Logique du premier ordre (HAI504I)

Licence Informatique
Département Informatique
Faculté des Sciences de Montpellier
Université de Montpellier

FACULTÉ DES SCIENCES MONTPELLIER

Examen du 11 janvier 2023

Le seul document autorisé est l'aide-mémoire (fourni en fin de sujet). L'examen dure 2h. Le barème est donné à titre indicatif, et pourra être légèrement modifié au vu des copies rendues. Le sujet comporte 2 pages (hors aide-mémoire) et il y a 5 exercices.

Il est recommandé de lire attentivement tout *le sujet avant de commencer à le traiter.*

Exercice 1 (sur 2 points)

Donner l'énoncé du premier théorème d'incomplétude de Gödel. Vous veillerez à introduire précisément les notions utilisées dans cet énoncé.

Exercice 2 (sur 5 points)

L'objectif de cet exercice est d'utiliser la méthode de résolution pour établir que C est conséquence logique de H_1, H_2 et H_3.

H_1 : *Si un étudiant de L2 résout ce problème alors tout étudiant de L3 résout ce problème.*

H_2 : *Icare est étudiant de L3.*

H_3 : *Icare ne résout pas ce problème.*

C : *Aucun étudiant de L2 ne résout ce problème.*

1. Définir le langage que vous aller utiliser (prédicats, constantes, symboles de fonctions).
2. Modéliser chaque phrase H_1, H_2, H_3, C par une formule de la logique du premier ordre sur le langage que vous avez défini à la question précédente.
3. Donner la formule (ou les formules, suivant la manière dont vous procédez) sur laquelle (lesquelles) la résolution s'appliquera, puis la (les) skolémiser et (les) clausifier.
4. Appliquer la méthode de résolution aux clauses obtenues.

Exercice 3 (sur 3 points)

1. Définir en Prolog un prédicat `conc(L1,L2,L3)` qui est vrai lorsque `L3` est la liste résultant de la concaténation de `L1` et `L2`, par exemple `?- conc([a,b,a],[b,a],[a,b,a,b,a])` rend true.
2. Définir en Prolog un prédicat `rev(L1,L2)` qui est vrai lorsque `L2` est la liste `L1` à l'envers par exemple `?- rev([a,b,d],[d,b,a])` rend true.

Exercice 4 (sur 5 points)

Soit $\mathcal{L}$ un langage du premier ordre. Soit T une théorie cohérente sur ce langage (un ensemble pas forcément fini de formules closes) et soit G une formule close sur ce même langage. On rappelle qu'une théorie T démontre une formule H, notation $T \vdash H$ lorsqu'il existe une partie finie $\{T_1, \ldots, T_p\}$ de T telle que le séquent $T_1, \ldots, T_p \vdash G$ soit démontrable dans $\mathcal{LK}$. Une théorie est dite cohérente lorsqu'elle ne démontre pas $\perp$.

1. Montrer que si T ne démontre pas G alors $T \cup \neg G$ est cohérente. Vous énoncerez clairement les théorèmes utilisés.
2. Montrer que si T ne démontre ni G ni $\neg G$ alors il existe des modèles de T dans lesquels G est vraie et d'autres dans lesquels G est fausse. Vous énoncerez clairement les théorèmes utilisés.
3. Connaissez-vous des exemples de théories T telles que, pour certaines formules G, T ne démontre ni G ni $\neg G$? Si oui, connaissez-vous un exemple de telle formule G ? Vous énoncerez clairement les théorèmes utilisés.

Exercice 5 (sur 5 points)

On considère les deux formules suivantes sur le langage défini par P unique prédicat, unaire.
$G : (\forall x.\ P(x)) \Rightarrow \perp$
$H : \exists x.(P(x) \Rightarrow \perp)$

1. Dessinez l'arbre de chacune des deux formules.
2. Montrez $H \Rightarrow G$ dans le calcul des séquents.
3. En déduire que $H \Rightarrow G$ est vraie dans toute interprétation. Vous énoncerez clairement le théorème utilisé.
4. Montrez en raisonnant dans une interprétation I quelconque que $G \Rightarrow H$ est vraie dans toute interprétation. Vous pourrez considérer deux cas, suivant que l'interprétation de $\forall x.P(x)$ est vraie ou fausse pour I.
5. En déduire que $G \Rightarrow H$ est démontrable dans LK, sans chercher à construire cette preuve. Vous énoncerez clairement le théorème utilisé.
6. Montrez $G \Rightarrow H$ dans le calcul des séquents.

Logique du premier ordre (HAI504I)

Licence Informatique (L3)
Département Informatique
Faculté des Sciences de Montpellier

Aide-mémoire

Ce document est le seul autorisé lors des examens de HAI504I. Il doit être reprographié et utilisé en l'état, et en particulier, aucune note manuscrite ne doit y être ajoutée.

Syntaxe de la logique du premier ordre

Soient $\mathcal{V}$ l'ensemble de variables d'individu, $\mathcal{S}_\mathcal{F}$ l'ensemble de symboles de fonctions, et $\mathcal{S}_\mathcal{P}$ l'ensemble de symboles de prédicats, tels que $\mathcal{S}_\mathcal{F} \cap \mathcal{S}_\mathcal{P} = \emptyset$.

Les termes du premier ordre sont le 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 $t_1, \ldots, t_n \in \mathcal{T}$ alors $f(t_1, \ldots, t_n) \in \mathcal{T}$.

Les constantes sont des fonctions d'arité 0.

Les formules du premier ordre sont le plus petit ensemble $\mathcal{F}$ t.q. :
- Si $P \in \mathcal{S}_\mathcal{P}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $P(t_1, \ldots, t_n) \in \mathcal{F}$;
- $\bot, \top \in \mathcal{F}$;
- 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 $\forall x.\Phi, \exists x.\Phi \in \mathcal{F}$.

Les connecteurs $\wedge$, $\vee$, et $\Leftrightarrow$ associent à gauche, tandis que le connecteur $\Rightarrow$ associe à droite. Pour les connecteurs, on a la précédence : $\neg \succ \wedge \succ \vee \succ \Rightarrow \succ \Leftrightarrow$. 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.

Sémantique de la logique du premier ordre

Une interprétation I est un ensemble non vide D_I, appelé le domaine de l'interprétation, muni d'une application $I(f)$ de D_I^n vers D_I pour chaque symbole de fonction f d'arité n ou d'un élément $I(c)$ de D_I pour chaque symbole de fonction c d'arité 0 (constante), et d'une application $I(P)$ de D_I^n vers $\mathcal{B}$ pour chaque symbole de prédicat P d'arité n.

Une affectation ρ est une application de $\mathcal{V}$ vers D_I. Pour toute affectation ρ, $\rho[v/x]$ est l'affectation envoyant chaque variable y autre que x vers $\rho(y)$, et x vers v.

Dans une interprétation I, et modulo l'affectation ρ, la sémantique des termes est définie par :
- Si $x \in \mathcal{V}$ alors $[\![x]\!]^I_\rho = \rho(x)$;
- Si $f \in \mathcal{S}_\mathcal{F}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $[\![f(t_1, \ldots, t_n)]\!]^I_\rho = I(f)([\![t_1]\!]^I_\rho, \ldots, [\![t_1]\!]^I_\rho)$;
  Si $f \in \mathcal{S}_\mathcal{F}$ d'arité 0 (constante) alors $[\![f]\!]^I_\rho = I(f)$.

Dans une interprétation I, et modulo l'affectation ρ, la sémantique des formules est définie par :

- Si $P \in \mathcal{S}_{\mathcal{P}}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $[\![P(t_1, \ldots, t_n)]\!]^I_\rho = I(P)([\![t_1]\!]^I_\rho, \ldots, [\![t_1]\!]^I_\rho)$;
- $[\![\top]\!]^I_\rho = T$, $[\![\bot]\!]^I_\rho = F$;
- Si $\Phi \in \mathcal{F}$ alors $[\![\neg\Phi]\!]^I_\rho = \neg_{\mathcal{B}}[\![\Phi]\!]^I_\rho$;
- Si $\Phi, \Phi' \in \mathcal{F}$ alors :
  - $[\![\Phi \wedge \Phi']\!]^I_\rho = [\![\Phi]\!]^I_\rho \wedge_{\mathcal{B}} [\![\Phi']\!]^I_\rho$, $[\![\Phi \vee \Phi']\!]^I_\rho = [\![\Phi]\!]^I_\rho \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho$, $[\![\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\Phi]\!]^I_\rho \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho$, $[\![\Phi \Leftrightarrow \Phi']\!]^I_\rho = [\![\Phi]\!]^I_\rho \Leftrightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho$.
- Si $x \in \mathcal{V}$ et $\Phi \in \mathcal{F}$ alors :
  - $[\![\forall x.\Phi]\!]^I_\rho = \bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}$, $[\![\exists x.\Phi]\!]^I_\rho = \bigvee_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}$.

Règles de preuve du calcul des séquents (système LK)

$$\frac{}{\Gamma, A \vdash \Delta, A}\ \mathsf{ax} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, B}\ \mathsf{cut}$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \mathsf{cont}_{\mathsf{left}} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\ \mathsf{cont}_{\mathsf{right}}$$

$$\frac{\Gamma \vdash \Delta, A \qquad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}\ \Rightarrow_{\mathsf{left}} \qquad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B}\ \Rightarrow_{\mathsf{right}}$$

$$\frac{\Gamma \vdash \Delta, A, B \qquad \Gamma, A, B \vdash \Delta}{\Gamma, A \Leftrightarrow B \vdash \Delta}\ \Leftrightarrow_{\mathsf{left}} \qquad \frac{\Gamma, A \vdash \Delta, B \qquad \Gamma, B \vdash \Delta, A}{\Gamma \vdash \Delta, A \Leftrightarrow B}\ \Leftrightarrow_{\mathsf{right}}$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{\mathsf{left}} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\ \wedge_{\mathsf{right}}$$

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee_{\mathsf{left}} \qquad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\ \vee_{\mathsf{right}}$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta}\ \neg_{\mathsf{left}} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\ \neg_{\mathsf{right}}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot_{\mathsf{left}} \qquad \frac{}{\Gamma \vdash \Delta, \top}\ \top_{\mathsf{right}}$$

$$\frac{\Gamma, A(t) \vdash \Delta}{\Gamma, \forall x.A(x) \vdash \Delta}\ \forall_{\mathsf{left}} \qquad \frac{\Gamma \vdash \Delta, A(x)}{\Gamma \vdash \Delta, \forall x.A(x)}\ \forall_{\mathsf{right}},\ x \notin \Gamma, \Delta$$

$$\frac{\Gamma, A(x) \vdash \Delta}{\Gamma, \exists x.A(x) \vdash \Delta}\ \exists_{\mathsf{left}},\ x \notin \Gamma, \Delta \qquad \frac{\Gamma \vdash \Delta, A(t)}{\Gamma \vdash \Delta, \exists x.A(x)}\ \exists_{\mathsf{right}}$$

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \ldots, x_n)/x]$, où $\{x_1, \ldots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \ldots, x_n)/x]$, où $\{x_1, \ldots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
  - Skolémisation : $\forall x_1. \ldots .\forall x_n.s(\Phi)$, où $\{x_1, \ldots, x_n\} = FV(s(\Phi))$;
  - Herbrandisation : $\exists x_1. \ldots .\exists x_n.h(\Phi)$, où $\{x_1, \ldots, x_n\} = FV(h(\Phi))$.