Université de Strasbourg
Logique et Programmation Logique — Feuille de CT

Examen Janvier 2011

Notes de cours manuscrites autorisées. Le barème est donné à titre indicatif.

Exercice 1 : résolution (5 points)

En utilisant la méthode de la résolution, montrer que l'ensemble des trois formules suivantes (où a est une constante) est insatisfiable (nommer et détailler chacune des étapes) :

$$\begin{array}{ll} F_1 & \exists z\ (q(f(z)) \wedge s(f(z), a)) \\ F_2 & \forall x \forall y \neg \exists z\ (p(x, y) \wedge s(x, z)) \\ F_3 & \forall x\ [q(x) \wedge \exists y\ s(x, y)] \Rightarrow [\exists y\ (r(y) \wedge p(x, y))] \end{array}$$

Exercice 2 : déduction naturelle (5 points)

1. Démontrer en utilisant la déduction naturelle que la formule suivante est valide (on présentera la preuve sous forme d'un arbre étiqueté par les règles utilisées) :

$$A \vee (B \wedge C) \Rightarrow (C \wedge B) \vee A$$

2. Supposons qu'on ajoute la règle suivante aux règles de la déduction naturelle pour la logique propositionnelle :

$$\frac{\Gamma, A \wedge B, A, B \vdash F}{\Gamma, A \wedge B \vdash F}$$

Le système formel obtenu est-il correct ? complet ? donner une preuve détaillée de vos affirmations (la réponse seule ne donne pas de points, non même pas un !).

Exercice 3 : satisfiabilité, indépendance (5 points)

Considérons les formules :

$$\begin{array}{ll} F_1 : & \forall x\ P(x, x) \\ F_2 : & \forall x\ \forall y\ (P(x, y) \Rightarrow P(y, x)) \\ F_3 : & \forall x\ \forall y\ \forall z\ (P(x, y) \wedge P(y, z) \Rightarrow P(x, z)) \end{array}$$

1. Montrer que $F_1, F_3 \not\models F_2$.
2. Montrer que $F_1, F_2 \not\models F_3$.
3. Montrer que $F_2, F_3 \not\models F_1$.
4. Montrer que $\{F_1, F_2, F_3\}$ est satisfiable.
5. Montrer que $F_1 \wedge F_2 \wedge F_3$ n'est pas valide.

Exercice 4 : prolog (5 points)

1. Définir un prédicat `split/4` qui étant donnée une liste d'entiers L et un entier X calcule la liste L1 des nombres inférieurs à X et la liste L2 des nombres supérieurs ou égaux à X.
2. Définir un prédicat `sum/2` qui étant donnée une liste d'entiers L calcule la somme de ses éléments.