IENAC-S-95 Mercredi 23 octobre 1996

Examen de logique et calculabilité

Durée : 2 h

Tous documents autorisés

1 Logique

1. Soit le jugement suivant :

   > «Si un algorithme est simple, il est correct. Donc un algorithme est correct *ou* compliqué.»

   Montrer avec un argument sémantique (par exemple une table de vérité) que ce propos est ou n'est pas une tautologie suivant qu'on interprète le «ou» classiquement ou exclusivement.

2. Formaliser les faits suivants et montrer par résolution que si la bonne a dit vrai, le majordome a menti :

   > «La bonne dit qu'elle a vu le majordome dans la salle à manger. La salle à manger et la cuisine sont deux pièces voisines. Le coup de feu a été tiré dans la cuisine et a pu être entendu dans toutes les pièces voisines. Le majordome, qui entend correctement, dit qu'il n'a pas entendu le coup de feu.»

3. On considère une signature $\Sigma = \{c_2, n_0\}$, des symboles de prédicat l_1, mb_2, ap_3 et l'ensemble $\mathcal{P}$ de formules suivant :

$$l(n) \tag{1}$$

$$\forall x \forall y\ l(y) \rightarrow l(c(x,y)) \tag{2}$$

$$\forall x \forall y\ mb(x, c(x,y)) \tag{3}$$

$$\forall x \forall y \forall z\ mb(x,y) \rightarrow mb(x, c(z,y)) \tag{4}$$

$$\forall y\ ap(n,y,y) \tag{5}$$

$$\forall e \forall x \forall y \forall z\ ap(x,y,z) \rightarrow ap(c(e,x), y, c(e,z)) \tag{6}$$

   (a) Montrer par résolution que $\mathcal{P} \vdash \forall x \forall y\ mb(x, c(y, c(x,n)))$.

   (b) Montrer en utilisant le calcul des séquents que $\mathcal{P} \vdash \forall a \forall x\ ap(c(a,n), x, c(a,x))$.

   (c) On considère le domaine $\mathcal{L}$, le plus petit ensemble vérifiant :
   - le tuple vide $<> \in \mathcal{L}$;
   - pour tout $x \in \mathbb{N}$ et $y \in \mathcal{L}$, la paire $< x, y > \in \mathcal{L}$.

      i. Décrire les éléments de $\mathcal{L}$.

      ii. Donner une interprétation des symboles c, n, l, mb, ap dans $\mathcal{L} \cup \mathbb{N}$ telle que toutes les formules de $\mathcal{P}$ soient valides dans $\mathcal{L} \cup \mathbb{N}$ (i.e. $\mathcal{L} \cup \mathbb{N}$ est un modèle pour $\mathcal{P}$). Indication : on pourra prendre $\hat{c} : \mathbb{N} \times \mathcal{L} \longrightarrow \mathcal{L}$ où $\hat{c}$ est l'interprétation du symbole c.

2 Calculabilité

On considère un langage de programmation $\mathcal{L}$ dont la grammaire est la suivante :

$$
\begin{array}{rcl}
\textit{variable} & ::= & \{x, y, z \ldots\} \\
\textit{terme} & ::= & \texttt{0} \\
 & ::= & \textit{variable} \\
 & ::= & \textit{variable}\texttt{+ 1} \\
\textit{programme} & ::= & \textit{variable}\ \texttt{:=}\textit{terme} \\
 & ::= & \textit{programme}\ \texttt{;}\ \textit{programme} \\
 & ::= & \texttt{do}\textit{variable}\texttt{times}\ \textit{programme}\texttt{end}
\end{array}
$$

où chaque variable peut contenir un entier, `:=` est l'affectation, `;` la séquence et où `do`x`times` p`end` est evalué en calculant la valeur a de x puis en exécutant a fois le programme p.

1. Écrire en $\mathcal{L}$ le programme qui effectue l'addition de deux nombres x et y (au départ dans les variables `x` et `y`) et qui range le résultat $x + y$ dans la variable `z`.
2. Montrer que $\mathcal{L}$ n'est pas TURING-complet, i.e. il existe des fonctions calculables non exprimables dans $\mathcal{L}$ (ne pas chercher nécessairement à exhiber une telle fonction).
3. Proposer une construction syntaxique à ajouter à $\mathcal{L}$ pour obtenir un langage $\mathcal{L}'$ TURING-complet (justifier).
4. On considère une implémentation du langage $\mathcal{L}'$ où les variables sont codées par des mots finis de 32 bits. Que peut-on dire du problème de l'arrêt pour cette implémentation ?