Examen — 2ème année de Master Recherche MDFI

Logique et Théorie du Calcul — Paul Ruet

14 décembre 2009, 14h - 17h

Les documents sont autorisés. Les exercices sont indépendants. Le barème est indicatif.
Attention à la clarté de la rédaction.

Exercice 1 (1 point) Montrer que la fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ définie par $f(0) = f(1) = 1$ et $f(n+2) = 2f(n)f(n+1)$ est primitive réursive.

Exercice 2 (5 points) On dit que la fonction $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ est définie par itération à partir de $g : \mathbb{N} \rightarrow \mathbb{N}$ quand :

$$f(x, n) = \underbrace{(g \circ \dots \circ g)}_n(x),$$

avec $f(x, 0) = x$ par convention. Soit C la plus petite classe des fonctions contenant les fonctions initiales usuelles (projections, fonction nulle, fonction successeur), la fonction J de codage des couples ($J(x, y) = (x + y)(x + y + 1)/2 + y$) et les fonctions de décodage $\pi_1^2 \circ J^{-1}, \pi_2^2 \circ J^{-1}$, et close par composition et par itération.

1. Montrer que toutes les fonctions de C sont primitives réursives.
2. Montrer que pour tout $k \geq 1$, la fonction $J^{(k)} : \mathbb{N}^k \rightarrow \mathbb{N}$ de codage des k -uplets ($J^{(1)}$ est l'identité et $J^{(k+1)}(x_1, \dots, x_{k+1}) = J(J^{(k)}(x_1, \dots, x_k), x_{k+1})$) et les fonctions de décodage associées sont dans C .
3. Montrer que C est l'ensemble des fonctions primitives réursives.

Exercice 3 (3 points) Pour chacune des fonctions suivantes définies sur les entiers, construire une machine de Turing qui la calcule.

1. $(x, y) \mapsto x \dot{-} y$ définie par $x \dot{-} y = x - y$ si $x \geq y$, 0 sinon ;
2. $(x, y) \mapsto x \times y$.

La représentation des entiers est laissée au choix.

Exercice 4 (4 points) Deux machines de Turing M et M' de même alphabet Σ sont dites dénotationnellement équivalentes si elles ont le même comportement asymptotique, c'est-à-dire si pour tout $v \in \Sigma^*$: $M(v) \uparrow \Leftrightarrow M'(v) \uparrow$, $M(v) \downarrow_n \Leftrightarrow M'(v) \downarrow_n$, $M(v) \downarrow_o \Leftrightarrow M'(v) \downarrow_o$, et $M(v) = w \Leftrightarrow M'(v) = w$.

1. Montrer que toute machine de Turing est dénotationnellement équivalente à une machine de Turing dont la fonction de transition n'utilise que les déplacements $+1$ et -1 (jamais de sur-place).

2. Montrer que toute machine de Turing est dénotationnellement équivalente à une machine de Turing où la contrainte ci-dessus sur les déplacements est remplacée par la contrainte suivante sur l'écriture : on suppose que le langage est $\{0, 1\}$ et on demande, étant donnée une machine M , que la nouvelle machine M' change systématiquement la lettre lue, autrement dit $T(s, 0) = (s', 1, d)$ et $T(s, 1) = (s', 0, d)$.
3. Les deux contraintes ci-dessus sont-elles compatibles (peut-on construire M' satisfaisant les deux) ?

Exercice 5 (2 points) Soient L un langage (pour la logique du premier ordre) et ϕ une formule à une variable libre. Donner une preuve de la formule $\exists x \forall y (\phi(y) \Rightarrow \phi(x))$ dans le calcul des séquents classique. (Pour commencer, on pourra prouver cette formule sous l'hypothèse $\exists x \phi(x)$.)

Exercice 6 (5 points) On note $\phi^\bullet$ le code ($\in \mathbb{N}$) d'une formule quelconque ϕ du langage de l'arithmétique, et $\{v_0, v_1, \dots\}$ l'ensemble infini dénombrable des variables.

1. Montrer que la fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ définie par :

$$f(k) = \begin{cases} (\theta[s^k(0)/v_0])^\bullet & \text{si } k = \theta^\bullet \text{ pour une certaine formule } \theta, \\ 0 & \text{si } k \text{ n'est pas le code d'une formule,} \end{cases}$$

est définissable. Soit donc $\beta(x, y)$ une formule définissant f .

2. Soient $\psi(y)$ une formule quelconque à une variable libre y dans le langage de l'arithmétique, et $\delta(x)$ la formule $\forall y (\beta(x, y) \Rightarrow \psi(y))$ à une variable libre x . Montrer que $\mathbb{N} \models \delta[s^n(0)/x] \Leftrightarrow \psi[s^{f(n)}(0)/y]$ pour tout entier n .
3. A l'aide d'une diagonalisation, donner une formule close ϕ dans le langage de l'arithmétique telle que $\mathbb{N} \models \phi \Leftrightarrow \psi[s^{\phi^\bullet}(0)/y]$.
4. En conclure qu'il n'existe pas de formule V à une variable libre, disons v_0 , dans le langage de l'arithmétique telle que pour toute formule close ϕ , $\mathbb{N} \models \phi$ si et seulement si $\mathbb{N} \models V[s^{\phi^\bullet}(0)/v_0]$. Autrement dit, la vérité (dans $\mathbb{N}$) n'est pas définissable par une formule de l'arithmétique.