Programme

La Logique = Étude des Raisonnements Valides

Une logique = un système formel permettant de réaliser des raisonnements valides

- Introduction
- Le langage de la LP (syntaxe)
- La sémantique de la LP
- Équivalence logique et Substitution
- Conséquence logique
- Formes normales et clausale
- Méthode de résolution
- Méthode des tableaux
- Méthode de Davis et Putnam
- Initiation à la logique des prédicats

Programme

- **Introduction**
- Le langage de la LP (syntaxe)
- La sémantique de la LP
- Équivalence logique et Substitution
- Conséquence logique
- Formes normales et clausale
- Méthode de résolution
- Méthode des tableaux
- Méthode de Davis et Putnam
- Initiation à la logique des prédicats

Introduction : Exemple

« Si le prévenu a commis le vol, c'est que ce vol a été minutieusement préparé, ou alors le prévenu avait un complice.

Si le vol a été minutieusement préparé, alors, si le prévenu avait un complice, un butin plus important eût été emporté.

Or, le butin n'a pas été important.

Donc, le prévenu n'a pas commis le vol. »

- Question 1 : cette argumentation est-elle convaincante ?
- Question 2 : cette argumentation (à défaut d'être convaincante) est-elle valide ?

Introduction : Exemple

- Langage symbolique
  - p = le prévenu a commis le vol
  - q = le vol a été minutieusement préparé
  - r = le prévenu avait un complice
  - s = le butin a été important

« Si le prévenu a commis le vol, c'est que ce vol a été minutieusement préparé, ou alors le prévenu avait un complice.

Si le vol a été minutieusement préparé, alors, si le prévenu avait un complice, un butin plus important eût été emporté.

Or, le butin n'a pas été important.

Donc, le prévenu n'a pas commis le vol. »

Introduction : Exemple

$H1 = p \rightarrow (q \vee r)$

$H2 = q \rightarrow (r \rightarrow s)$

$H3 = \neg s$

$C = \neg p$

Validité = *C est-elle conséquence logique de H1, H2 et H3 ?*

noté *H1, H2, H3 |= C ?*

ⓘ la validité d'un raisonnement ne dépend que des relations entre ses arguments

p = l'âne a chanté un air
q = l'air était en sol mineur
r = l'âne a vu un singe
s = la foudre est tombée

Introduction: Origines Philosophiques

- Aristote (4^{e} siècle av JC) : « Art de la Pensée »

  Logique formelle en langage naturel !
  - Conception : formation des concepts
    - Termes généraux vs. termes singuliers
  - Jugement : activité qui consiste à affirmer ou nier
    - Notion de proposition
  - Raisonnement : production de nouvelles propositions à partir de propositions préalablement affirmées ou niées de manière logique
    - Notion d'inférence (raisonnement valide élémentaire)

Liens entre termes à l'intérieur d'une proposition et non liens entre propositions

Introduction: origines mathématiques

- Boole (19^{e} s) assimile le raisonnement logique à un calcul
  - Algèbre de Boole
- Frege (19^{e} s) introduit l'idée d'une langue artificielle pour se débarrasser de l'ambiguïté du langage naturel
  - Introduction des quantificateurs (avec Peirce)
- Russel (20^{e} s)
  - Paradoxes : Langage Objet vs. Métalangage
  - Calcul des Prédicats

Introduction: fondateurs de la logique moderne (20^e s)

- Tarski
  - Théorie de la vérité (sémantique de la logique)
- Church, Turing, Goedel
  - Systèmes de règles de production

Introduction : logique et informatique?

- Base du fonctionnement de nos machines
- Représentation des propriétés d'un système, conditions de réalisation d'une action
- Expression de la validité d'un programme
- Compréhension et écriture des preuves
- Base de l'intelligence artificielle

Langages Formels

- Alphabet
  - Ensemble de symboles utilisables *A={a,b,c}*
- Mot ou expression
  - Une suite d'élément de l'alphabet *aaba*
- Langage
  - A* ou un de ses sous-ensemble
  - *A*={ε,a,b,c,aa,ab,ac,ba,bb…. cccabccaaa,…}*
  - *L1={ε,ab,aabb,aaabbb…} les mots ayant autant de a que de b, les a précédents les b*

Langages Formels

- Outils de définition d'un langage
  – expression régulière, automate, grammaire, **définition par induction structurelle**
- Règles de construction *r* (règles de production)
  – Des fonctions partielles définies sur A*
    - Données : soit n éléments $m_1,\ldots,m_n$ de A* (vérifiant parfois des conditions spécifiques). *n* est appelée l'arité de la fonction
    - Résultat : un nouvel élément de A*, noté $r(m_1,\ldots,m_n)$, défini à partir des données, de A et de l'opérateur de concaténation, noté .

Ex. r1: $A^* \rightarrow A^*$
$m \mapsto r1(m)=a.m.b$

r2: $A^* \times A^* \rightarrow A^*$
$m_1,m_2 \mapsto r2(m_1,m_2) = m_1.b.m_2$

Définition de langages par induction structurelle

Soit un alphabet A, soit un sous-ensemble B de A* appelé la base et un ensemble R de règles de constructions sur A*.

On définit un langage $L \subseteq A^*$ par induction comme le plus petit ensemble tel que :

(**base**) L contient B

(**cons**) pour toute $r \in R$ et tout $m_1, \ldots m_n \in L$ (où n est l'arité de r), si $r(m_1, \ldots m_n)$ est défini alors $r(m_1, \ldots m_n) \in L$

Définition inductive d'une fonction sur les mots d'un langage défini par induction

- Soit L un langage défini par induction avec (B,R)

On définit une fonction f sur les mots de L par induction ainsi :

(base) pour tout mot m de B, **on fixe la valeur f(m)**

(cons) et pour tout $r \in R$ et tout $m_1, \ldots m_n \in L$ (si $r(m_1, \ldots m_n)$ est défini),

on définit $f(r(m_1, \ldots m_n))$ en fonction de $f(m_1), \ldots f(m_n)$

Preuve par induction structurelle

- Soit L un langage défini par induction avec (B,R) et P une propriété :

Prouver que tous les mots de L vérifient P par induction structurelle c'est :

(base) pour tout mot m de B, **prouver que m vérifie P**

(cons) et pour tout r de R et tout $m_1,\ldots m_n \in L$ (si $r(m_1,\ldots m_n)$ est défini), **prouver que** :

si $m_1,\ldots m_n$ vérifient P alors $r(m_1,\ldots m_n)$ vérifie P