Modèles de Calcul
Examen - Seconde session
L3 Informatique

19/06/2012

Durée : 3h00 sans document.

1 Machines de Turing : calculabilité

Parmi les problèmes suivants dire ceux qui sont (in)décidables. Justifer. (Ils sont indépendants et listés approximativement par ordre de difficulté croissante).

Q.1
- **Donnée :** le code d'une machine de Turing M
- **Question :** $L(M) \subseteq 0^*$?

Q.2
- **Donnée :** Deux formules ϕ, ψ; de la logique du premier ordre, sans variable libre.
- **Question :** ϕ et ψ sont logiquement équivalentes ?

Q.3
- **Donnée :** Le code d'une machine de Turing qui calcule la fonction (partielle) f
- **Question :** existe-t-il un mot w tel que $f(w) = w$?

Q.4
- **Donnée :** Deux codes de machines de Turing $M1; M2$
- **Question :** $L(M1) \cap L(M2) = \emptyset$?

Q.5
- **Donnée :** Deux codes de machines de Turing $M1; M2$ qui s'arrêtent sur toutes les données.
- **Question :** $L(M1) \cap L(M2) = \emptyset$?

2 Machine de Turing : Programmation

Pour chaque machine de Turing construite il faut donner une explication de son principe en plus de sa table de transition. Une machine comportant la moindre faute sans explications ne sera pas comptabilisée.

Vous pouvez introduire autant de symboles que vous voulez pour l'alphabet de bande, cette dernière sera considérée comme infinie dans les deux sens.

Q.6 Programmez une machine qui reconnait le langage $L = \{a^n b^{2^n} \mid \forall n \in \mathcal{N}\}$

Q.7 On note $\langle M \rangle$ le mot qui code la machine de Turing M. Soit $L_k = \{\langle M \rangle | |L(M)| \geq k\}$., montrer que :

- L_k n'est pas récursif.
- $L_k \in RE$.

3 λ-calcul : Programmation

Pour programmer il vous est conseillé d'introduire des macros pour les fonctions intermédiaires et de les utiliser ensuite pour éviter d'avoir un programme illisible.

Rappels

Les booléens sont définis par $\texttt{true} \equiv \lambda xy.x$ et $\texttt{false} \equiv \lambda xy.y$. Le if est codé par $\texttt{if} \equiv \lambda z.(z\ M\ N)$
Les couples sont égalements encodés par $[M, N] \equiv \lambda z.(z\ M\ N)$.
Les entiers peuvent être codés par :

Church	Barendregt
$\lceil 0 \rceil \equiv \lambda fx.x$	$\lfloor 0 \rfloor \equiv \lambda x.x$
$\lceil n \rceil \equiv \lambda fx.(\overbrace{f\ \ldots(f}^{n}\ x))$	$\lfloor n \rfloor \equiv [\texttt{false}, \lfloor n-1 \rfloor]$
$0_c? \equiv \lambda n.(n\ \lambda x.\texttt{false}\ \texttt{true})$	$0_b? \equiv \lambda x.(x\ \texttt{true})$

On considère que les fonctions de base suivantes **Add, Mult, Div, Sub** codants respectivement l'addition, la multiplication, la division et la soustraction ainsi que l'égalité, notée **Eg** sur les entiers sont données.

De plus on rappelle que le combinateur de point fixe Y est défini par $Y \stackrel{\text{def}}{=} \lambda f.(\lambda x.(f\ (x\ x))\ \lambda x.(f\ (x\ x)))$ et qu'il a la propriété suivante : $(Y\ M) \rightarrow^*_\beta (M\ (Y\ M))$.

On considère également les fonctions sur les listes dont on rappelle le codage en λ-calcul : une liste $[1; 2; 3; 4]$ est codée par le terme $(cons\ \lceil 1 \rceil\ (cons\ \lceil 2 \rceil\ (cons\ \lceil 3 \rceil\ (cons\ \lceil 4 \rceil\ Nil))))$ où

$$
\begin{array}{lcl}
cons & \stackrel{\text{def}}{=} & \lambda elc.(c\ e\ l) \\
Nil & \stackrel{\text{def}}{=} & \lambda l.\texttt{true} \\
head & \stackrel{\text{def}}{=} & \lambda l.(l\ \texttt{true}) \\
tail & \stackrel{\text{def}}{=} & \lambda l.(l\ \texttt{false}) \\
Null? & \stackrel{\text{def}}{=} & \lambda l.(l\ \lambda ht.\texttt{false})
\end{array}
$$

Q.8 Encodez un prédicat **Prem** tel que (**Prem** $\lceil n \rceil$) se réduit vers $\texttt{true}$ si et seulement si n est un nombre premier.

Q.9 Encodez un prédicat **Parfait** qui permet de savoir si un nombre est parfait. On rappelle que la condition pour qu'un nombre soit parfait est qu'il doit être égal à la somme de ses diviseurs.

4 λ-calcul : Problème

On considère le λ-calcul simplement typé suivant : $\Lambda_\mathcal{N} ::= \mathbf{S} \mid \mathbf{0} \mid \emptyset \mid \mathbf{x} \mid (\mathbf{\Lambda_\mathcal{N}}\ \mathbf{\Lambda_\mathcal{N}}) \mid \lambda\mathbf{x} : \tau.\mathbf{\Lambda_\mathcal{N}}$

où x est dans un ensemble fini dénombrable de variables, $\mathbf{S}, \mathbf{0}$ et $\emptyset$ sont des constantes et avec les types définis par $\tau ::= \mathcal{N} \mid \mathcal{U} \mid \tau \to \tau$. Les axiomes pour typer les constantes sont : $\Gamma \vdash \mathbf{0} : \mathcal{N}$, $\Gamma \vdash \mathbf{S} : \mathcal{N} \to \mathcal{N}$ et $\Gamma \vdash \emptyset : \mathcal{U}$.

On définit inductivement une relation d'ordre $\leq$ sur les termes (bien typés) par :

- Base : pour tout terme M on a $\emptyset \leq M$, pour toute variable $x \leq x$ et pour les constantes $\mathbf{0} \leq \mathbf{0}$ et $\mathbf{S} \leq \mathbf{S}$.
- Si $M \leq M'$ et $N \leq N'$ alors $(M\ N) \leq (M'\ N')$.
- Si $M \leq M'$ alors $\lambda x : \tau.M \leq \lambda x : \tau.M'$

Q.8 Montrer que si $M \leq N$ et $M \rightarrow_\beta M'$ alors il existe N' tel que $N \rightarrow_\beta^* N'$ et $M' \leq N'$.

On pourra utiliser la propriété suivante supposée démontrée : Pour tous termes P, Q, R et S, pour toute variable x tels que l'on ait : $P \leq R$ et $Q \leq S$ alors $P[x := Q] \leq R[x := S]$

Q.10 Soient M, N deux termes clos de type $\mathcal{N}$ en forme normale tels que $M \leq N$. Montrer que $M \equiv N$.

Q.11 Soient M, N deux termes clos de type $\mathcal{N}$ tels que $M \leq N$. Supposons que la forme normale de M est M_{norm}. Que peut on dire de M_{norm} ? Que peut on dire de la forme normale de N ? Quelle conclusion peut on tirer de ces observations ?