Module FOND1

Magistère
Informatique et
Télécommunications
1ère année

Devoir maison

Le devoir devra être rendu **le vendredi 10 avril 2009**. *Les sources Ocaml du devoir devront être envoyés par mail à David Pichardie pour cette date. Le devoir peut être réalisé par petits groupes, mais chaque étudiant doit rendre une copie personnelle en indiquant clairement ses collaborateurs (et/ou collaboratrices). Les sources Ocaml seront indentés et commentés* **avec soin**. *L'évaluation du devoir tiendra compte de la précision des arguments et de la présentation générale.*

Le but de ce devoir est d'étudier une représentation « compacte » pour les preuves du système de Hilbert vu en TD.

Rappels : On reprend la définition des formules propositionnelles (on note $\mathcal{F}$ cet ensemble) en se restreignant aux variables propositionnelles (prises dans un ensemble $\mathcal{P}$) et au connecteur d'implication. L'ensemble des formules f *démontrables* (nous noterons $\vdash f$ ce fait) est définie par induction de la manière suivante :

– toute formule f pouvant être identifiée avec l'un des *axiomes* du système suivant est démontrable.

$$A \Rightarrow (B \Rightarrow A) \qquad (\mathcal{K})$$
$$(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)) \qquad (\mathcal{S})$$

où A,B et C désignent des formules quelconques. On note ces règles

$$\frac{}{f}\,\mathcal{K} \quad \text{et} \quad \frac{}{f}\,\mathcal{S}$$

– si g et $g \Rightarrow f$ sont démontrables alors f est démontrable. On note cette règle

$$\frac{g \Rightarrow f \quad g}{f}\,\mathsf{MP}$$

Exemples :

1. Soit $a \in \mathcal{P}$, la formule $a \Rightarrow (a \Rightarrow (a \Rightarrow a))$ est démontrable grâce à l'arbre de preuve suivant

$$\frac{\dfrac{}{(a \Rightarrow (a \Rightarrow a)) \Rightarrow (a \Rightarrow (a \Rightarrow (a \Rightarrow a)))}\,\mathcal{K} \qquad \dfrac{}{a \Rightarrow (a \Rightarrow a)}\,\mathcal{K}}{a \Rightarrow (a \Rightarrow (a \Rightarrow a))}\,\mathsf{MP}$$

2. Soit $a, b \in \mathcal{P}$, la formule $(a \Rightarrow b) \Rightarrow (a \Rightarrow a)$ est démontrable grâce à l'arbre de preuve suivant

$$\frac{\dfrac{}{(a \Rightarrow (b \Rightarrow a)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow a))}\,\mathcal{S} \qquad \dfrac{}{a \Rightarrow (b \Rightarrow a)}\,\mathcal{K}}{(a \Rightarrow b) \Rightarrow (a \Rightarrow a)}\,\mathsf{MP}$$

On propose de représenter les preuves de ce système à l'aide du langage $\mathcal{T}$ suivant.

Définition : Soit A l'alphabet $\{\mathcal{S}, \mathcal{K}, (,)\}$, $\mathcal{T}$ est le sous-ensemble de A^* défini inductivement par : $\mathcal{S} \in \mathcal{T}$, $\mathcal{K} \in \mathcal{T}$ et pour tous $t_1, t_2 \in \mathcal{T}$, $(t_1\ t_2) \in \mathcal{T}$.
Les éléments de $\mathcal{T}$ seront appelés *termes*. Pour simplifier l'écriture des termes, on prendra comme convention que la notation $(?\ ?)$ est associative à gauche. On pourra ainsi noter $(\mathcal{S}\ \mathcal{K}\ \mathcal{S})$ le terme $((\mathcal{S}\ \mathcal{K})\ \mathcal{S})$.
Les preuves de ce système de Hilbert sont codées par la fonction Φ qui transforme un arbre de preuve en un terme. F est définie inductivement par :

$$
\begin{aligned}
\Phi\left(\frac{}{\dots}\,\mathcal{S}\right) &= \mathcal{S} \\
\Phi\left(\frac{}{\dots}\,\mathcal{K}\right) &= \mathcal{K} \\
\Phi\left(\frac{\pi_1 \quad \pi_2}{\dots}\,\mathsf{MP}\right) &= (\Phi(\pi_1)\ \Phi(\pi_2))
\end{aligned}
$$

Exemples : Les preuves des deux exemples précédents sont codées par les termes $(\mathcal{K}\ \mathcal{K})$ et $(\mathcal{S}\ \mathcal{K})$.

Question 1. Soit $p \in \mathcal{P}$, donner le terme associé à une preuve de $p \Rightarrow p$.

On va maintenant « typer » les termes avec des formules. On notera $t : f$ le fait qu'un terme t admette la formule f pour type. Étant donné une formule f, on dira que f est *habité* s'il existe un terme t qui admette f pour type (i.e. $t : f$).

Question 2. Proposer des règles de typage. Le système de types ainsi défini devra vérifier la propriété

$$\forall f \in \mathcal{F}, \quad f \text{ est démontrable} \iff f \text{ est habité} \tag{1}$$

Question 3. Prouver la propriété (1).

Étant donné un terme t, on note $\mathrm{Types}(t)$ l'ensemble des types admis par t.

Question 4. Donner $\mathrm{Types}(\mathcal{K})$. Parmi ces types, lesquels vous paraissent les plus *simples* ? Définir formellement la relation « être plus simple que ». On notera $\preceq$ cette relation. La relation $\preceq$ devra vérifier les propriétés suivantes :

$$\forall t \in \mathcal{T},\ \forall f_1, f_2 \in \mathcal{F},\ \text{si } t : f_1 \text{ et } f_1 \preceq f_2 \text{ alors } t : f_2 \tag{2}$$

$$\forall t \in \mathcal{T}, \quad \begin{array}{l} \text{si } \mathrm{Types}(t) \neq \emptyset \text{ alors,} \\ \text{il existe } f \in \mathrm{Types}(t) \text{ tel que } \forall g \in \mathrm{Types}(t),\ f \preceq g \end{array} \tag{3}$$

Question 5. Prouver la propriété (2).

Question 6. Proposer un algorithme qui, pour tout terme t,
– renvoie `faux` si t n'est pas typable,
– renvoie sinon un type valide pour t, le plus simple possible.

Question 7. Prouver la validité de cet algorithme.

Question 8. Implanter cet algorithme en Ocaml en utilisant les définitions données en annexe. Donner le type obtenu pour le terme

$$(\mathcal{S}\ (\mathcal{S}\ (\mathcal{K}\ \mathcal{S})\ (\mathcal{S}\ (\mathcal{S}\ (\mathcal{K}\ \mathcal{S})\ (\mathcal{K}\ \mathcal{K}))\ (\mathcal{K}\ \mathcal{K})))\ (\mathcal{S}\ (\mathcal{K}\ \mathcal{K})\ (\mathcal{S}\ \mathcal{K}\ \mathcal{K})))$$

Question 9. Déduire des questions précédentes un vérificateur de type qui, étant donnés un terme t et une formule f, teste si t admet f pour type. Écrire cette fonction en Ocaml. Utiliser pour cela la librairie et le squelette mis en ligne à l'adresse suivante :

`http://www.irisa.fr/lande/pichardie/L3/LOG/maison`