Logique - EISTI - ING 2

Yannick Le Nir

Ecole Internationale des Sciences du Traitement de l'Information

yannick.lenir@eisti.fr

Sommaire

Avant Propos

Documents annexes

L'ensemble des documents liés au cours de Logique se trouve sur le site `http ://sifoci.eisti.fr`.
Vous y trouverez le polycopié de Chrysostome Baskiotis dont est issue cette présentation, ainsi qu'une bibliographie, le planning et le programme détaillé de ce cours.

Intelligence artificielle

Contenu du module

- Théorie des langages
- Logique computationnelle
- Prolog
- Décidabilité
- Intelligence artificielle
- Systèmes experts

Options

- Options IDSI, ISICO, ISIN, DESI.
- Toutes les autres options dans différents proportions

Connaissance

Définition de l'IA

Connaissance de la connaissance. Résolution de problèmes par modélisation de la connaissances.

Historique

- Descartes (XVI-XVII siecle) : modelisation de la connaissance
- Henri Bergson (1907) : intelligence, instinct et connaissance
- Gaston Bachelard (1938) : construction de la connaissance
- Gaston Berger (1941) : connaissance indéfinissable
- Hervé Zwirn (2002) : connaissance forte et faible (kolmogorov)

Contenu du cours

Eléments traités

1. La logique comme langage pour le traitement de la connaissance
2. La logique du premier ordre
3. Des algorithmes pour l'inférence logique
4. Des applications pratiques

Sommaire

Intelligence

Définition

Ensemble des fonctions mentales qui permettent à un être vivant d'adapter son comportement à son environnement

Fonctions du système nerveux

- perception de l'environnement
- reconnaissance des situations
- décision d'actions
- mémorisation

Intelligence artificielle

Objectif

Etant donnée une tâche précise, dont l'exécution par l'homme requiert de l'intelligence, ils'agit de construire un logiciel permettant à un ordinateur d'exécuter la même tache avec des résultats comparables à ceux obtenus par l'homme.

Méthode

- Il existe un algo qui résoud le problème (cas non traité)
- Il n'existe pas d'algorithme de résolution
  - recherche d'une stratégie
  - ensembles de règles pour se rapprocher de la solution (heuristique)

Méthode heuristique

Postulat

Toute tâche cognitive peut être accomplie par un ordinateur programmé heuristiquement.

Représentation des connaissances (stockage et recherche)

- organisation
- extraction
- modification
- performances

Données symboliques

Contraintes

- Différentes des données numériques classiques
- Utilisées dans des logiciels non algorithmiques car non explicites (variations de la connaissance, choix, inconnues, ...)
- Nécessitent des règles de gestions d'informations symboliques

Résolution d'un problème

Etapes

1. Acquisition des informations (recherche)
2. Représentation des connaissances (IA)

Objectif

- Forme utilisable par un ordinateur
- Donc, langage de description :
  - syntaxe : correction
  - sémantique : sens

Langage pour la connaissance

Propriétés souhaitées

- Concision
- Non ambigu
- Indépendant du contexte
- Formalisme

Logique mathématique

- Pour modéliser et stocker la connaissance
- Comme langage de communication avec l'ordinateur

Sommaire

Objectifs de la logique

Représenter la connaissance

- Forme descriptive
- Propositions déclaratives

Analyser et évaluer les propositions

- Attention aux ambiguités : "Tout homme aime une femme"
- Problème des quantificateurs
- Réduire la complexité du langage en restant complet
- Démarche d'abstraction

Formalisation

Noms

- Noms propres (attention au sens dénotationnel)
- Noms communs (attention au sens figuré)
  - extension : dénotation (ensemble des objets désignés)
  - intension : sens (fonction permettant de décider l'équivalence)

Notations

- Noms propres : constantes a,b,c,...
- Noms communs ou noms propres à plusieurs dénotations : variables X,Y,Z,...

Formalisation (suite)

Qualité

Fonction permettant de qualifier certaines valeurs, appelé *foncteur*.
Exemple : "rouge" est la valeur de qualité "couleur" appliqué à "livre"

Propriété

Fonction booléenne sur un objet, appelé *prédicat* ou *fonction propositionnelle*.

Proposition

Assemblage de mots exprimant une pensée : ensemble de mots auquel on associe une fonction de vérité (vrai ou faux).
Noté p,q,...(variable propositionnelle)

Analyse et Evaluation

Raisonnement déductif

Tous les A sont des B
C est A.
Donc, C est B.

Aristote

Tous les êtres vivant sur Terre sont mortels.
Socrate est un être vivant sur Terre.
Donc Socrate est un mortel.

Analyse et Evaluation

Raisonnement inductif

$A_1, A_2, \cdots, A_n$ sont B.
$A_1, A_2, \cdots, A_n$ sont C.
Donc tous B est C.

Bacon

Ces êtres vivants (Socrate, Héraclite, Parménide,...) sont sur la Terre.
Ces êtres vivants (Socrate, Héraclite, Parménide,...) sont mortels.
Donc, tous les êtres vivants qui sont sur la Terre sont mortels.

Analyse et Evaluation

Raisonnement abductif

Tous les A qui sont B sont aussi C.
D est C.
Donc, D est B.

Peirce

Tous les êtres vivants qui sont sur la Terre sont mortels.
Ces êtres vivants (Socrate, Héraclite, Parménide,...) sont mortels.
Donc, ces êtres vivants (Socrate, Héraclite, Parménide,...) sont sur la Terre.

Analyse et Evaluation

Raisonnement par analogie

A est P.	A est P.
B est similaire à A.	A et B sont S.
Donc, B est P.	Donc, B est P.

Jurisprudence

Socrate est un être vivant sur Terre et qui est mortel.
Héraclite est un être vivant sur Terre et qui est comme Socrate.
Donc, Héraclite est mortel.

Logiques formelle et computationnelle

Raisonner correctement $\mapsto$ Ecrire des programmes correctes

La logique formelle est une version décontextualisée de la Logique :

- un langage formel ;
- une syntaxe sans ambiguïtés ;
- une sémantique précise, et
- des règles de formation des propositions.

La logique computationnelle permet d'automatiser l'application des différents types de raisonnements sur les représentations formelle des connaissances :

- syntaxe et sémantique des raisonnements (aspect formel)
- efficacité du raisonnement (aspect computationnel)

Sommaire

Eléments du langage

Langage formel $\mathcal{L}_0$

Les propositions seront représentées par des symboles de valeur de vérité vraie ou fausse :

- Ensemble V_p, au plus dénombrable, des propositions notées $p, q, \ldots$.
- Ensemble Ξ, au plus dénombrable, des constantes.
- Ensemble L des connecteurs :
  - logique unaires : la négation $\neg$
  - propositionnels binaires :
    - disjonction : $\vee$
    - conjonction : $\wedge$
    - implication : $\rightarrow$
    - équivalence : $\leftrightarrow$
- Séparateurs : parenthèses (,) et crochets [,].

Eléments du langage

Définition

Un atome est un proposition dont la structure interne ne nous préoccupe pas. *Notation :* $p, q, r, \ldots$

Définition

Une formule bien formée (fbf) :

- un atome
- proposition obtenue à partir des fbf A et B :
  - $\not A$
  - $A \vee B$, $A \wedge B$
  - $A \rightarrow B$, $A \leftrightarrow B$

Eléments du langage

Notations

- $A_0 = \{V_p, \Xi, L\}$ (alphabet du langage)
- $F_0 = A, B, C, \ldots$ (ensemble des fbf)
- $\mathcal{L}_0 = \{A_0, F_O\}$ (langage d'ordre 0 : langage du calcul propositionnel)

Connecteurs de $\mathcal{L}_0$

Propriétés

1. $p \leftrightarrow \neg\neg p$ (involution)
2. Loi de de Morgan :
   $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$
   $\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$
3. $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ (implication)
4. $p \rightarrow (q \rightarrow p)$ (introduction implication)
5. $(p \rightarrow (q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$ (distributivite)
6. $(p \rightarrow q) \rightarrow ((p \rightarrow \neg q) \rightarrow \neg p)$ (contradiction)

Connecteurs de $\mathcal{L}_0$

Propriétés déduites

Soit $P_1 : p \longrightarrow q$

- $P_2 : \neg q \longrightarrow \neg p$ (contrapositive de P_1)
- $P_3 : q \longrightarrow p$ (inverse de P_1)
- $P_4 : \neg(p \longrightarrow p)$ (négation de P_1)

Interprétation sémantique

Valeur de vérité

Donner un sens aux (fbf) : Tout atome peut prendre deux valeurs : vrai (1) et faux (0) et la valeur de vérité d'une fbf est complètement déterminée par la valeur de chacun de ses atomes.

Table de vérité de $\mathcal{L}_0$

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Interprétation

Définition

Soit Δ un ensemble de variables propositionnelles.
$\phi : \Delta \rightarrow \{0,1\}$ est une *valuation* de Δ.
Si A est une fbf, Δ_A est l'ensemble des variables propositionnelles de A et $\phi(\Delta_A)$ est une valuation de A appartenant à $\{0,1\}^n$ $(cardA = n)$.

Définition

Soit A une fbf.
Une *interprétation* I de A est une valuation de Δ_A : i.e. $\phi_I : \Delta_A \rightarrow \{0,1\}$, notée $\phi_I(\Delta_A)$ ou $\phi_I(A)$, appartenant à $\{0,1\}^{cardA}$

Satisfaisabilité

Définition

Soit A une fbf et I une interprétation. A est *satisfiable* par I, notée $I \models A$, si on a une des situations suivantes :

- si $A \in V_p$, alors $I \models A$ ssi $\phi_I(A) = 1$
- si A est de la forme $(\neg B)$, alors $I \models A$ ssi $I \not\models B$
- si A est de la forme $(B \wedge C)$, alors $I \models A$ ssi $I \models B$ et $I \models C$
- si A est de la forme $(B \vee C)$, alors $I \models A$ ssi $I \models B$ ou $I \models C$
- si A est de la forme $(B \rightarrow C)$, alors $I \models A$ ssi soit $I \not\models B$, soit $I \models C$
- si A est de la forme $(B \leftrightarrow C)$, alors $I \models A$ ssi soit $I \not\models B$ et $I \not\models C$, soit $I \models B$ et $I \models C$

Modèle

Définition

Soit A une fbf et I une interprétation. Si $I \models A$, alors I est un *modèle* pour A, noté $M(A)$. On dit que A est vraie dans I. On note par $\mathcal{M}$ un ensemble de modèles. $\forall I \in \mathcal{M}$, on a $I \models A$, d'où $\mathcal{M} \models A$.

Extension

Soit F un ensemble de fbf. I est un modèle pour F si I est un modèle pour chaque fbf de F.

Propriétés des interprétations

Définitions

- une fbf A vraie pour toute interprétation est une *tautologie*, notée $\models A$.
- une fbf A fausse pour toute interprétation est *sémantiquement inconsistante* ou *insatisfiable*.
- une fbf pour laquelle il y a au moins une interprétation qui la satisfait est *sémantiquement consistante* ou *satisfiable*.
- une fbf est *complète* ssi elle a un modèle exactement.
- un ensemble de fbf est *mutuellement exclusif* ssi chaque interprétation satisfait au plus à une fbf.
- un ensemble de fbf est *mutuellement exhaustif* ssi chaque interprétation satisfait au moins à une fbf.

Equivalence

Définition

Deux fbf Aet B sont *équivalentes*, noté $A \equiv B$ ssi la fbf $A \leftrightarrow B$ est une tautologie.

Définition

Soit une fbf B dont les atomes appartiennent a l'ensemble des atomes de $A_1, A_2, \cdots, A_n$. B est une *conséquence valide* des $A_1, A_2, ..., A_n$, noté $A_1, A_2, ..., A_n \models B$ si B prend la valeur 1 quand tous les A_i sont simultanéments à 1.
On notera $A \models B$ ssi tout modèle de A est aussi un modèle de $B\cdot$

Théorème

$A \models B$ est équivalent à $\models (A \rightarrow B)$.

Modèles et connaissance

Propriétés

Soit A, $A_1, \cdots, A_n$ et B des fbf :

- $\mathcal{M}(A \wedge B) = \mathcal{M}(A) \cap \mathcal{M}(B)$
- $\mathcal{M}(A \vee B) = \mathcal{M}(A) \cup \mathcal{M}(B)$
- $\mathcal{M}(\neg A) = \mathcal{M}(A)^C$
- A est satisfiable ssi $\mathcal{M}(A) \neq 0$
- $A \rightarrow B$ ssi $\mathcal{M}(A) \subset \mathcal{M}(B)$
- A est équivalente à B ssi $\mathcal{M}(A) = \mathcal{M}(B)$
- $A_1, \cdots, A_n$ sont mutuellement exclusives ssi $\forall i \neq j$, $\mathcal{M}(A_i) \cap \mathcal{M}(A_j) = 0$.

Modèles et connaissance

Théorème
Si $\mathcal{M}' \subseteq \mathcal{M}$ et $\mathcal{M} \models A$ alors $\mathcal{M}' \models A$

Evaluation syntaxique

Définition

Une fbf A est un *théorème*, noté $\vdash A$, si A est un axiome ou si A est obtenue par application des règles d'inférence sur d'autres théorèmes.

Axiomes du calcul propositionnel

Introduction : $A \rightarrow (B \rightarrow A)$
Négation : $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
Distributivité : $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

Evaluation syntaxique

Règles d'inférence

Modus Ponens : $\frac{\vdash A, \vdash A \rightarrow B}{\vdash B}$	Modus tollens : $\frac{\vdash A \rightarrow B, \vdash \neg B}{\vdash \neg A}$
Elimination de ET : $\frac{\vdash A \wedge B}{\vdash A, \vdash B}$	Introduction de ET : $\frac{\vdash A, \vdash B}{\vdash A \wedge B}$
Introduction de OU : $\frac{\vdash B}{\vdash A \vee B \vee C \cdots}$	Involution : $\frac{\vdash \neg\neg A}{\vdash A}$
Double résolution : $\frac{\vdash A \vee B, \vdash \neg B \vee C}{\vdash A \vee C}$	

Démonstration

Définition

Soit un théorème A. Une *démonstration* de A est une suite finie $(A_1, A_2, \cdots, A_n, A)$ où chaque A_i est soit un axiome, soit le résultat d'une règle d'inférence appliquée sur des éléments A_j précédemment obtenus $(j < i)$.

Définition

Une fbf A est une *déduction* de l'ensemble de fbf $B_1, B_2, \cdots, B_n$, noté $B_1, B_2, \cdots, B_n \vdash A$ si il existe une suite finie $(A_1, A_2, \cdots, A_n)$ où chaque A_i est soit un axiome, soit un des B_i, soit il est obtenu par application d'une règle d'inférence sur des éléments A_j précédemment obtenus. Les fbf B_i sont appelées des *hypothèses*.

Equivalence modèle/démonstration

Définition

- Une logique est *adéquate* si tout théorème $\vdash A$ est une formule valide $\models A$.
- Une logique est *syntaxiquement consistante* s'il n'existe aucune formule du langage telle que $\vdash A$ et $\vdash \neg A$
- Une logique est *(faiblement) complète* si toute formule valide est un théorème, i.e. $(\models A) \rightarrow (\vdash A)$

Théorèmes

- Le calcul propositionnel est adéquat
- Le calcul propositionnel est syntaxiquement consistant
- Le calcul propositionnel est (faiblement) complet

Substitution

Définition

Une *substitution* est un ensemble fini de couples de la forme $(x_1/A_1, x_2/A_2, \cdots, x_n/A_n)$, où chaque A_i est une fbf et chaque x_i une variable telle que $x_i \neq A_i$ et $A_i \neq A_j$ si $i \neq j$. La substitution vide sera notée par ε.

Définition

Soient deux substitutions $\sigma = (x_1/A_1, x_2/A_2, \cdots, x_n/A_n)$ et $\tau = (y_1/B_1, y_2/B_2, \cdots, y_m/B_m)$. La composition $\sigma\tau$ est obtenue depuis
$\{(x_1/A_1\tau, x_2/A_2\tau, \cdots, x_n/A_n\tau, y_1/B_1, y_2/B_2, \cdots, y_m/B_m\}$
en supprimant :

- tous les $x_i/A_i\tau$ avec $x_i = A_i\tau, \ i = 1, \cdots, n$
- tous les y_j/B_j avec $y_j \in \{x_1, x_2, \cdots, x_n\}, \ j = 1, \cdots, m$

Substitution

Propriétés

- $E(\sigma\tau) = (E\sigma)\tau$
- $(\rho\sigma)\tau = \rho(\sigma\tau)$
- $\varepsilon\theta = \theta\varepsilon = \theta$
- Si $\models E$, alors $\models E\theta$

Sommaire

Introduction

Calcul des prédicats

- Représentation formelle de la connaissance
- Objets constituant l'univers du discours
- Limitations d'expressivité du calcul propositionnel
- Fonctions propositionnelles logiques dont le résultat est :
  - valeur de vérité : *Prédicats*
  - objet du discours : *Foncteurs*

Eléments du langage

Définition

Langage $\mathcal{L}_1$ d'alphabet $A_1 = \{\mathbb{V}, \Xi, \mathbb{F}, \mathbb{P}, \mathbb{L}\}$

- Ensemble **V** des variables $X, Y, \cdots$
- Ensemble Ξ des constantes $a, b, \cdots$
- Ensemble **F** de fonctions $\mathrm{f} : \mathbf{V} x \Xi \rightarrow \mathbf{V} \cup \Xi$ d'arité quelconque, notés $f, g, \cdots$ ou $F, G, \cdots$, appelés *foncteurs*.
- Un ensemble **P** de fonctions d'arité quelconque $f : \mathbf{V} x \Xi \rightarrow \{Vrai, Faux\}$, notés $p, q, \cdots$ ou $P, Q, \cdots$, appelés *prédicats*.
- Un ensemble **L** de connecteurs et quantificateurs :
  - connecteur logique unaire : $\neg$
  - connecteurs propositionnels binaires : $\wedge, \vee, \rightarrow, \leftrightarrow$
  - quantificateurs : existentiel $\exists$, universel $\forall$

Terme

Définition

Un *Terme* est récursivement défini par :

- une constante est un terme
- une variable est un terme
- si f est foncteur d'arité n et $t_1, t_2, \cdots, t_n$ sont des termes, alors $f(t_1, t_2, \cdots, t_n)$ est un terme.

Ensemble des termes noté $\mathbb{T}$, obtenu par application des règles précédentes un nombre fini de fois.

Formules bien formées calcul des prédicats

Définition

L'ensemble $\mathbb{F}$ des formules bien formées est le plus petit ensemble tel que :

- Si p est un prédicat d'arité n et $t_1, t_2, \cdots, t_n$ sont des termes, alors $p(t_1, t_2, \cdots, t_n) \in \mathbb{F}$
- Si $F, G \in \mathbb{F}$ alors
  - $\neg F$
  - $F \vee G, F \wedge G$
  - $F \rightarrow G, F \leftrightarrow G$

  sont aussi des *fbf*
- Si $F \in \mathbb{F}$ et X est une variable, alors $(\forall X)F \in \mathbb{F}$ et $(\exists X)F \in \mathbb{F}$

Propriétés du calcul des prédicats

Définitions

1. Si p est un prédicat d'arité n et $t_1, t_2, \cdots, t_n$ sont des termes alors $p(t_1, t_2, \cdots, t_n)$ est une *formule atomique*
2. Le triplet $\mathcal{L}_1 = \{A_1, \mathbb{T}, \mathbb{F}\}$ est le langage d'ordre un ou le langage du calcul des prédicats.
3. La *portée* d'un quantificateur est la partie de la formule qui se trouve sous l'influence du quantificateur.
   - Une variable est soit liée (sous l'influence d'un quantificateur), soit libre.
   - Une formule sans variables libres est une *formule close*
   - Une formule sans quantificateurs est une *formule ouverte*
4. Soit f une *fbf* avec variables libres $x_1, x_2, \cdots, x_n$
   La cloture universelle (resp. existentielle) de F, $\forall F$ (resp. $\exists F$) est la formule close $\forall x_1, \forall x_2, \cdots, \forall x_n F$ (resp. $\exists x_1, \cdots F$)

Interprétation

Définitions

1. Le *domaine de l'interprétation* couvre une partie seulement de l'univers du discours : $\mathbb{D}(I) \subset \mathcal{U}$
2. Une *interprétation* I sur un langage $\mathcal{L}$ est un domaine d'interprétation non vide $\mathbb{D}(I)$ et une application qui associe :
   - chaque constante $a \in \Xi$ avec un élément $a_I \in \mathbb{D}(I)$
   - chaque foncteur $f \in \mathbf{F}$ d'arité n avec une fonction $f_I : \mathbb{D}(I)^n \longrightarrow \mathbb{D}(I)$
   - chaque prédicat $p \in \mathbf{P}$ d'arité n avec une relation $p_I \subseteq \mathbb{D}(I)^n$

Sémantique

Définitions

1. On appelle *assignation* d'un ensemble de variables $W \subset V$ relativement à une interprétation I, une application $\bar{\varphi}_I : W \to \mathbb{D}(I)$
2. La *signification* φ_I d'un ensemble de termes $\mathbb{T}$ relativement à une interprétation I, est définie par :
   - si $t \in \mathbb{T}$ est une constante alors $\varphi_I(t) = t_I$
   - si $t \in \mathbb{T}$ est une variable alors $\varphi_I(t) = \bar{\varphi}_I(t)$
   - si $t \in \mathbb{T}$ est un terme de la forme $f(t_1, t_2, \cdots, t_n)$ alors $\varphi_I(t) = f_I(\varphi_I(t_1), \varphi_I(t_2), \cdots, \varphi_I(t_n))$

Sémantique

Définition

La valeur de vérité de la signification d'une *fbf* relativement à une interprétation I est définie par :

- Si $F = p(t_1, t_2, \cdots, t_n)$ alors $I(F) = p_I(\varphi_I(t_1), \varphi_I(t_2), \cdots, \varphi_I(t_n))$.
- Si F est de la forme $\neg G, G \vee H, G \wedge H, G \rightarrow H, G \leftrightarrow H$ alors $I(F)$ est égale à la valeur de vérité de la forme correxpondante.
- Si $F = \forall x G(x, y, z, \cdots)$ alors $I(F) = 1$ si $\forall \sigma = (x/a)$ avec $a \in \mathbb{D}(I)$ nous avons $I(G\sigma) = 1$ (vraie). Sinon $I(F) = 0$ (fausse).
- Si $F = \exists x G(x, y, z, \cdots)$ alors $I(F) = 1$ si $\exists \sigma = (x/a)$ avec $a \in \mathbb{D}(I)$ nous avons $I(G\sigma) = 1$ (vraie). Sinon $I(F) = 0$ (fausse).

Satisfiabilité

Définitions

1. Une *fbf* F est *satisfiable* ou *sémantiquement consistante* s'il existe une interprétation I telle que la valeur de vérité de F par rapport à I est égale à 1. I est alors un modèle de F, noté $I \models F$.
2. Une *fbf* F qui est vraie pour toute interprétation est appelée *formule valide*, notée $\models F$
3. Soit B une *fbf* close et $A = \{A_1, A_2, \cdots, A_n\}$ un ensemble de *fbf* closes. B est une *conséquence sémantique* des $A_1, A_2, \cdots, A_n$, noté $A \models B$ ou $A_1, A_2, \cdots, A_n \models B$ si pour toute interprétation I telle que $I(A_i) = 1, \forall i = 1, \cdots, n$, on a $I(B) = 1$.

Satisfaisabilité

Théorème

Soient A un ensemble de *fbf* closes et B une *fbf* close. Alors $A \models B$ ssi $A \cup \{\neg B\}$ est insatisfiable

Théorèmes

Distributivité des quantificateurs :

1. $\models \forall x(\varphi \wedge \psi) \leftrightarrow \forall x\varphi \wedge \forall x\psi$
2. $\models \exists x(\varphi \vee \psi) \leftrightarrow \exists x\varphi \vee \exists x\psi$
3. $\models \forall x(\varphi(x) \vee \psi) \leftrightarrow \forall x\varphi(x) \vee \psi$ si x n'est pas une variable libre de ψ
4. $\models \exists x(\varphi(x) \wedge \psi) \leftrightarrow \exists x\varphi(x) \wedge \psi$ si x n'est pas une variable libre de ψ

Evaluation syntaxique

Principe

- Déduire le bien fondé d'une formule
- Théorie de la démonstration

Théorème

Une *fbf* est un *théorème*, et l'on note $\vdash A$, si A est un axiome ou si A est une formule obtenue par application des règles d'inférence sur d'autres théorèmes :

Règles d'inférences et Axiomes du calcul des prédicats

Règles d'inférences

Modus Ponens : $\frac{\vdash A, \vdash A \to B}{\vdash B}$ Modus tollens : $\frac{\vdash A \to B, \vdash \neg B}{\vdash \neg A}$
Elimination de ET : $\frac{\vdash A \wedge B}{\vdash A, \vdash B}$ Introduction de ET : $\frac{\vdash A, \vdash B}{\vdash A \wedge B}$
Généralisation : $\frac{\vdash A}{\vdash \forall x A}$
Exemplification universelle : $\frac{\vdash \forall x A}{\vdash A\sigma}, \sigma = \{x/a\}$
Exemplification existentielle : $\frac{\vdash \exists x A}{\vdash A\sigma}, \sigma = (x/s(x))$ avec $s(.)$: constante obtenue par application de s à x.

Règles d'inférences et Axiomes du calcul des prédicats

Axiomes

Introduction : $A \rightarrow (B \rightarrow A)$ Négation : $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
Distributivité : $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
Exemplification universelle : $\forall x A(x) \rightarrow A(c)$ Généralisation universelle : $((A \rightarrow B) \rightarrow (A \rightarrow \forall x B))$

Démonstration

Définition

Soit un théorème A. Une *démonstration* de A est une suite finie $(A_1, A_2, \cdots, A_n, A)$ où chaque A_i est soit un axiome, soit le résultat d'une règle d'inférence appliquée sur des éléments A_j précédemment obtenus.

Définition

Une fbf close A est une *déduction* de l'ensemble de fbf closes $B_1, B_2, \cdots, B_n$, noté $B_1, B_2, \cdots, B_n \vdash A$ si il existe une suite finie $(A_1, A_2, \cdots, A_n)$ où chaque A_i est soit un axiome, soit un des B_i, soit il est obtenu par application d'une règle d'inférence sur des éléments A_j précédemment obtenus. Les fbf B_i sont appelées des *hypothèses*.

Déduction et Théorie

Théorème

Soit A une *fbf* close. Si $A \vdash B$, alors, $\vdash A \rightarrow B$ et vice versa.

Définition

Une *Théorie* $\mathcal{T}$ est une collection des théorèmes avec la propriété $\mathcal{T} \vdash p \rightarrow p \in \mathcal{T}$

Définition

Soient $\mathcal{T}$ et $\mathcal{T}'$ deux théories sur les langages $\mathcal{L}$ et $\mathcal{L}'$ respectivement.

- $\mathcal{T}'$ est une *extension* de $\mathcal{T}$ si $\mathcal{T} \subseteq \mathcal{T}'$
- $\mathcal{T}'$ est une *extension conservative* de $\mathcal{T}$ si $\mathcal{T}' \cap \mathcal{L} = \mathcal{T}$, i.e. tous les théorèmes de $\mathcal{T}'$ dans le langage $\mathcal{L}$ sont aussi des théorèmes de $\mathcal{T}$

Consistance

Définition

Une logique est *syntaxiquement consistante* s'il n'existe aucune formule du langage telle que nous avons en même temps $\vdash A$ et $\neg \vdash A$.

Théorème

Le calcul des prédicats est syntaxiquement consistant

Introduction

Intelligence, connaissances et langages

Objectifs et méthodes de la logique

Calcul propositionnel
- Eléments du langage
- Interprétation sémantique - Modèles
- Evaluation syntaxique - Démonstration

Calcul des prédicats
- Introduction
- Interprétation
- Modèles

Unification

Réponse correcte à un programme

Modèles de herbrand. Unification

Introduction

Rappels : Clauses de Horn

- strictes (règles) : $p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow q$
- négatives (questions) : $p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow$
- positives (faits) : $\rightarrow q$

Utilisation

- Programme Prolog $\equiv$ ensemble de clauses de Horn
- Programme E prouve $\mathcal{A} \equiv E \cup \{\neg\mathcal{A}\}$ insatisfiable (i.e pas de modèle)
- Réduction de l'espace de recherche $->$ univers de Herbrand

Interprétation de Herbrand

Définition

Une *fbf* sans variables est appelée *fbf filtrée*. Une terme sans variables est appelé *terme filtré*. Une *formule atomique* ou atome sans variable est appelé *atome filtré*.

Objectif

Réduire le nombre d'interprétations possibles pour un ensemble donné de *fbf*.

Solution

Modifier notre façon de voir l'univers du discours : langage de la logique du premier ordre.

Interprétation de Herbrand

Programme défini

Soit $\mathcal{L} = (A, \mathbb{T}, \mathbb{F})$ un langage du premier ordre. Soit l'ensemble des *fbf* E ayant pour langage $\mathcal{L}$. Si E ne contient que des clauses sans éléments contradictoires et de la connaissance positive, alors E est un *programme défini*.

Interprétation de Herbrand

Modèles des Programmes définis

Tout programme défini contient au moins un modèle. Il est établit via l'univers et la base de Herbrand

Définition : Univers de Herbrand

Soit un ensemble de clauses E de langage $\mathcal{L}$. L'*univers de Herbrand* $\mathcal{U}_E$ de E est l'ensemble de tous les termes filtrés que l'on peut former en utilisant les constantes et les foncteurs de E. Si E ne contient pas de symbole de constante, on en ajoute un arbitrairement (a_H : constante de Herbrand).

Interprétation de Herbrand

Définition : Base de Herbrand

Soit un ensemble de clauses E de langage $\mathcal{L}$. La *base de Herbrand* $\mathcal{B}_E$ de E est l'ensemble de tous les atomes filtrés que l'on peut former en utilisant les symboles de prédicats de E, appliqués aux termes filtrés de $\mathcal{U}_E$.

Interprétation de Herbrand

Construction de $\mathcal{U}_E$

- À chaque constante a de E correspond la même constante dans $\mathcal{U}_E$
- S'il n'existe aucune constante dans E, alors on introduit la constante de Herbrand a_H dans $\mathcal{U}_E$
- À chaque foncteur f d'arité n, correspond dans $\mathcal{U}_E$ le même foncteur avec comme arguments des termes clos de $\mathcal{U}_E$

Construction de $\mathcal{B}_E$

- À chaque prédicat p d'arité n, correspond dans $\mathcal{B}_E$ le même prédicat avec comme arguments des termes clos de $\mathcal{U}_E$

Interprétation de Herbrand

Conclusion

L'univers de Herbrand est plus pauvre que l'univers du discours (pas de variables). Il est donc plus facile de calculer les valeurs de vérites de E.

Définition

Soit une ensemble de clauses E de langage $\mathcal{L}$.
L'*interprétation de Herbrand* I_E est une application entre E et $\mathcal{U}_E$ associant :

- à chaque constante $c \in E$ la même constante $c_{I_E} = c \in \mathcal{U}_E$
- à chaque foncteur f de E d'arité n, le foncteur $f_{I_E}((t_1)_{I_E}, \cdots, (t_n)_{I_E}) = f(t_1, \cdots, t_n)$
- à chaque prédicat p de E d'arité n, une relation quelconque $p_{I_E} \subseteq \mathcal{U}_E^n$

Modèle de Herbrand

Définition

Soit un ensemble de clauses E de langage $\mathcal{L}$. Soit I_E une interprétation de Herbrand. Si cette interprétation est un modèle pour chaque clause de E, alors elle est un modèle de herbrand pour E

Théorèmes

- Soit I'_E un modèle pour E. $I_E = \{B \in \mathcal{B}_E / \models_{I'_E} B\}$ est un modèle de Herbrand pour E
- Un ensemble de clauses E est insatisfiable ssi il n'a pas de modèle de Herbrand

Modèle minimal d'Herbrand

Objectif

Transformation d'un modèle non Herbrand en modèle de Herbrand

Théorème

Soit E un programme défini et I' un modèle non-Herbrand de E. Alors l'ensemble $I = \{A \in \mathcal{B}_E / \models_{I'} A\}$ est un modèle de Herbrand

Corollaire

Soit E un programme défini. La base de Herbrand $\mathcal{B}_E$ de E est un modèle de Herbrand de E

Modèle minimal d'Herbrand

Theorèmes

- Soient E un programme défini, $\mathcal{M}_E = \{\mathcal{I}_1, \mathcal{I}_2, \cdots\}$ une famille non vide de modèles de Herbrand de E. L'intersection $\mathcal{I} = \mathcal{I}_1 \cap \mathcal{I}_2 \cap \cdots$ est aussi un modèle d'Herbrand pour E (modèle minimal).
- Le modèle minimal d'Herbrand d'un programme défini est l'ensemble de toutes les conséquences logiques filtrées.

Opérateur de la conséquence immédiate

Définition

Soit E un programme défini. L'opérateur de conséquence immédiate $T_E : \mathcal{P}(\mathcal{B}_E) \to \mathcal{P}(\mathcal{B}_E)$ est défini par :

$$\begin{aligned} \mathcal{P}(\mathcal{B}_E) \ni \mathcal{I} \mapsto \mathcal{T}_E(\mathcal{I}) \quad = \quad & \{A \circ \sigma_f / A \circ \sigma_f \leftarrow B_1 \circ \sigma_f, \cdots, B_n \circ \sigma_f \\ & \text{où } (A \leftarrow B_1, \cdots, B_n) \in E \\ & \text{et } \{B_1 \circ \sigma_f, \cdots, B_n \circ \sigma_f\} \subseteq \mathcal{I}\} \end{aligned}$$

Unification

Objectif

Réduire le nombre d'éléments d'un ensemble de clauses E avant d'examiner s'il est satisfiable.

Définition

Soit $E = \{A_1, \cdots, A_n\}$ un ensemble de clauses. Une substitution σ est un unificateur de E ssi $A_1\sigma = \cdots = A_n\sigma$. Dans ce cas on dit que E est unifiable par σ.

Unification

Définition

Soit $E = \{A_1, \cdots, A_n\}$ un ensemble de clauses. Un unificateur θ de E est l'unificateur le plus général (UPG) si pour tout unificateur σ de E il existe une substitution τ telle que $\sigma = \theta \ o \ \tau$

Ensemble non apparié

Représentation

Les foncteurs et prédicats sont de la forme $r(s_1, \cdots, s_n)$ et peuvent être représentés par des listes $[s_0 = r, s_1, \cdots, s_n]$.

Définition

Soient 2 structure $u_1 = [s_0, \cdots, s_n]$ et $u_2 = [r_0, \cdots, r_n]$. L'ensemble non apparié $D(u_1, u_2)$ est défini par :

- Si s_0 et r_0 sont différents alors $D(u_1, u_2) = \{u_1, u_2\}$
- Sinon et si les sous-structures s_i et r_i sont identiques pour $i = 1, \cdots, k-1$, tandis que les sous-structures s_k et r_k sont différentes, alors $D_k(u_1, u_2) = \{t_1 = (s_k, \cdots, s_n), t_2 = (r_k, \cdots, r_n)\}$, où t_1 et t_2 sont les termes respectifs de u_1 et u_2 commençant au k-ième rang.

Algorithme d'Unification

Algorithme

1. Si $u_1 = u_2$, alors l'UPG de u_1 et u_2 est $\sigma = \varepsilon$ (identité). FIN. Sinon, posons $\sigma = \varepsilon$
2. Tant que $u_1\sigma \neq u_2\sigma$ faire
   Trouver la première sous-structure de u_1 qui soit différente de la sous-structure correspondante de u_2. Soit $D_k(u_1, u_2) = \{t_1, t_2\}$, l'ensemble non apparié à ces deux sous-structures non identiques.
   2.1 (Echec normal) Si ni t_1 ni t_2 n'est une variable, alors sortie en echec.
   2.2 (Vérification d'occurrence) Si l'un de t_1, t_2 est une variable contenue dans l'autre terme, alors sortie echec.
   2.3 (Suite) Sinon, si t_1 est une variable, alors on pose $\sigma = \sigma \ o \ \tau$ où $\tau = (t_1/t_2)$

Théorème de l'unification

Correction

Soit E un ensemble fini de clauses. Si E est unifiable, alors l'algorithme de l'unification termine en donnant l'UPG pour E. Sinon l'algorithme se termine en echec.

Complexité

En raison de l'étape de vérification d'occurrence, l'unification a un coût exponentiel en fonction de la longueur de l'entrée. Pour cette raison, PROLOG utilise l'algorithme d'unification sans cette étape. Cette catastrophe théorique peut avoir un impact pratique via les cercles vicieux entrainant des substitutions infinies ...

Réponse correcte à un programme

Définition

Soit E un programme défini, B un but $\leftarrow B_1, \cdots, B_n$. Une réponse à $E \cup \{B\}$ est une substitution $\sigma_{E,B}$ pour les variables du but B.

Définition

On dit que $\sigma_{E,B}$ est une réponse correcte à $E \cup \{B\}$ si $B_k \ o \ \sigma_{E,B}$ est une conséquence logique de E pour tout $k = 1, \cdots, n$, i.e.

$$E \models B_k \ o \ \sigma_{E,B}, \ \forall k = 1, \cdots, n$$

Réponse correcte à un programme

Théorème

Soit E un programme défini, B un but $\leftarrow B_1, \cdots, B_n$ et $\sigma_{E,B}$ une réponse telle que $B_k \ o \ \sigma_{E,B}$ soit filtré pour tout $k = 1, \cdots, n$. Alors les propositions suivantes sont équivalentes :

1. $\sigma_{E,B}$ est correcte
2. $B_k \ o \ \sigma_{E,B}$ est vrai pour tout $k = 1, \cdots, n$ dans tout modèle de Herbrand de E.
3. $B_k \ o \ \sigma_{E,B}$ est vrai pour tout $k = 1, \cdots, n$ dans $\mathcal{I}_E$ (point fixe des conséquences logiques de E)

Récapitulatif

Etapes pour la construction d'un programme

1. Trouver un modèle d'Herbrand à partir des modèles non-Herbrand
2. Trouver le modèle minimal d'Herbrand
3. Extraire la connaissance positive incluse dans le modèle minimal d'Herbrand afin de l'utiliser pour la construction du programme. En effet tous les éléments d'un programme qui se vérifient font partie du modéle d'Herbrand