# Cours de programmation fonctionnelle et logique
# Cours 9 : éléments de logique

## Table des matières



## 1 Langages du premier ordre

### 1.1 Lexique

- *Prédicat* : fonction à valeurs booléennes, c'est-à-dire qui renvoie soit « vrai », soit « faux ».
- *Arité* d'une fonction : nombre d'arguments admis par cette fonction.
- *Variables propositionnelles* : elles sont considérées comme des prédicats 0-aire (fonctions sans arguments et à valeurs booléennes). Ce sont donc des variables à valeurs booléennes.
- *Fonctions.*
- *Constantes (d'individus)* : fonctions 0-aires.
- *Variables* dites *individuelles.*
- *Connecteurs* : $\neg$ (non), $\wedge$ (et), $\vee$ (ou), $\rightarrow$ (implique) et $\leftrightarrow$ (équivalent à).
- *Quantificateurs* : $\forall$ (quel que soit), $\exists$ (il existe).
- *Parenthèses* : pour éliminer les ambiguïtés.

*Langage du premier ordre* : les quantificateurs ne portent que sur les individus.

$$\exists x, r(x) = \text{vrai : autorisé}$$

$$\exists r, r(x) = \text{vrai : interdit}$$

### 1.2 Termes

Définition :

- Les variables d'individus et les constantes d'individus sont des termes.
- Si $t_1, ..., t_n$ sont des termes, $f_i^n(t_1, ..., t_n)$ est un terme ($f_i^n$ est la $i^{\text{e}}$fonction d'arité $n$).

Un terme est un élément de base du langage.

### 1.3 Formules

Il existe trois règles de formation des formules :

1. Introduction de prédicats : si $t_1, ..., t_n$ sont des termes, $r_i^n(t_1, ..., t_n)$ est une formule dite atomique ($r_i^n$ est le $i^{\text{e}}$prédicat d'arité $n$). En particulier : les variables propositionnelles sont des formules dites atomiques.
2. Introduction de connecteurs : si $A$ et $B$ sont deux formules, $\neg(A)$, $(A \wedge B)$ ; $(A \vee B)$, $(A \to B)$ et $(A \leftrightarrow B)$ sont des formules.
3. Introduction de quantificateurs : si $A$ est une formule et si $x$ est une variable d'individu, $\forall x\ (A)$ et $\exists x\ (A)$ sont des formules.

### 1.4 Règles de dominance entre connecteurs

Les connecteurs sont classés du plus prioritaire au moins prioritaire :

- $\neg$
- $\wedge$, $\vee$
- $\to$, $\leftrightarrow$
- $\exists$, $\forall$

## 2 Calcul propositionnel

### 2.1 Définition

Ici, nous ne conservons des langages du premier ordre que :

- les variables propositionnelles (notées $p$, $q$, $r$, etc.) ;
- les connecteurs ;
- les parenthèses.

Exemple : $\neg p \vee \neg(\neg q \to r)$.

### 2.2 Tables de valeurs de vérité

Nous codons « vrai » par 1 et « faux » par 0. Nous calculons, dans une table de vérité, les valeurs de vérité d'une formule, en fonction des valeurs des variables propositionnelles (cf. figure 1).

| $p$ | $q$ | $\neg p$ | $p \wedge q$ | $p \vee q$ | $p \to q$ | $p \leftrightarrow q$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |

FIG. 1: Table de vérité des connecteurs

$p \wedge q$ est vrai si et seulement si $p$ est vrai et si $q$ est vrai.
« $p$ vrai implique $q$ vrai » est équivalent à $q \vee (\neg p)$.
*Valuation* : fonction qui assigne à chaque variable $p$ une valeur $v(p)$ égale à 0 ou 1.

### 2.3 Analyse de vérité d'une formule

Il s'agit simplement de calculer la valeur de vérité de la formule.

$$\neg(p \vee q \rightarrow p) \wedge q$$

| $p = 0$ | | $p = 1$ |
|---|---|---|
| $\neg(0 \vee q \rightarrow 0) \wedge q$ | | $\neg(1 \vee q \rightarrow 1) \wedge q$ |
| $\neg(q \rightarrow 0) \wedge q$ | | $\neg(1 \rightarrow 1) \wedge q$ |
| $q = 0$ | $q = 1$ | $\neg(1) \wedge q$ |
| $\neg(0 \rightarrow 0) \wedge 0$ | $\neg(1 \rightarrow 0) \wedge 1$ | $\neg(1) \wedge q$ |
| $\neg(1) \wedge 0$ | $\neg(0) \wedge 1$ | $0 \wedge q$ |
| $0 \wedge 0$ | $1 \wedge 1$ | $0$ |
| $0$ | $1$ | $0$ |

Formule toujours fausse : *antilogie*. Exemple : $q \wedge \neg q$.
Formule toujours vraie : *tautologie*. Exemple : $q \vee \neg q$.

## 3 Méthode de résolution

### 3.1 Vocabulaire

- *Littéral* : variable propositionnelle ou sa négation. Si $u$ est un littéral, $\bar{u} = \neg u$, $u$ et $\bar{u}$ sont dits complémentaires.
- *Clause* : disjonction de littéraux : $p \vee q \vee \neg r$.
- *Forme clausale* : Forme Normale Conjonctive, ou conjonction de clauses (conjonction de disjonctions) : $(p \vee q \vee \neg r) \wedge (p \vee \neg q)$.

### 3.2 Règle de résolution

Soient deux clauses $C_1$ et $C_2$. Soit $u$ un littéral tel que $u \in C_1$ et $\bar{u} \in C_2$ Alors :

$$\frac{C_1, C_2}{C_1 \vee C_2 \setminus \{u, \bar{u}\}}$$

où $C_1, C_2$ signifie $C_1 \wedge C_2$. Autrement dit, nous remplaçons l'expression $C_1 \wedge C_2$ par l'expression $C_1 \vee C_2 \setminus \{u, \bar{u}\}$.

#### Explications

$C_1 = C_1' \vee u$ et $C_2 = C_2' \vee \bar{u}$.

Alors $C_1 \wedge C_2 \mapsto C_1' \vee C_2'$. $C_1' \vee C_2'$ est la *résolvante* de $C_1$ et $C_2$. Toute valuation, tout modèle, qui valide $C_1 \wedge C_2$ valide $C_1' \vee C'2$.

#### Démonstration

1. Montrons tout d'abord l'implication : $v(C_1 \wedge C_2) = 1 \Rightarrow v(C_1' \vee C_2') = 1$.
   Soit $v$ une valuation qui valide $C_1 \wedge C_2$. Si $v(C_1 \wedge C_2) = 1$, alors $v(C_1) = 1$ et $v(C_2) = 1$. $u$ peut prendre deux valeurs, et nous avons donc deux cas à considérer.
   (a) $v(u) = 1$ et $v(\bar{u}) = 0$. Or $v(C_2) = 1$ et $C_2 = C_2' \vee \bar{u}$. Donc, $v(\bar{u}) = 0$ implique $v(C_2') = 1$. Donc $v(C_1' \vee C_2') = 1$.
   (b) $v(u) = 0$ et $v(\bar{u}) = 1$. Or $v(C_1) = 1$ et $C_1 = C_1' \vee u$. Donc, $v(u) = 0$ implique $v(C_1') = 1$. Donc $v(C_1' \vee C_2') = 1$.

2. Nous montrons maintenant l'implication : $v(C_1' \vee C_2') = 1 \Rightarrow v(C_1 \wedge C_2) = 1$.
   Soit $v$ une valuation qui valide $C_1' \vee C_2'$. On veut en déduire une valuation $v'$ qui valide $C_1 \wedge C_2$. Nous avons encore deux cas à considérer, suivant que $v(C_1')$ vaut 1 ou 0.
   (a) $v(C_1') = 1$. On pose alors $v' = v$ avec $v'(\bar{u}) = 1$.
   $v'(C_1) = v'(C_1' \vee u) = v'(C_1') \vee v'(u) = 1 \vee 0 = 1$.
   $v'(C_2) = v'(C_2' \vee \bar{u}) = v'(C_2') \vee 1 = 1$.
   (b) $v(C_1') = 0$, alors $v(C_2') = 1$. On pose alors $v' = v$ avec $v'(u) = 1$.
   $v'(C_1) = v'(C_1' \vee u) = v'(C_1') \vee v'(u) = 0 \vee 1 = 1$.
   $v'(C_2) = v'(C_2' \vee \bar{u}) = v'(C_2') \vee v'(\bar{u}) = 1 \vee 0 = 1$.

### 3.3 Preuve par résolution

Soit $\Sigma$ un ensemble de clauses. Nous appelons preuve par résolution à partir de $\Sigma$ une suite de clauses $C_1, ...C_n$ telle que :

- soit $C_i \in \Sigma$ ;
- soit $C_i$ est obtenue à partir de deux clauses antérieures et de la règle de résolution.

Le dernier élément de la suite de clauses s'appelle le *but* ou la conclusion. Si le but est la clause vide, l'ensemble de clauses est inconsistant.

### 3.4 Exemple : arbre de résolution/réfutation

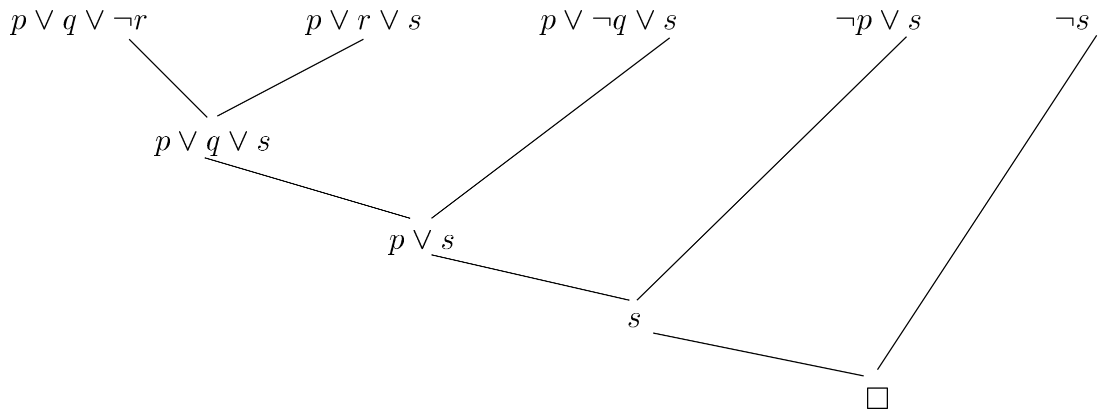


FIG. 2: Exemple d'arbre de réfutation

$\square$ : clause vide ou réfutation.

### 3.5 Application à la programmation logique

**Clause de Horn.** Une clause de Horn est une clause qui a au plus un littéral positif. Exemple : $\neg p \vee q \vee \neg r$. Une clause de Horn avec un littéral positif est dite *positive*. Une clause de Horn dont les littéraux sont négatifs est dite *négative*.

**Lien entre clauses de Horn et programmation logique.**

$$\begin{aligned} \neg p \vee \neq q \vee \neg r \vee s \quad &\Leftrightarrow \\ \neg(p \vee q \vee r) \vee s \quad &\Leftrightarrow \\ (p \wedge q \wedge r) \rightarrow s \end{aligned}$$

La dernière équation est une assertion logique typique : si les conditions $p$, $q$ et $r$ sont vérifiées, alors la propriété $s$ est vraie.

Soit un ensemble de propriétés logiques $C_1, ..., C_n$, $C_i$ étant une clause de Horn. A-t-on le résultat $B$ ?

Question : la formule $C_1 \wedge ... \wedge C_n \to B$ est-elle vraie ? Autrement dit, la formule $B \vee \neg(C_1 \wedge ... \wedge C_n)$ est-elle vraie, c'est-à-dire, est-elle une tautologie ?

**Preuve par réfutation :** nous supposons la formule $\neg(B \vee \neg(C_1 \wedge ... \wedge C_n))$ vraie et nous montrons l'inconsistance. Or $\neg(B \vee \neg(C_1 \wedge ... \wedge C_n)) = \neg B \wedge \neg\neg(C_1 \wedge ... \wedge C_n)) = \neg B \wedge C_1 \wedge ... \wedge C_n$. Notre problème revient donc à montrer l'inconsistance de l'ensemble des clause $\neg B, C_1, ..., C_n$.

**Exemple.** On suppose que l'on a les propriétés suivantes : $C_1 : p$, $C_2 : p- > q$ et $C_3 : q- > r$. La question est : « a-t-on $r$ ? » (la formule $r$ est-elle vraie).

Commençons par transformer nos propriétés en clauses de Horn.

$$C_1 \quad p \tag{1}$$
$$C_2 \quad q \vee \neg p \tag{2}$$
$$C_3 \quad r \vee \neg q \tag{3}$$
$$\text{But} \quad r \tag{4}$$

Nous raisonnons par réfutation, c'est-à-dire par l'absurde. Soit $E = p \wedge (q \vee \neg p) \wedge (r \vee \neg q) \wedge \neg r$. Nous sommes donc à la recherche d'une instanciation de $p$, $q$ et $r$ telle que la formule $E$ soit vraie. Nous voulons en fait montrer qu'une telle instanciation n'existe pas.

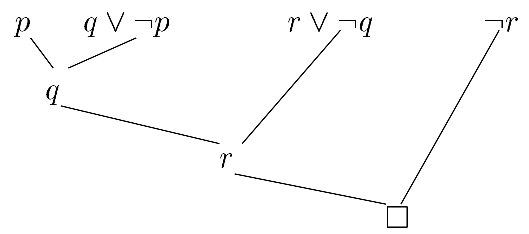


FIG. 3: Preuve de la formule $(p \wedge (p \to q) \wedge (q \to r)) \to r$.

Nous aboutissons sur la clause vide, il n'existe donc aps d'instanciation de $p$, $q$ et $r$ qui permette de valider $E$. Donc la formule $E$ est toujours fausse et la formule $\neg E$ est toujours vraie. Donc $r$ est vraie à chaque fois que les clauses $C_1$, $C_2$ et $C_3$ le sont. Nous avons donc prouvé la formule : $(p \wedge (p \to q) \wedge (q \to r)) \to r$.

## 3.6 Complétude

Si $\Sigma$ est un ensemble de clauses inconsistant, alors il existe un arbre de réfutation de $\Sigma$. On peut donc prouver une formule qui est vraie !