Université de Strasbourg
Logique et Programmation Logique — Feuille de CT (Correction)

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) :

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

Correction :

1. Mise sous forme prénexe :

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

   Attention :
   - renommage de y en w pour éviter une capture de variables,
   - transformation de $\exists$ en $\forall$ quand $\exists$ est un gauche d'un $\Rightarrow$,
   - transformation de $\exists$ en $\forall$ quand on traverse une négation.

2. Skolémisation :
   On introduit une nouvelle constante z_0 et un nouveau symbole de fonction g d'arité un.

$$\begin{array}{ll} F_1 & q(f(z_0)) \wedge s(f(z_0), a) \\ F_2 & \forall x \forall y \forall z\ \neg(p(x, y) \wedge s(x, z)) \\ F_3 & \forall x \forall w\ ((q(x) \wedge s(x, w)) \Rightarrow (r(g(x)) \wedge p(x, g(x)))) \end{array}$$

3. Mise en forme de clauses :

$$\begin{array}{ll} C_1 & q(f(z_0)) \\ C_2 & s(f(z_0), a) \\ C_3 & \neg p(x, y) \vee \neg s(x, z) \\ C_4 & \neg q(x) \vee \neg s(x, w) \vee r(g(x)) \\ C_5 & \neg q(x) \vee \neg s(x, w) \vee p(x, g(x)) \end{array}$$

4. Résolution :
   On remarquera que la clause C_4 est inutile car aucune autre clause ne contient $r()$.

$$\begin{array}{lll} C_6 & \neg s(f(z_0), w) \vee p(f(z_0), g(f(z_0))) & (C_1 + C_5) \\ C_7 & p(f(z_0), g(f(z_0))) & (C_2 + C_6) \\ C_8 & \neg s(f(z_0), z) & (C_7 + C_3) \\ C_9 & \perp & (C_8 + C_2) \end{array}$$

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 !).

Correction :

1.

$$\dfrac{\dfrac{}{A \vee (B \wedge C) \vdash A \vee (B \wedge C)} \qquad \dfrac{\dfrac{}{A \vee (B \wedge C), A \vdash A}}{A \vee (B \wedge C), A \vdash (C \wedge B) \vee A}\,\text{Intro } \vee_d \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A \vee (B \wedge C), B \wedge C \vdash B \wedge C}}{A \vee (B \wedge C), B \wedge C \vdash C}\,\text{Elim } \wedge_d \qquad \dfrac{\dfrac{}{A \vee (B \wedge C), B \wedge C \vdash B \wedge C}}{A \vee (B \wedge C), B \wedge C \vdash B}}{A \vee (B \wedge C), B \wedge C \vdash C \wedge B}\,\text{Int}}{A \vee (B \wedge C), B \wedge C \vdash (C \wedge B) \vee A}\,\text{Intro } \vee_g}{\dfrac{A \vee (B \wedge C) \vdash (C \wedge B) \vee A}{\vdash\vdash A \vee (B \wedge C) \Rightarrow (C \wedge B) \vee A}\,\text{Intro } \Rightarrow}\,\text{Elim } \vee$$

2. Rappelons que le système formel pour la déduction naturelle présenté en cours est correct et complet.
   (a) Le système obtenu ici est donc complet (toute formule valide est un théorème), car on rajoute une règle donc l'ensemble des théorèmes que l'on peut prouver avec le système obtenu est plus grand que celui de la déduction naturelle.
   (b) Le système obtenu est aussi correct. Pour le montrer il y a deux possibilités :
      i. On montre que la nouvelle règle préserve la validité.
      ii. On montre que la nouvelle règle peut être simulée avec les autres règles de la déduction naturelle.

   Ici, on choisit la deuxième solution :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dots}{\Gamma, A \wedge B, A, B \vdash F}}{\Gamma, A \wedge B, A \vdash B \Rightarrow F}\,\text{Intro } \Rightarrow \qquad \dfrac{\dfrac{}{\Gamma, A \wedge B, A, B \vdash A \wedge B}\,\text{Ax}}{\Gamma, A \wedge B, A \vdash B}\,\text{Elim } \wedge_d}{\Gamma, A \wedge B, A \vdash F}\,\text{Elim } \Rightarrow}{\Gamma, A \wedge B \vdash A \Rightarrow F}\,\text{Intro } \Rightarrow \qquad \dfrac{\dfrac{}{\Gamma, A \wedge B \vdash A \wedge B}\,\text{Ax}}{\Gamma, A \wedge B \vdash A}\,\text{Elim } \wedge_g}{\Gamma, A \wedge B \vdash F}\,\text{Elim } \Rightarrow$$

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.

Correction :

1. Soit $U = \mathbb{N}$ et $P(x,y) = x \leq y$
2. Soit $U = \{1,2,3\}$ et $P(x,y) = (x,y) \in \{(1,1),(2,2),(3,3)\}$
3. Soit $U = \{1,2,3\}$ et $P(x,y) = (x,y) \in \{(1,1),(1,2),(2,1),(2,2)\}$
4. Soit $U = \mathbb{N}$ et $P(x,y) = x = y$.
5. Soit $U = \mathbb{N}$ et $P(x,y) = x < y$.

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.

   Correction :

   ```
   split([],X,[],[]).
   split([Y|R],X,[Y|L1],L2) :- split(R,X,L1,L2), Y<X.
   split([Y|R],X,L1,[Y|L2]) :- split(R,X,L1,L2), Y>=X.
   ```

2. Définir un prédicat `sum/2` qui étant donnée une liste d'entiers L calcule la somme de ses éléments.

   Correction :

   ```
   sum([],0).
   sum([T|R],Z) :- sum(R,Y), Z is T+Y.
   ```