

# De la logique propositionnelle à la logique du premier ordre

David Delahaye

Faculté des Sciences  
[David.Delahaye@lirmm.fr](mailto:David.Delahaye@lirmm.fr)

Licence L3 2022-2023

# Limites de la logique propositionnelle

## Problèmes d'expressivité

- L'« atome » est la variable propositionnelle, qui est indécomposable ;
- Comment rendre compte de points communs entre propositions ?  
« Marie dort » et « Pierre ne dort pas » ;
- Comment représenter le partage d'entités ?  
« Pierre ne dort pas » et « Pierre regarde Marie ».

# Logique du premier ordre (ou calcul des prédicats)

## Définitions préliminaires

- $\mathcal{V} \equiv$  ensemble de variables d'individu  $x, y$ , etc. ;
- $\mathcal{S}_{\mathcal{F}} \equiv$  ensemble de symboles de fonctions  $f, g$ , etc. ;
- $\mathcal{S}_{\mathcal{P}} \equiv$  ensemble de symboles de prédicats  $P, Q$ , etc. ;
- $\mathcal{S}_{\mathcal{F}} \cap \mathcal{S}_{\mathcal{P}} = \emptyset$  ;
- Arité (nombre d'arguments)  $m : \mathcal{S}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{P}} \rightarrow \mathbb{N}$  :
  - ▶ Exemple : pour  $f(x, y)$  avec  $f \in \mathcal{S}_{\mathcal{F}}$ ,  $m(f) = 2$  ;
  - ▶ Exemple : pour  $P(x, y, z)$  avec  $P \in \mathcal{S}_{\mathcal{P}}$ ,  $m(P) = 3$ .

# Logique du premier ordre (ou calcul des prédicats)

## Termes du premier ordre

- Plus petit ensemble  $\mathcal{T}$  t.q. :
  - ▶ Si  $x \in \mathcal{V}$  alors  $x \in \mathcal{T}$  ;
  - ▶ Si  $f \in \mathcal{S}_{\mathcal{F}}$  d'arité  $n$  et  $t_1, \dots, t_n \in \mathcal{T}$  alors  $f(t_1, \dots, t_n) \in \mathcal{T}$ .
- Les constantes sont des fonctions d'arité 0 ;
- Important : dans un premier temps, nous ne considérerons que les fonctions d'arité 0 (constantes) dans nos exemples.

## Formules du premier ordre

- Plus petit ensemble  $\mathcal{F}$  t.q. :
  - ▶ Si  $P \in \mathcal{S}_{\mathcal{P}}$  d'arité  $n$  et  $t_1, \dots, t_n \in \mathcal{T}$  alors  $P(t_1, \dots, t_n) \in \mathcal{F}$  ;
  - ▶  $\perp, \top \in \mathcal{F}$  ;
  - ▶ Si  $\Phi \in \mathcal{F}$  alors  $\neg \Phi \in \mathcal{F}$  ;
  - ▶ Si  $\Phi, \Phi' \in \mathcal{F}$  alors  $\Phi \wedge \Phi', \Phi \vee \Phi', \Phi \Rightarrow \Phi', \Phi \Leftrightarrow \Phi' \in \mathcal{F}$  ;
  - ▶ Si  $x \in \mathcal{V}$  et  $\Phi \in \mathcal{F}$  alors  $\forall x. \Phi, \exists x. \Phi \in \mathcal{F}$ .

# Logique du premier ordre (ou calcul des prédicats)

## Associativité et précedence des connecteurs

- Inchangées par rapport à la logique propositionnelle.

## Notation pointée pour les quantificateurs

- La portée d'un quantificateur va jusqu'à la parenthèse fermante de la formule du quantificateur ;
- Si la formule du quantificateur n'est pas parenthésée, la portée du quantificateur va jusqu'à la fin de la formule ;
- Donc, si on veut arrêter la portée d'un quantificateur, il suffit d'utiliser des parenthèses pour limiter explicitement la portée du quantificateur ;
- Exemple :
  - ▶  $\exists x.P(x) \Rightarrow P(a) \wedge P(b) \equiv \exists x.(P(x) \Rightarrow P(a) \wedge P(b))$  ;
  - ▶ Si on veut que le  $\exists$  ne porte que sur  $P(x)$ , on doit écrire :  
 $(\exists x.P(x)) \Rightarrow P(a) \wedge P(b)$ .
- Notation :  $\forall x, y.\Phi \equiv \forall x.\forall y.\Phi$  (idem pour  $\exists$ ).

# Variables libres, variables liées

## Définitions

- Une variable  $x$  est libre dans une formule  $\Phi$  ssi il existe une occurrence de  $x$  dans  $\Phi$  qui n'est sous la portée d'aucun quantificateur ;
- Une variable  $x$  est liée dans une formule  $\Phi$  ssi il existe une occurrence de  $x$  dans  $\Phi$  qui est sous la portée d'un quantificateur ;
- Occurrence  $\equiv$  position d'un terme/formule dans une formule.

## Définitions

- L'ensemble des variables libres  $FV(\Phi)$  et l'ensemble des variables liées  $BV(\Phi)$  d'une formule  $\Phi$  sont définis par récurrence structurale par :
  - ▶ Si  $x \in \mathcal{V}$  alors  $FV(x) = \{x\}$ ,  $BV(x) = \emptyset$ ;
  - ▶ Si  $f \in \mathcal{S}_{\mathcal{F}}$  d'arité  $n$  et  $t_1, \dots, t_n \in \mathcal{T}$  alors  
 $FV(f(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$ ,  $BV(f(t_1, \dots, t_n)) = \emptyset$ ;
  - ▶ Si  $P \in \mathcal{S}_{\mathcal{P}}$  d'arité  $n$  et  $t_1, \dots, t_n \in \mathcal{T}$  alors  
 $FV(P(t_1, \dots, t_n)) = FV(t_1) \cup \dots \cup FV(t_n)$ ,  $BV(P(t_1, \dots, t_n)) = \emptyset$ ;
  - ▶  $FV(\top) = FV(\perp) = \emptyset$ ,  $BV(\top) = BV(\perp) = \emptyset$ ;
  - ▶ Si  $\Phi \in \mathcal{F}$  alors  $FV(\neg\Phi) = FV(\Phi)$ ,  $BV(\neg\Phi) = BV(\Phi)$ ;
  - ▶ Si  $\Phi, \Phi' \in \mathcal{F}$  alors  $FV(\Phi \wedge \Phi') = FV(\Phi \vee \Phi') = FV(\Phi \Rightarrow \Phi') = FV(\Phi \Leftrightarrow \Phi') = FV(\Phi) \cup FV(\Phi')$ ,  
 $BV(\Phi \wedge \Phi') = BV(\Phi \vee \Phi') = BV(\Phi \Rightarrow \Phi') = BV(\Phi \Leftrightarrow \Phi') = BV(\Phi) \cup BV(\Phi')$ ;
  - ▶ Si  $x \in \mathcal{V}$  et  $\Phi \in \mathcal{F}$  alors  $FV(\forall x.\Phi) = FV(\exists x.\Phi) = FV(\Phi) \setminus \{x\}$ ,  
 $BV(\forall x.\Phi) = BV(\exists x.\Phi) = BV(\Phi) \cup \{x\}$ .

# Variables libres, variables liées

## Exemples

- $y$  est libre dans  $\forall x.P(x, y)$  ;
- $x$  est liée dans  $\forall x.P(x, y)$  ;
- Dans la formule  $(\forall x.P(x, y)) \wedge (\exists z.Q(z) \vee R(t))$  :
  - ▶ L'ensemble des variables libres est  $\{y, t\}$  ;
  - ▶ L'ensemble des variables liées est  $\{x, z\}$ .
- Une variable peut être libre et liée à la fois (c'est-à-dire qu'elle possède une occurrence où elle est libre et une autre où elle est liée), par exemple :  $(\forall x.P(x, y)) \wedge Q(x)$ , où  $x$  est libre (deuxième occurrence) et liée (première occurrence) à la fois.



# Variables libres, variables liées

## Formule polie ou propre

- Une formule est polie ou propre si aucune variable n'est à la fois libre et liée dans cette formule, et si aucune variable liée n'est soumise à plus d'une quantification ;
- Exemples :
  - ▶  $(\forall x.P(x, y)) \wedge (\exists z.Q(z) \vee R(t))$  est une formule polie ;
  - ▶  $(\forall x.P(x, y)) \wedge Q(x)$  n'est pas une formule polie ;
  - ▶  $(\forall x.P(x, y)) \wedge \exists x.Q(x)$  n'est pas une formule polie.

# Variables libres, variables liées

## $\alpha$ -conversion

- Il est toujours possible de renommer les variables liées d'une formule (en utilisant des variables « fraîches ») sans changer la validité de cette formule ;
- Ce processus est appelé  $\alpha$ -conversion ;
- On peut donc toujours transformer une formule non polie en une formule polie par  $\alpha$ -conversion ;
- Exemple :  $(\forall x.P(x, y)) \wedge Q(x)$  peut être transformée en  $(\forall z.P(z, y)) \wedge Q(x)$ , où l'occurrence liée de  $x$  a été transformée en  $z$ .

# Variables libres, variables liées

## Formule close

- Une formule est close ou fermée si aucune variable n'est libre dans cette formule ;
- Un énoncé est une formule close ;
- Une théorie est un ensemble d'énoncés.

## Conditions nécessaires et suffisantes

- Dire que  $A$  est une condition nécessaire pour  $B$  signifie que pour que  $B$  soit réalisée, il faut que  $A$  le soit :  $B \Rightarrow A$  ;
- Dire que  $A$  est une condition suffisante pour  $B$  signifie que si  $A$  est réalisée alors  $B$  le sera :  $A \Rightarrow B$  ;
- Dire que  $A$  est une condition nécessaire et suffisante pour  $B$  signifie que  $A$  et  $B$  sont réalisées en même temps :  $A \Leftrightarrow B$ .

## Conditions nécessaires et suffisantes

- Condition nécessaire : « Il est nécessaire d'avoir le permis de conduire pour conduire une voiture ».
  - Modélisation :
    - ▶  $P(x) \equiv x$  a le permis de conduire ;
    - ▶  $C(x) \equiv x$  conduit une voiture.
- $\forall x. C(x) \Rightarrow P(x).$

## Conditions nécessaires et suffisantes

- Condition suffisante : « Il suffit qu'il neige à Montpellier pour qu'il neige à Oslo » ;
  - Modélisation :
    - ▶  $N(x) \equiv$  il neige à  $x$  ;
    - ▶  $m \equiv$  Montpellier ;
    - ▶  $o \equiv$  Oslo.
- $N(m) \Rightarrow N(o).$

## Prédicats de « typage »

- La logique du premier ordre peut être sortée (avec une ou plusieurs sortes) afin de typer les termes du premier ordre manipulés ;
- En l'absence de sortes, il faut avoir recours à des prédicats qui vont jouer ce rôle de typage ;
- Par exemple, « Les chats n'aiment pas les chiens » :
  - ▶  $Chat(x) \equiv x$  est un chat ;
  - ▶  $Chien(x) \equiv x$  est un chien ;
  - ▶  $A(x, y) \equiv x$  aime  $y$ .

$$\forall x. Chat(x) \Rightarrow \forall y. Chien(y) \Rightarrow \neg A(x, y).$$

## Prédicats de « typage »

- Attention au connecteur utilisé pour introduire les prédicats de typage (selon qu'il s'agit d'un  $\forall$  ou d'un  $\exists$ ) ;
- « Tous les chats aiment boire du lait » :
  - ▶  $Chat(x) \equiv x$  est un chat ;
  - ▶  $B(x) \equiv x$  aime boire du lait.

$$\forall x. Chat(x) \Rightarrow B(x).$$

- « Il existe un chat qui n'aime pas boire du lait » :
  - ▶  $Chat(x) \equiv x$  est un chat ;
  - ▶  $B(x) \equiv x$  aime boire du lait.

$$\exists x. Chat(x) \wedge \neg B(x).$$



## Modélisations équivalentes

- Deux formules peuvent être équivalentes (même sémantique) même si elles ne sont pas égales syntaxiquement ;
- De ce fait, deux modélisations d'un même problème peuvent être équivalentes même si elles ne sont pas syntaxiquement égales ;
- Par exemple, « Les chats n'aiment pas les chiens » :
  - ▶  $Chat(x) \equiv x \text{ est un chat ;}$
  - ▶  $Chien(x) \equiv x \text{ est un chien ;}$
  - ▶  $A(x, y) \equiv x \text{ aime } y.$

$\forall x. Chat(x) \Rightarrow \forall y. Chien(y) \Rightarrow \neg A(x, y),$

$\forall x, y. Chat(x) \Rightarrow Chien(y) \Rightarrow \neg A(x, y),$

et  $\forall x, y. Chat(x) \wedge Chien(y) \Rightarrow \neg A(x, y)$

sont des modélisations équivalentes.