Examen du cours de λ-calcul *
du 10 Avril 2002

(Enoncé, solutions et commentaires)

Vous disposez de 4 heures; les notes du cours sont autorisées.

I. Lambda-calcul non typé et vérification de typage dans le Système F. Dans ce qui suit $A \to B$ désigne l'ensemble de toutes les fonctions de l'ensemble A dans l'ensemble B, et $N \to_{rec} N$ l'ensemble des fonctions récursives totales de N dans N.

A. Soit $G \in (\,(N \to_{rec} N) \times N) \to (\,N \to_{rec} N)$ définie par:
$G(f,a) \equiv g$ avec $g(0) = a$ et $g(n+1) = f(n)$.

1. Etant donnés $a \in N$ et un représentant u de de f dans Λ^0 (relativement aux entiers de Church) donner un représentant de $g \equiv G(f,a)$. On pourra supposer connu un représentant p du prédécesseur.

2. En déduire un terme W qui représente la fonctionnelle G elle-même (en un sens évident).

3. Représenter plus généralement pour tout $p \geq 1$ la fonctionnelle:
$G_p \in (\,(N \to_{rec} N) \times N^{p+1}) \to (\,N \to_{rec} N)$ définie par:
$G_p(f,\bar{a}) \equiv g$ avec $g(k) = a_k$ si $k \leq p$ et $g(n+p+1) = f(n)$.

B.

1. Vérifier que $\vdash_F Z : \forall X\,(N, X, X \to X)$, où $Z =_{def} \lambda\nu.((\nu\,\lambda d.F)V)$

2. Vérifier que $\vdash_F W : (N \to N), N, N \to N$, ou modifier votre W pour qu'il en soit ainsi.

C. (Facultatif)

1. Pouvait-on faire le même travail avec les récursives partielles au lieu des récursives totales?

2. Déduire de B que si $f \in N \to N$ est prouvablement totale dans l'arithmétique du second ordre alors il en est de même pour $G(f,a)$.

---

*DEA de Logique et Fondements de l'Informatique (2001-2002), UFR de Mathématiques, Université Paris 7.

II. Programmation dans AF_2. On rappelle que le type des arbres binaires (sans étiquettes) peut être représenté dans AF_2 par la formule suivante:

$T[x] =_{def} \forall X\, ((\forall y, z\, (Xy, Xz \to Xcyz)), X\varepsilon \to Xx\,)$

où ε et c sont des constantes pour les constructeurs du type (resp. 0-aire et binaire), et cyz abrège $c(y, z)$.

On désigne par T le type du système F sous-jacent à $T[x]$.

On supposera connus des termes représentant respectivement dans AF_2 les fonctions de base éventuellement nécessaires sur les entiers.

A.

1. On fixe deux variables x, f de Λ et on désigne par $\mathcal{T}^*$ le plus petit sous-ensemble de Λ contenant x et tel que $u, u' \in \mathcal{T}^*$ implique $fuu' \in \mathcal{T}^*$. Vérifier que pour tout $u \in \mathcal{T}^*$ le terme de la forme $\lambda f.\lambda x.u$ est clos normal et typable de type T dans F.

2. On désigne par $\mathcal{T}$ le plus petit ensemble de Λ contenant $\lambda f.\lambda x.x$ et tel que $t, t' \in \mathcal{T}$ implique $\lambda f.\lambda x.(f(tf)x)(t'f)x \in \mathcal{T}$. Vérifier que tout terme de $\mathcal{T}$ est de type $T[\tau]$ pour un certain terme τ du langage du premier ordre associé.

B. Programmer dans AF_2, c'est-à-dire représenter par un terme de type adéquat qu'on précisera, et relativement à un ensemble d'équations qu'on précisera évidemment également, et en justifiant rapidement les solutions proposées:

1) Les constructeurs du type (c et ε)

2) Le calcul de la longueur de la branche la plus à gauche.

3) Un test qui répond V si l'arbre est trivial et F sinon.

4) Le calcul du nombre de feuilles (extrémités basses) de l'arbre.

C. (facultatif). *Cette question n'a été proposée qu'aux étudiants qui avaient fait tout le reste et fini au bout de trois heures, et a été traitée, correctement, par deux étudiants, dans le cadre de F.*

Montrer que T est un type de données (au choix: dans F ou dans $AF2$).

III. Sémantique continue. Dans ce qui suit $\mathcal{D}$ est un *dcpo* quelconque. On désignera par $\mathcal{F}$ l'ensemble de toutes les fonctions de $\mathcal{D}$ dans $\mathcal{D}$, et par $[\mathcal{D} \to \mathcal{D}]$ le dcpo des fonctions continues de $\mathcal{F}$, muni de l'ordre extensionnel ("point par point").

Les non mathématiciens de formation pourront, si ils le préfèrent, travailler en supposant que $\mathcal{D}$ est de la forme $(P^{coh}(D, \frown\!\!\!\!\smile), \subseteq)$, où D est un ensemble et $\frown\!\!\!\!\smile$ une relation de cohérence sur cet ensemble (i.e. une relation binaire réflexive et symétrique). On rappelle que les éléments compacts d'un tel domaine sont les parties finies cohérentes $a, b\ldots$ de D.

[Remarque: ce choix n'a été fait par personne!]

Pour $d, e \in \mathcal{D}$ on désigne par $\varepsilon_{d,e}$ la fonction de $\mathcal{F}$ définie par:

$$\varepsilon_{d,e}(x) =_{def} \left\{ \begin{array}{c} e \text{ si } x \succeq d \\ \perp \text{ sinon} \end{array} \right\}$$

1. Montrer que $\varepsilon_{d,e}$ est continue ssi d est compact

2. Montrer que $\varepsilon_{d,e}$ est compacte dans $[\mathcal{D} \to \mathcal{D}]$ ssi on a de plus e compact.

3. Soit $f \in [\mathcal{D} \to \mathcal{D}]$. Montrer que pour tous h, k compacts on a $\varepsilon_{h,k} \preceq f$ ssi $(h,k) \in Tr(f)$ (c'est-à-dire ssi $k \preceq f(h)$).

C. (facultatif). *Cette question n'a été proposée qu'aux étudiants qui avaient fait tout le reste et fini au bout de trois heures, et a été traitée, correctement, par un étudiant, qui a en fait pû finir le problème et traiter toutes les questions facultatives.*

On considère les fonctions $g_{d,e}$, pour $d, e \in \mathcal{D}$, définies par:

$g_{d,e}(x) =_{def} \left\{ \begin{array}{c} e \text{ si } x \not\preceq d \\ \perp \text{ sinon} \end{array} \right\}$

Discuter la continuité, et éventuellement la compacité, de $g_{d,e}$.

Solutions de l'examen du 10 Avril 2002.

Barême:

Le problème (questions non facultatives) a été noté sur 21,5 (=7+ 8,5 +6). Chaque question facultative correctement traitée rapportait de 1 à 2 points de plus suivant la difficulté (mais plafonnement des bonus!). Les raisonnements et présentations soignées ont été appréciés.

I)

A1. $g =_{def} \lambda\nu.Z\nu a(u)(pred)\nu$

A2. $W =_{def} \lambda f.\lambda a.\lambda\nu.Z\nu a(f)(pred)\nu$

A3. En se servant du fait que $G_{p+1}(f, a\bar{b}) = G_1(G_p(f,\bar{b}), a)$ on trouve $W_{p+1} = \lambda f.\lambda a_0....\lambda a_p.(W[W_p f a_1...a_p])a_0$, et finalement $W_{p+1} = \lambda f.\lambda a_0....\lambda a_p.(W(W...(Wf)a_p)...a_1)a_0$. On a des solutions un peu plus économique en terme de nombre d'étapes de réduction, mais un peu plus pénible à écrire, en remplaçant chaque Whb, et éventuellement $Z\nu$, par leurs réduits de tête “naturels”.

B1. Soit $Z' =_{def} (\nu\lambda d.F)V$. Dans le contexte $\nu : N$ on a $Z' : B$, car dans ce contexte on a $\lambda d.F : B \to B$, et $V : B$, et $\nu : (B \to B), B \to B$. Donc $Z' : \forall X\,(X, X \to X)$ donc $Z' : X, X \to X$. On termine avec une $\to$-intro suivie d'une $\forall X$-intro (pour avoir successivement $\vdash Z : N \to (X, X \to X)$ et $\vdash Z : \forall X\,(N \to (X, X \to X))$.

B2. Suit de ce que $\vdash_F Z : N, N, N \to N$

C1. Sans problème (mais demanderait tout de même une petite justification).

C2. Suit de ce que les fonctions prouvablement totales sont exactement les fonctions représentables par un terme clos typable de type $N \to N$ dans F, et du typage de W.

II)

A1-2. Inductions simples sur la structure des termes.

B1. $\lambda f.\lambda x.x : T[\varepsilon]$ et

$\lambda\mu\lambda\nu.\lambda f.\lambda x.(f(\mu f)x)(\nu f)x : \forall x.\forall y.(T[x], T[y] \to T[cxy])$.

Notez qu'il découlait de la preuve de A2 que $\vdash t : T\tau$ et $\vdash t' : T\tau'$ impliquait $\vdash \lambda f.\lambda x.(f(tf)x)(t'f)x : Tc\tau\tau'$, ce qui suggérait fortement un terme convenant au constructeur. Ce terme s'obtient par ailleurs directement dès qu'on cherche à prouver $Tcxy$ dans le contexte $\mu : Tx$ et $\nu : Ty$, et, plus précisément, si on cherche à prouver $Xcxy$ dans le contexte $\mu : Tx$, $\nu : Ty$, $f : \forall x \forall y\,(Xx, Xy \to Xcxy)$, $x : X\varepsilon$. Il s'agit en un certain sens du programme "standard" pour le constructeur. Un autre terme a été trouvé, mais avec une démonstration moins naturelle et plus compliquée.

B.2. On cherche $t : \forall x.(T[x] \to N[l_g x])$ en utilisant les équations: $l_g\varepsilon = 0$ et $l_g(cxy) = s(l_g(x))$. Les méthodes standard donnent, sans "casser" N :

$\lambda\mu.(\mu\, \lambda x\lambda y.suc\, x)\hat{0}$, où *suc* est un terme pour le successeur, et en le cassant:

$\lambda\mu\lambda f\lambda a.(\mu\, \lambda x.\lambda y.fx)a =_\eta \lambda\mu\lambda f.\mu\, \lambda x\lambda y.fx$.

B.3. $Z =_{def} \lambda\mu.((\mu\, \lambda d\lambda d'.F)V)$ est le terme le plus simple qui convient.

B.4.On cherche $t : \forall x.(T[x] \to N[n_f x])$ en utilisant les équations: $n_f\varepsilon = 1$ et $n_f(cxy) = n_f(x) + n_f(y)$. On trouve, sans casser N, le terme $\lambda\mu.\mu\, plus\, \hat{1}$.

III)

C. $g_{d,e}$ est toujours continue, elle n'est sûrement pas compacte si e n'est pas compact, mais la condition n'est pas suffisante (contre-exemples).

Quelques erreurs commises.

- Attention, dans le système F le passage de $\vdash_F \lambda x.t : A \to B$ à $x : A \vdash_F t : B$ demande une petite justification, contrairement aux systèmes E et S où c'est une conséquence immédiate des règles. Ici l'hypothèse permet de déduire, par *axiome* plus $\to$-élim): $x : A \vdash_F (\lambda x.t)x : B$, et il faut ensuite utiliser, par exemple, la conservation du typage (dans F) par β-réduction.

- Dans un *dcpo* l'ordre n'est pas nécessairement total (et il ne l'est jamais dans les cas d'intérêt), la négation de $x \succeq d$ n'implique donc pas $d \prec x$.

- Pour W_p certains d'entre vous on imaginé des solutions compliquées faisant intervenir le terme de Maurey et des fonctions $\lambda\nu.p_k^\nu$ où p_k^ν serait la ν-ième projection sur les k-uples, à supposer que ν soit un entier assez petit. En tant que fonction sur N il s'agit d'une fonction de graphe fini, donc évidemment représentable, mais il est difficile de donner un sens au terme p_k^ν pour ν une variable, et l'introduction d'une telle notation $(\lambda\nu.p_k^\nu)$ demande une définition/justification qui serait, au moins, pénible. De toute manière il est bien préférable, et infiniment plus simple de passer par des conditionnels successifs.