

# Aide-mémoire de logique n° 3 – Logique du premier ordre

## Syntaxe

Les formules de la logique du premier ordre comprennent :

- des connecteurs logiques, les mêmes qu'en logique propositionnelle
- des variables. On considère qu'une infinité de variables différentes est disponible et leur ensemble est noté  $\mathcal{V}$
- le quantificateur universel  $\forall$  et le quantificateur existentiel  $\exists$
- des symboles de fonctions et des symboles de prédicats, chacun muni d'une *arité* qui représente son nombre d'arguments.

L'ensemble  $\mathcal{F}$  des symboles de fonctions et celui  $\mathcal{P}$  des symboles de prédicats varient en fonction du langage du premier ordre considéré. Un langage est caractérisé par sa signature  $\Sigma = (\mathcal{F}, \mathcal{P})$

Un **terme** d'un langage du premier ordre est :

- soit une variable (un élément de  $\mathcal{V}$ ),
- soit un symbole de fonction suivi entre parenthèses d'un nombre de sous-termes égal à son arité (définition inductive : les sous-termes sont eux-mêmes des termes).

Un terme représente un objet du domaine du discours.

Un **atome** (ou formule atomique) d'un langage du premier ordre est un symbole de prédicat suivi entre parenthèses d'un nombre de termes égal à son arité.

Un atome représente une relation entre un certain nombre d'objets du domaine.

**Conventions :**

- un symbole de fonction d'arité 0 (sans argument) est appelé *constante*, un symbole de prédicat d'arité 0 est appelé *proposition*, et on n'écrit pas les parenthèses vides (on les considère directement comme un terme et un atome respectivement).
- lorsqu'on ne précise pas explicitement la nature des symboles employés, les lettres minuscules de a à e représentent des constantes, les lettres minuscules de f à j des fonctions, les lettres minuscules de s à z des variables, et les lettres majuscules des prédicats.

Une **formule** du premier ordre est construite à partir d'atomes, de connecteurs et de quantificateurs. Tout atome est une formule, deux formules quelconques peuvent être reliées par un connecteur binaire ( $\wedge, \vee, \Rightarrow, \Leftrightarrow$ ) et une formule peut être précédée d'une négation ou d'un quantificateur suivi d'une variable. Pour les connecteurs, les priorités sont les mêmes qu'en propositionnel ( $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$  dans l'ordre), et les quantificateurs ont la priorité la plus faible (ils sont pris en compte en dernier).

## Sémantique

Une *réalisation* d'un langage du premier ordre de signature  $\Sigma$  est un couple  $(\mathcal{D}, \mathbf{I})$ , où  $\mathcal{D}$  est un ensemble **non vide** quelconque, appelé domaine du discours, et  $\mathbf{I}$  une fonction qui à chaque symbole de  $\Sigma$  associe une interprétation :

- si  $f$  est un symbole de fonction d'arité  $n$ ,  $\mathbf{I}(f)$  est une application de  $\mathcal{D}^n$  dans  $\mathcal{D}$ . (Si  $n = 0$ , c'est simplement une valeur de  $\mathcal{D}$ .)
- si  $P$  est un symbole de prédicat d'arité  $n$ ,  $\mathbf{I}(P)$  est un sous-ensemble de  $\mathcal{D}^n$ , c'est-à-dire un ensemble de  $n$ -uplets. (Si  $n = 0$ ,  $\mathbf{I}(P)$  est simplement une valeur de vérité.)

Une *affectation* ou *valuation* des variables, étant donné un langage **et** une réalisation de ce langage, de domaine  $\mathcal{D}$ , est une application  $\theta : \mathcal{V} \rightarrow \mathcal{D}$ .

La *valeur* d'un **terme** du langage pour une réalisation  $(\mathcal{D}, \mathbf{I})$  et une affectation  $\theta$  données est un élément de  $\mathcal{D}$  (un objet du domaine). Elle est notée  $\text{Val}_{(\mathcal{D}, \mathbf{I})}(t, \theta)$ . Elle est calculée en remplaçant chaque variable par son affectation et chaque symbole de fonction par son interprétation.

La *valeur de vérité* (ou *évaluation*) d'une **formule** du langage est définie en fonction d'une réalisation  $(\mathcal{D}, \mathbf{I})$  et d'une affectation  $\theta$ , comme suit :

- Pour un atome,  $\mathcal{E}(P(t_1, \dots, t_n), (\mathcal{D}, \mathbf{I}), \theta)$  vaut **V** si et seulement si le  $n$ -uplet  $(\text{Val}_{(\mathcal{D}, \mathbf{I})}(t_1, \theta), \dots, \text{Val}_{(\mathcal{D}, \mathbf{I})}(t_n, \theta))$  appartient à l'interprétation de  $P$  (c'est-à-dire à l'ensemble  $\mathbf{I}(P)$ ).
- Pour les connecteurs, les règles sont les mêmes qu'en propositionnel (tables de vérité).
- Pour les quantificateurs, on modifie l'affectation  $\theta$  pour faire parcourir à la variable quantifiée toutes les valeurs de  $\mathcal{D}$  et on considère l'ensemble des valeurs de vérité obtenues pour la sous-formule :
  - $\mathcal{E}(\forall x \varphi, (\mathcal{D}, \mathbf{I}), \theta)$  vaut **V** si et seulement si  $\mathcal{E}(\varphi, (\mathcal{D}, \mathbf{I}), \theta\{x \mapsto d\})$  vaut toujours **V** (quel que soit  $d \in \mathcal{D}$ ).
  - $\mathcal{E}(\exists x \varphi, (\mathcal{D}, \mathbf{I}), \theta)$  vaut **V** si et seulement si  $\mathcal{E}(\varphi, (\mathcal{D}, \mathbf{I}), \theta\{x \mapsto d\})$  ne vaut pas toujours **F** (c'est-à-dire vaut **V** pour au moins un  $d \in \mathcal{D}$ ). On rappelle que l'ensemble  $\mathcal{D}$  ne peut pas être vide.

On dit que  $\psi$  est une *conséquence* de  $\varphi$ , et on note  $\varphi \models \psi$ , si pour toute réalisation  $(\mathcal{D}, \mathbf{I})$  et toute affectation  $\theta$  telles que  $\mathcal{E}(\varphi, (\mathcal{D}, \mathbf{I}), \theta) = \mathbf{V}$ , on a également  $\mathcal{E}(\psi, (\mathcal{D}, \mathbf{I}), \theta) = \mathbf{V}$ .

$\varphi$  et  $\psi$  sont dites *équivalentes*,  $\varphi \equiv \psi$ , si  $\varphi \models \psi$  et  $\psi \models \varphi$ .

## Variabes libres et liées, formules ouvertes et fermées

Dans une formule, les occurrences d'une variable  $x$  qui se situent dans une sous-formule commençant par  $\forall x$  ou  $\exists x$  sont dites *liées*. Les autres occurrences sont dites *libres*. Il est possible de renommer les occurrences liées d'une variable sans changer la signification de la formule (ce sont des variables muettes). Les règles d'équivalence sont, **si  $y$  est une variable n'apparaissant pas déjà dans  $\varphi$**  :  $\forall x\varphi \equiv \forall y\varphi[y/x]$  et  $\exists x\varphi \equiv \exists y\varphi[y/x]$ .

La formule est dite *normalisée* si :

- une variable donnée n'a soit que des occurrences liées, soit que des occurrences libres, et
- tous les quantificateurs concernent des variables différentes.

Une formule qui ne comprend aucune variable libre est dite *fermée*. Une formule fermée  $\varphi$  a la propriété que  $\mathcal{E}(\varphi, (\mathcal{D}, \mathcal{I}), \theta)$  **ne dépend pas de  $\theta$** . On peut donc écrire simplement  $\mathcal{E}(\varphi, (\mathcal{D}, \mathcal{I}))$  dans ce cas.

## Modèles, satisfaisabilité, validité

Soit  $\varphi$  une formule **fermée** sur un langage de signature  $\Sigma$ . Un *modèle* de  $\varphi$  est une réalisation  $(\mathcal{D}, \mathcal{I})$  de  $\Sigma$  telle que  $\mathcal{E}(\varphi, (\mathcal{D}, \mathcal{I})) = \mathbf{V}$ . Si  $(\mathcal{D}, \mathcal{I})$  est un modèle de  $\varphi$ , on dit qu'il *satisfait*  $\varphi$  et on note  $(\mathcal{D}, \mathcal{I}) \models \varphi$ .

Une formule (fermée)  $\varphi$  est dite *valide* si toute réalisation est un modèle de  $\varphi$ . On note alors  $\models \varphi$ .

Une formule est dite *satisfaisable* s'il existe une réalisation qui la satisfait. Sinon, elle est dite insatisfaisable.

Un *ensemble* de formules est dit satisfaisable s'il existe une même réalisation qui satisfait à la fois toutes les formules de l'ensemble. Il est dit valide si toutes les formules qu'il contient sont valides.

## Propriétés des relations binaires

Les symboles de prédicat d'arité 2 représentent des relations binaires. Beaucoup de propriétés des relations binaires sont représentables par des formules du premier ordre et ont des noms : la relation représentée par  $P^2$  dans une réalisation  $(\mathcal{D}, \mathcal{I})$  est dite :

- réflexive si  $(\mathcal{D}, \mathcal{I}) \models \forall x P(x, x)$
- irreflexive si  $(\mathcal{D}, \mathcal{I}) \models \forall x \neg P(x, x)$
- symétrique si  $(\mathcal{D}, \mathcal{I}) \models \forall x \forall y P(x, y) \Rightarrow P(y, x)$
- transitive si  $(\mathcal{D}, \mathcal{I}) \models \forall x \forall y \forall z P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$

**Relation d'égalité** Normalement, en logique du premier ordre, les symboles de prédicat utilisés dans une formule peuvent être interprétés par n'importe quelle relation. Il est souvent utile de faire un cas particulier pour la relation d'égalité : on utilise le symbole  $=$  dont l'interprétation est imposée. Étant donné un domaine  $\mathcal{D}$ , on ne considère que les interprétations  $\mathcal{I}$  telles que  $\mathcal{I}(=) = \{(x, x) \mid x \in \mathcal{D}\}$ .

Certaines propriétés ne peuvent être exprimées qu'en utilisant le prédicat d'égalité. On peut ainsi ajouter aux propriétés des relations binaires l'antisymétrie :

$\mathcal{I}(P)$  est *antisymétrique* si  $(\mathcal{D}, \mathcal{I}) \models \forall x \forall y \neg(x = y) \Rightarrow \neg(P(x, y) \wedge P(y, x))$

## Classification des relations transitives

Une relation binaire transitive est :

- une relation de préordre si elle est réflexive, sans condition de symétrie (ex. être de taille inférieure ou égale à)
- une relation d'ordre strict si elle est irreflexive (ex. être strictement plus gros que). **Remarque** : Une relation d'ordre strict est obligatoirement antisymétrique (exercice).
- une relation d'équivalence si elle est réflexive et symétrique (ex. être de la même espèce, avoir le même nom)
- une relation d'ordre (sous-entendu large) si elle est réflexive et antisymétrique (ex.  $\leq$  sur les nombres mais aussi inclusion pour les ensembles, divisibilité pour les entiers naturels, ...). **Remarque** : pour caractériser une relation d'ordre, on a besoin du prédicat d'égalité, ce qui n'est pas le cas pour les types de relations précédents.

**Remarque** : à partir d'un préordre on peut toujours construire une relation d'équivalence et une relation d'ordre strict en posant :

$$\forall x \forall y P_{eq}(x, y) \Leftrightarrow P(x, y) \wedge P(y, x)$$

$$\forall x \forall y P_{ord}(x, y) \Leftrightarrow P(x, y) \wedge \neg P(y, x)$$