f appliquée aux interprétations de t_1, t_2 , etc.

Par exemple, si le terme est $x + 1$, en supposant que $\mu(x) = 3$, son interprétation sous μ est

- 1 est interprété comme le nombre **un**,
- x est interprété comme le nombre **trois**,
- $x + 1$ est interprété comme **un plus trois**, c'est à dire **quatre**.

On peut finalement définir l'interprétation d'un prédicat ϕ **sous une affectation** μ . L'interprétation d'un prédicat ne va plus être un élément du domaine, mais plutôt une valeur de vérité (vrai ou faux).

- Si ϕ est de la forme $P(t_1, t_2, \dots)$, où P est un symbole de relation et t_1, t_2, \dots sont des termes, l'interprétation de P appliquée aux interprétations de t_1, t_2 , etc.
- Si ϕ est la forme $\psi \bullet \chi$, avec \bullet un opérateur du [Calcul des propositions](#) et ψ et χ deux prédicats, la valeur de vérité de ϕ est obtenue à partir des valeurs de vérité de ψ et χ grâce à la table de vérité de \bullet .
- Si ϕ est de la forme $\exists x. \psi$, son interprétation est vraie s'il existe une autre affectation μ' , qui diffère de μ seulement en x , et telle que l'interprétation de ψ sous μ' est vraie. En pratique ceci modélise le fait qu'il existe une valeur du domaine qui rend vrai ψ .
- Si ϕ est de la forme $\forall x. \psi$, son interprétation est vraie si pour toute affectation μ' , qui diffère de μ seulement en x , l'interprétation de ψ sous μ' est vraie. En pratique ceci modélise le fait que toute valeur du domaine rend vrai ψ .

Exemple : on fixe $\mu(x) = 2$ et $\mu(y) = 1$. La formule $\forall x. (x > 0) \rightarrow (x > y)$ est fausse car il existe un μ' , par exemple $\mu'(x) = 1, \mu'(y) = 1$, qui diffère de μ seulement en x et qui rend $(x > 0) \rightarrow (x > y)$ fausse. Au contraire, si on prend $\mu(x) = 2$ et $\mu(y) = 0$, la même formule devient vraie (sous l'affectation μ). On observe que la valeur de $\mu(x)$ ne change rien à la vérité de la formule, car x y est **lié**.

Lorsque la formule ϕ ne contient pas de variable **libre**, sa valeur de vérité ne dépend pas de l'affectation initiale μ , et on dit simplement que ϕ est vraie ou fausse dans le modèle.

Exemple : $\forall x. \forall y. \exists z. (x < z < y)$ est fausse dans les entiers. En effet, pour n'importe quelle affectation initiale, il suffit de prendre $\mu'(x) = 0$ et $\mu'(y) = 1$ pour voir qu'il n'existe pas de choix pour z qui vérifie $x < z < y$. Par contre le même prédicat est vrai dans les rationnels, il existe en effet toujours un nombre rationnel compris entre deux nombres rationnels distincts. On voit ici que, même si la vérité d'un prédicat clos ne dépend pas de l'affectation, elle peut bien dépendre du modèle choisi.

Un prédicat **clos** qui est vrai dans un modèle est dit **satisfaisable**, un prédicat qui est vrai dans tout modèle est dit **valide**. Ceci est analogue aux définitions de satisfaisabilité et de tautologie en [Calcul des prédicats](#).

Puisque ces notions ne sont bien définies que pour les prédicats clos, on s'intéresse par la suite principalement à ceux-ci. Une convention courante pour traiter les prédicats non-clos consiste à convenir que toutes les variables libres sont **universellement quantifiées**. Ainsi lorsque l'on s'interroge sur la validité du prédicat

$$\forall x. x < y$$

il est sous-entendu qu'on parle de la validité de

$$\forall y. \forall x. x < y$$

Théorie des modèles

Comme pour le [Calcul des propositions](#), nous nous intéressons aux équivalences entre prédicats qui sont valides indépendamment de la façon dont on interprète les variables. Lorsque un prédicat ϕ est **valide** on note

$$\models \phi$$

Si ϕ et ψ sont deux prédicats, et si tout modèle qui rend vrai ϕ rend aussi vrai ψ , on dit que ψ est une **conséquence logique** (ou **sémantique**) de ϕ et on note

$$\phi \models \psi$$

Si $\phi \models \psi$ et $\psi \models \phi$ on dit que ϕ et ψ sont équivalentes et on note

$$\phi \equiv \psi \quad \text{ou} \quad \phi \Leftrightarrow \psi \quad \text{ou} \quad \phi = \psi$$

En général ces équivalences sont indépendantes du langage employé. Ainsi les équivalences qu'on peut espérer démontrer ressembleront plutôt à

$$\neg(\forall x. \psi) \equiv \exists x. \neg\psi$$

pour tout prédicat ψ .

Exercice : Montrer que cette équivalence est bien vérifiée.

α -renommage

Une des méthodes les plus simples pour obtenir des équivalences consiste à renommer certaines variables sans altérer la sémantique du prédicat. Ce procédé s'appelle α -renommage (ou α -conversion).

Si ϕ est une formule, contenant une variable x liée, on peut remplacer toutes les occurrences liées de x par une nouvelle variable y (en faisant attention à ce que y ne soit pas déjà employée et libre dans ϕ), sans altérer la sémantique de ϕ .

Par exemple, si $\phi(y)$ est la formule

$$(\forall x. (x = 2 \vee y = 3)) \vee (\forall x. x = 3)$$

elle peut être changée en

$$(\forall x. (x = 2 \vee y = 3)) \vee (\forall z. z = 3)$$

sans en altérer la sémantique. Lorsque l'on procède à un α -renommage il faut faire attention à ce que la nouvelle variable ne soit pas déjà libre dans la formule ; en effet la formule

$$(\forall y. (y = 2 \vee y = 3)) \vee (\forall x. x = 3)$$

n'est pas équivalente à $\phi(y)$ (en effet, y n'y est plus libre).

Il est aussi possible de renommer les variables libres, par exemple y dans la formule précédente, mais dans ce cas il ne fait pas beaucoup de sens de parler d'équivalence, car la formule n'est pas close. Par exemple, la formule

$$(\forall x. (x = 2 \vee z = 3)) \vee (\forall x. x = 3)$$

n'est pas équivalente à $\phi(y)$, néanmoins on peut noter $\phi(z)$ cette nouvelle formule. En général, si $\phi(x)$ est une formule de variable libre x , on notera $\phi(y)$ ou encore $\phi[y/x]$ la même formule dans laquelle toutes les occurrences libres de x ont été remplacées par y .

Forme normale prénexe

La **forme normale prénexe** est une technique permettant de transformer un prédicat quelconque en un prédicat équivalent sous une forme standard. Un prédicat est dit en **forme normale prénexe** (en anglais, PNF) s'il est de la forme

$$\Theta a. \Theta b. \dots \Theta z. P(a, b, \dots, z)$$

où les Θ sont des quantificateurs \forall ou \exists , et $P(a, b, \dots, z)$ est un prédicat sans quantificateurs. Pour tout prédicat il existe un prédicat sémantiquement équivalent (en logique classique) en PNF ; ce prédicat peut être obtenu à l'aide de quelques transformations élémentaires, que nous listons ci-dessous.

Les équivalences remarquables du calcul des propositions :

- Lois de De Morgan : $\neg(A \vee B) \equiv \neg A \wedge \neg B$, $\neg(A \wedge B) \equiv \neg A \vee \neg B$;
- Distributivité : $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$, $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$;
- Implication : $A \rightarrow B \equiv \neg A \vee B$;
- Double négation : $\neg\neg A \equiv A$.

Les équivalences remarquables du calcul des prédicats :

- $\neg\forall x. P \equiv \exists x. \neg P$;
- $\neg\exists x. P \equiv \forall x. \neg P$;
- $(\forall x. P) \wedge Q \equiv \forall x. (P \wedge Q)$ seulement si x n'est pas libre dans Q ;
- $(\forall x. P) \vee Q \equiv \forall x. (P \vee Q)$ seulement si x n'est pas libre dans Q ;
- $(\exists x. P) \wedge Q \equiv \exists x. (P \wedge Q)$ seulement si x n'est pas libre dans Q ;
- $(\exists x. P) \vee Q \equiv \exists x. (P \vee Q)$ seulement si x n'est pas libre dans Q .

Moyennant un α -renommage de la variable x , ces quatre dernières règles peuvent être appliquées en toute circonstance.

D'autres règles de transformation, notamment concernant le connecteur \rightarrow , peuvent être déduites à partir de celles-ci.

Exercice : Prouver les équivalences remarquables ci-dessus.

Théorie de la preuve

La **Théorie de la preuve** de la logique du premier ordre n'est pas plus compliquée que celle du **Calcul des propositions**: en effet les **systèmes de preuve** sont les mêmes, après ajout de quelques règles d'inférence pour les quantificateurs. Voir **Théorie de la preuve**.



2011-2020 Mélanie Boudard <<http://melanie.boudard.free.fr/>>, Christina Boura <<http://christina-boura.info/en/content/home>>, Luca De Feo <<http://defeo.lu>>, licensed under the **Creative Commons 4.0 Attribution-ShareAlike** <<http://creativecommons.org/licenses/by-sa/4.0/>>.