Langages de programmation: sémantique et typage
Examen partiel 2004-2005

François Pottier

7 janvier 2005

Toutes les questions, à l'exception des questions 5 et 6, peuvent être traitées sans avoir nécessairement répondu à celles qui précèdent.

Présentation du système F

Le système F ou λ-*calcul polymorphe* est une extension du λ-calcul simplement typé. Les types, environnements de typage, expressions et contextes y sont définis de la façon suivante :

$$\begin{array}{rcl}
\tau & ::= & \alpha \mid \tau \to \tau \mid \forall\alpha.\tau \\
\Gamma & ::= & \bullet \mid \Gamma; x : \tau \\
e & ::= & x \mid \lambda(x : \tau).e \mid e\ e \mid \Lambda\alpha.e \mid e\ \tau \\
C & ::= & [] \mid \lambda(x : \tau).C \mid C\ e \mid e\ C \mid \Lambda\alpha.C \mid C\ \tau
\end{array}$$

La variable α est liée dans l'expression $\Lambda\alpha.e$, qui est une *abstraction vis-à-vis d'un type.* L'expression $e\ \tau$ est une *application d'une expression à un type.* Ces deux constructions se combinent de la même façon que la λ-abstraction et l'application habituelles, comme le montrent les règles de réduction :

$$\begin{array}{rcllr}
(\lambda(x : \tau).e_1)\ e_2 & \longrightarrow & [e_2/x]e_1 & & (\beta) \\
(\Lambda\alpha.e)\ \tau & \longrightarrow & [\tau/\alpha]e & & (\beta\text{-}type) \\
C[e] & \longrightarrow & C[e'] & \text{si } e \longrightarrow e' & (contexte)
\end{array}$$

La méta-variable C dénote un contexte arbitraire : aucune stratégie d'évaluation n'est imposée.

Voici à présent les règles de typage du système F :

$$\textsc{Var}\quad \Gamma \vdash x : \Gamma(x) \qquad\qquad \textsc{Abs}\quad \frac{\Gamma; x : \tau' \vdash e : \tau}{\Gamma \vdash \lambda(x : \tau').e : \tau' \to \tau} \qquad\qquad \textsc{App}\quad \frac{\Gamma \vdash e : \tau' \to \tau \qquad \Gamma \vdash e' : \tau'}{\Gamma \vdash e\ e' : \tau}$$

$$\textsc{TAbs}\quad \frac{\Gamma \vdash e : \tau \qquad \alpha \notin \mathrm{ftv}(\Gamma)}{\Gamma \vdash \Lambda\alpha.e : \forall\alpha.\tau} \qquad\qquad \textsc{TApp}\quad \frac{\Gamma \vdash e : \forall\alpha.\tau}{\Gamma \vdash e\ \tau' : [\tau'/\alpha]\tau}$$

Les trois premières règles sont exactement les mêmes que dans le λ-calcul simplement typé. Dans le cas de la règle Abs, l'annotation de types τ' portée par la variable x dans l'expression $\lambda(x : \tau').e$ est utilisée pour imposer le type qui sera associé à x dans la prémisse. Nous présentons donc une variante du système F dans laquelle le type des paramètres de fonctions est déclaré explicitement.

La règle TAbs indique que, si une expression est abstraite vis-à-vis d'une variable de types α, alors son type sera universellement quantifié vis-à-vis de α. Il s'agit donc d'une règle *d'introduction* du quantificateur universel ou règle de *généralisation.*

La règle TApp indique que, si une expression admet un type universel $\forall\alpha.\tau$, alors elle peut être appliquée à un type τ', ce qui provoque le remplacement de α par τ' dans le type τ. Il s'agit donc d'une règle *d'élimination* du quantificateur universel ou règle *d'instanciation.*

Quelques questions élémentaires

Question 1 *Considérons le terme*

$$(\Lambda\alpha.\lambda(f : \beta \to \alpha).f\ x)\ (\alpha \to \beta)\ (\lambda(x : \beta).\lambda(y : \alpha).x)$$

Quelles en sont les variables de termes et variables de types libres ? Dans quel(s) environnement(s) est-il bien typé et quel est alors son type ? Donner une séquence de réductions issue de ce terme et conduisant à un terme irréductible. $\diamond$

Ajout de paires primitives

Question 2 *Indiquer comment la définition du système F (c'est-à-dire sa syntaxe, sa sémantique, et son système de types) doit être modifiée lorsque l'on souhaite introduire la notion de paire. On pourra considérer l'introduction et l'élimination des paires, au choix, soit comme de nouvelles constructions primitives, soit comme des applications de nouvelles constantes choisies de façon appropriée.* $\diamond$

Ajout de paires par codage

Il est naturel d'ajouter au λ-calcul une notion primitive de paire. Néanmoins, on peut démontrer que cela n'augmente pas l'expressivité de ce calcul. En effet, dans un cadre non typé, on peut considérer l'introduction et l'élimination des paires comme du sucre syntaxique :

$$\begin{array}{rcl} (e_1, e_2) & \equiv & \lambda f.(f\ e_1\ e_2) \\ \pi_i\ e & \equiv & e\ (\lambda x_1.\lambda x_2.x_i) \end{array}$$

Ce codage est dit « à la Church. »

Question 3 *Préciser quelles hypothèses de fraîcheur sont nécessaires pour ce codage. Vérifier, dans ce cadre non typé, que l'expression $\pi_i\ (e_1, e_2)$ a le comportement attendu.* $\diamond$

Revenons à présent à un cadre typé. Supposons donc donné le système F, *non doté de paires*, et supposons que l'on veuille y coder les paires « à la Church. » On peut se demander si un tel codage est toujours possible et, si oui, quelle en est la définition exacte. Peut-être sera-t-il cette fois nécessaire d'employer des abstractions et applications de types explicites.

Question 4 *Dans le système F non doté de paires, proposer une définition pour les trois notations suivantes :*

$$\begin{array}{rcl} \tau_1 \times \tau_2 & \equiv & \ldots \textit{un type} \ldots \\ ((e_1, e_2) : \tau_1 \times \tau_2) & \equiv & \ldots \textit{une expression} \ldots \\ \pi_i\ (e : \tau_1 \times \tau_2) & \equiv & \ldots \textit{une expression} \ldots \end{array}$$

L'introduction de paire est notée $((e_1, e_2) : \tau_1 \times \tau_2)$, et non simplement (e_1, e_2), car les types τ_1 et τ_2 peuvent apparaître dans l'expansion de ce sucre syntaxique. On suppose donc ces types fournis explicitement. De même, l'élimination de paire est notée $\pi_i\ (e : \tau_1 \times \tau_2)$ et non simplement $\pi_i\ e$. $\diamond$

Question 5 *Vérifier que les règles habituelles de typage des paires (question 2) sont admissibles dans le système obtenu par codage (question 4). Une règle de typage*

$$\frac{P}{C}$$

est dite admissible *si et seulement si l'énoncé $P \Rightarrow C$ est vrai.* $\diamond$

Question 6 *Ce codage « à la Church » des paires serait-il possible si on se plaçait non pas dans le cadre du système F mais dans celui de ML ?* $\diamond$

Question subsidiaire *Toujours dans le système F, définir un codage similaire pour les sommes.* $\diamond$