Lambda-calcul monadique

Documents autorisés (en particulier le poly).
Le but de ce problème est d'étudier une extension du lambda-calcul typé avec des types dits "monadiques".
La syntaxe des termes est:

$s, t, u, v, \ldots$	$::=$	$x, y, z, \ldots$	variables
	$\mid$	uv	applications
	$\mid$	$\lambda x \cdot u$	abstractions
	$\mid$	$\texttt{let}\,\texttt{val}\,x = u\,\texttt{in}\,v$	séquence
	$\mid$	$\texttt{val}\,u$	conversion en valeur

On identifie les termes modulo α-renommage, qui est la plus petite congruence (notée $=$) telle que $\lambda x \cdot u = \lambda y \cdot (u[x := y])$ et $\texttt{let}\,\texttt{val}\,x = u\,\texttt{in}\,v = \texttt{let}\,\texttt{val}\,y = u\,\texttt{in}\,(v[x := y])$. On supposera que tous les termes sont systématiquement convertis en des termes normalisés de sorte que deux variables liées ont des noms différents, et aucune variable libre n'est liée. La variable x est *liée* dans $\texttt{let}\,\texttt{val}\,x = u\,\texttt{in}\,v$; les variables libres de $\texttt{let}\,\texttt{val}\,x = u\,\texttt{in}\,v$ sont celles de u, plus celles de v sauf x; les variables libres de $\texttt{val}\,u$ sont celles de u. Ceci nous permettra d'ignorer en grande partie les problèmes de capture de variables.

Les règles de réduction sont:

$$\begin{array}{lrcl} (\beta) & (\lambda x \cdot u)v & \to & u[x := v] \\ (\texttt{let}) & \texttt{let}\,\texttt{val}\,x = \texttt{val}\,u\,\texttt{in}\,v & \to & v[x := u] \end{array}$$

I. Le calcul non typé

1. On définit la traduction suivante du lambda-calcul monadique vers le lambda-calcul pur:

$$\begin{array}{rcl@{\qquad}rcl} x^\circ & = & x & (\texttt{val}\,u)^\circ & = & u^\circ \\ (uv)^\circ & = & u^\circ v^\circ & (\texttt{let}\,\texttt{val}\,x = u\,\texttt{in}\,v)^\circ & = & (\lambda x \cdot v^\circ)u^\circ \\ (\lambda x \cdot u)^\circ & = & \lambda x \cdot u^\circ & & & \end{array}$$

   Montrer que $u^\circ[x := v^\circ] = (u[x := v])^\circ$ pour tous termes u, v et toute variable x. (On traitera les cas les plus importants.)

2. En déduire que si $u \to u'$ en lambda-calcul monadique, autrement dit par les règles (β) ou $(\texttt{let})$, alors $u^\circ \to u'^\circ$ dans le lambda-calcul pur.

3. On rappelle que le λ^*-calcul a pour termes $t, u, v, \ldots := x \mid uv \mid \lambda x \cdot u \mid (\lambda^* x \cdot u)v$, et pour règle de réduction unique $(\lambda^* \cdot u)v \to u[x := v]$. On rappelle aussi que le λ^*-calcul termine et est confluent (cf. cours). Soit $\to_{\texttt{let}}$ la notion de réduction fondée sur la seule règle $(\texttt{let})$ dans le λ-calcul monadique. Pourquoi $\to_{\texttt{let}}$ termine-t-il?

4. En admettant que la notion de réduction $\to$ du λ-calcul monadique est localement confluent, pourquoi ne peut-on pas appliquer le lemme de Newman et conclure qu'elle est confluente? Justifier.

5. On définit une notion de réduction parallèle $\Rightarrow$ par:

$$\frac{}{u \Rightarrow u} \qquad \frac{u \Rightarrow u' \quad v \Rightarrow v'}{uv \Rightarrow u'v'} \qquad \frac{u \Rightarrow u'}{\lambda x \cdot u \Rightarrow \lambda x \cdot u'}$$

$$\frac{u \Rightarrow u' \quad v \Rightarrow v'}{(\lambda x \cdot u)v \Rightarrow u'[x := v']} \qquad \frac{u \Rightarrow u'}{\texttt{val}\,u \Rightarrow \texttt{val}\,u'} \qquad \frac{u \Rightarrow u' \quad v \Rightarrow v'}{\texttt{let}\,\texttt{val}\,x = \texttt{val}\,u\,\texttt{in}\,v \Rightarrow v'[x := u']}$$

On admettra que $\Rightarrow$ est fortement confluente: si $u \Rightarrow u_1$ et $u \Rightarrow u_2$, alors il existe v tel que $u_1 \Rightarrow v$ et $u_2 \Rightarrow v$. En déduire que la notion de réduction du lambda-calcul monadique est confluente.

II. Traduction du lambda-calcul par valeur

On rappelle que le lambda-calcul par valeur est le langage dont les termes sont les lambda-termes usuels (non monadiques), et dont l'unique règle de réduction est:

$$(\beta_V) \quad (\lambda x \cdot u)V \to u[x := V]$$

où V est restreint à être une *P-valeur*, c'est-à-dire une variable ou une lambda-abstraction (mais pas une application).

1. On définit la traduction des lambda-termes en lambda-termes monadiques comme suit:

$$\begin{array}{rcl} x^\star & = & \mathtt{val}\, x \\ (uv)^\star & = & \mathtt{let\ val}\ x_u = u^\star\ \mathtt{in}\ (\mathtt{let\ val}\ x_v = v^\star\ \mathtt{in}\ x_u x_v) \\ (\lambda x \cdot u)^\star & = & \mathtt{val}(\lambda x \cdot u^\star) \end{array}$$

   où x_u et x_v sont des variables fraîches (ni libres ni liées ni dans u ni dans v). Montrer que si $u \to u'$ par (β_V), alors $u^\star \to^+ u'^\star$ dans le lambda-calcul monadique.

2. Donner un contre-exemple montrant que la proposition suivante est fausse: si $u \to u'$ dans le lambda-calcul (non nécessairement par valeur), alors $u^\star \to^* u'^\star$ dans le lambda-calcul monadique.

III. Le calcul simplement typé

Le lambda-calcul monadique admet une discipline des types simples dans laquelle on a un nouveau constructeur de type $\boldsymbol{T}$, appelé la *monade*. Précisément, les types sont:

$$\begin{array}{rcll} F & := & b & \text{types de base} \\ & | & F \supset F & \text{types de fonction} \\ & | & \boldsymbol{T}F & \text{type monadique} \end{array}$$

Les règles de typage sont:

$$\frac{}{\Gamma, x : F \vdash x : F}\,(Ax)$$

$$\frac{\Gamma, x : F \vdash u : G}{\Gamma \vdash \lambda x \cdot u : F \supset G}\,(\supset I) \qquad \frac{\Gamma \vdash u : F \supset G \quad \Gamma \vdash v : F}{\Gamma \vdash uv : G}\,(\supset E)$$

$$\frac{\Gamma \vdash u : F}{\Gamma \vdash \mathtt{val}\, u : \boldsymbol{T}F}\,(\boldsymbol{T}I) \qquad \frac{\Gamma \vdash u : \boldsymbol{T}F \quad \Gamma, x : F \vdash v : \boldsymbol{T}G}{\Gamma \vdash \mathtt{let\ val}\ x = u\ \mathtt{in}\ v : \boldsymbol{T}G}\,(\boldsymbol{T}E)$$

Il se trouve que l'on peut encore une fois interpréter ce système de typage comme les règles de déduction d'une logique, la *logique laxe* de Curry.

1. Montrer que si $\Gamma, x : F, \Delta \vdash u : G$ et $\Gamma \vdash v : F$ sont dérivables dans ce système, alors $\Gamma, \Delta \vdash u[x := v] : G$ est dérivable dans ce système.

2. Montrer l'autoréduction: si $\Gamma \vdash u : F$ dans ce système, et $u \to u'$ par (β) ou $(\mathtt{let})$, alors $\Gamma \vdash u' : F$.

3. Montrer, par un argument simple utilisant la question précédente et la question I.2, que le lambda-calcul monadique typé a la propriété de normalisation forte.

4. En déduire que la logique laxe est cohérente.

5. On définit une traduction de la logique minimale intuitionniste en logique laxe par:

$$b^\star = b \qquad (F \supset G)^\star = F^\star \supset \boldsymbol{T} G^\star$$

On étend cette traduction aux contextes par $(x_1 : F_1, \ldots, x_n : F_n)^\star = x_1 : F_1^\star, \ldots, x_n : F_n^\star$. Montrer que si u est un lambda-terme (non monadique) tel que $\Gamma \vdash u : F$, alors $\Gamma^\star \vdash u^\star : \boldsymbol{T} F^\star$. ($u^\star$ est défini dans la partie II.)

IV. La monade des continuations

On se fixe un type quelconque F_0 une fois pour toutes. On pose $\neg_0 F = F \supset F_0$, et l'on définit les constructions suivantes dans le lambda-calcul typé (non monadique):

$$\begin{array}{rcl} \boldsymbol{T} F & = & \neg_0 \neg_0 F \\ \mathtt{val}\ u & = & \lambda z \cdot z u \\ \mathtt{let}\,\mathtt{val}\ x = u\ \mathtt{in}\ v & = & \lambda z \cdot u(\lambda x \cdot v z) \end{array}$$

où z est une variable fraîche.

1. Montrer que, avec ces définitions, et en supposant que l'on dispose de la règle η du lambda-calcul, les règles de réduction (β) et $(\mathtt{let})$ de la section I sont valides.

2. Montrer que, avec ces définitions, les règles de typage de la section III sont valides.