Programmation, examen: appel par nom

Correction.

On va considérer un sous-langage de mini-Caml, dont les termes obéissent à la syntaxe :

$$\begin{array}{rcll} M, N, P, ... & ::= & x & \text{variables (non modifiables)} \\ & | & MN & \text{application de } M \text{ à } N \\ & | & \texttt{fun}\ x \to M & \text{fonction qui à } x \text{ associe } M \\ & | & \texttt{0}|\texttt{1}|\ldots| & \text{constantes entières } n \in \mathbb{Z} \end{array}$$

On en rappelle la sémantique opérationnelle à grands pas, donnée dans le cours :

$$\frac{}{\rho, \mu \vdash x \Rightarrow \mu, \rho(x)}\ (Var)$$

$$\frac{n \in \mathbb{Z}}{\rho, \mu \vdash n \Rightarrow \mu, n}\ (Cst)$$

$$\frac{\rho, \mu \vdash M \Rightarrow \mu', \overbrace{(x, M', \rho')}^{\text{clôture}} \quad \rho, \mu' \vdash N \Rightarrow \mu'', V \quad \rho'[x := V], \mu'' \vdash M' \Rightarrow \mu''', V'}{\rho, \mu \vdash MN \Rightarrow \mu''', V'}\ (App)$$

$$\frac{}{\rho, \mu \vdash \texttt{fun}\ x \to M \Rightarrow \mu, \underbrace{(x, M, \rho)}_{\text{clôture}}}\ (Fun)$$

On appellera dorénavant cette sémantique la sémantique *en appel par valeur* de mini-Caml. Noter que les environnements ρ envoient les variables x vers des *valeurs*, dans le domaine :

$$Val ::= \mathbb{Z} | \underbrace{Var \times Terme \times Env}_{\text{clôture}}$$

où $Env = Var \to Val$, et $Terme$ est l'ensemble des termes.

La sémantique *en appel par nom* de mini-Caml est en revanche la suivante, où les environnements $\varrho \in Env'$ envoient maintenant les variables vers des couples (M, ϱ) (des clôtures généralisées, car M n'est pas forcément une fonction).

$$\frac{}{\varrho, \mu \vdash x \Rightarrow \mu, \varrho(x)}\ (Var')$$

$$\frac{n \in \mathbb{Z}}{\varrho, \mu \vdash n \Rightarrow \mu, n}\ (Cst')$$

$$\frac{\varrho, \mu \vdash M \Rightarrow \mu', (\texttt{fun}\ x \to M', \varrho') \quad \varrho'[x := (N, \varrho)], \mu' \vdash M' \Rightarrow \mu'', r}{\varrho, \mu \vdash MN \Rightarrow \mu'', r}\ (App')$$

$$\frac{}{\varrho, \mu \vdash \texttt{fun}\ x \to M \Rightarrow \mu, (\texttt{fun}\ x \to M, \varrho)}\ (Fun')$$

1. Les jugements de la sémantique en appel par nom sont de la forme $\varrho, \mu \vdash M \Rightarrow \mu', r$, où r est le *résultat* de l'évaluation du terme M dans l'environnement ϱ, partant de la mémoire μ et aboutissant à la mémoire μ'.

   Pour que la définition de cette sémantique ait un sens, quel doit être l'ensemble Res des résultats r possibles ?

   L'ensemble $\mathbb{Z}$ union l'ensemble de toutes les clôtures généralisées.

2. Pour que cette définition ait un sens, l'ensemble des environnements Env' doit être tel que $Env' = Terme \times Env'$. Montrer que cette définition a un sens, en ceci qu'il existe un ensemble Env' qui vérifie l'équation récursive $Env' = Terme \times Env'$.

   Bonus : montrer qu'il en existe en fait un *plus petit*, qu'on appellera (l'unique) ensemble des environnements en appel par nom Env'.

   L'énoncé était erroné, et contenait deux erreurs. D'abord, la plus petite solution de l'équation est bien sûr l'ensemble vide ! La plus petite solution non vide, par contre, est l'ensemble des listes infinies de termes, autrement dit $(Terme)^{\mathbb{N}}$.

   La deuxième erreur est que l'ensemble des environnements Env' devait évidemment être tel que $Env' = Var \to Terme \times Env'$. L'ensemble vide est toujours une solution (en supposant que Var est non vide). Cependant, la plus petite solution non vide est l'ensemble des arbres infinis dont les sommets sont des termes dans $Terme$, qui ont un fils par variable dans Var, ce fils étant récursivement un arbre.

   Pour montrer qu'il existe une plus petit solution, on peut montrer que toute solution contient l'arbre infini ci-dessus, donc l'ensemble des arbres ci-dessus est la plus petite solution.

3. Donner les dérivations sémantiques tant en appel par valeur qu'en appel par nom de $(\texttt{fun}\ x \to 3)1$. Les résultats sont-ils les mêmes dans les deux sémantiques ?

   En appel par valeur :

$$\dfrac{\dfrac{}{\rho, \mu \vdash \texttt{fun}\ x \to 3 \Rightarrow \mu, (x, 3, \rho)}\,(Fun) \quad \dfrac{}{\rho, \mu \vdash 1 \Rightarrow \mu, 1}\,(Cst) \quad \dfrac{}{\rho[x := 1], \mu \vdash 3 \Rightarrow \mu, 3}\,(Cst)}{\rho, \mu \vdash (\texttt{fun}\ x \to 3)1 \Rightarrow \mu, 3}\,(App)$$

   En appel par nom :

$$\dfrac{\dfrac{}{\varrho, \mu \vdash \texttt{fun}\ x \to 3 \Rightarrow \mu, (\texttt{fun}\ x \to 3, \varrho)}\,(Fun') \quad \dfrac{}{\varrho[x := (1, \varrho)], \mu \vdash 3 \Rightarrow \mu, 3}\,(Cst')}{\varrho, \mu \vdash (\texttt{fun}\ x \to 3)1 \Rightarrow \mu, 3}\,(App')$$

   Les résultats sont donc les mêmes dans les deux sémantiques.

4. Soit Ω le terme $(\texttt{fun}\ x \to xx)(\texttt{fun}\ x \to xx)$. On rappelle qu'il n'existe aucune dérivation de $\rho, \mu \vdash \Omega \Rightarrow \mu', V$ en sémantique en appel par valeur. Montrer qu'il n'existe pas non plus de dérivation de $\varrho, \mu \vdash \Omega \Rightarrow \mu', r$ en sémantique en appel par nom.

   S'il en existe une, elle est nécessairement de la forme :

$$\dfrac{\dfrac{}{\varrho, \mu \vdash \texttt{fun}\ x \to xx \Rightarrow \mu, (\texttt{fun}\ x \to xx, \varrho)}\,(Fun') \quad \begin{array}{c} \vdots\ \pi \\ \varrho[x := (\texttt{fun}\ x \to xx, \varrho)], \mu \vdash xx \Rightarrow \mu'', r \end{array}}{\varrho, \mu \vdash (\texttt{fun}\ x \to xx)(\texttt{fun}\ x \to xx) \Rightarrow \mu'', r}\,(App')$$

   où π est de la forme :

$$\dfrac{\dfrac{}{\varrho', \mu \vdash x \Rightarrow \mu, (\texttt{fun}\ x \to xx, \varrho)}\,(Var') \quad \dfrac{\dfrac{}{\varrho[x := (x, \varrho')], \mu \vdash x \Rightarrow \mu, (x, \varrho')} \quad \vdots\ ?}{\varrho[x := (x, \varrho')], \mu \vdash xx \Rightarrow \mu'', r}\,(App')}{\underbrace{\varrho[x := (\texttt{fun}\ x \to xx, \varrho)]}_{\varrho'}, \mu \vdash xx \Rightarrow \mu'', r}\,(App')$$

mais il n'est pas possible d'écrire quoi que ce soit à la place du point d'interrogation " ?", car dans (x, ϱ'), x ne commence pas pas `fun`.

Le principal problème ici est la règle (Var'), qui aurait dû être écrite :

$$\frac{\varrho(x) = (N, \varrho') \quad \varrho', \mu \vdash N \Rightarrow \mu', r}{\varrho, \mu \vdash x \Rightarrow \mu, r} \; (Var')$$

Autrement dit, la demande de valeur de x aurait dû provoquer le calcul de la clôture (N, ϱ') qui est sa valeur. Même comme cela, Ω n'aurait pas eu de valeur, cette fois-ci pour une raison similaire au cas par valeur : il n'y a pas de dérivation de taille minimale de $\varrho', \mu \vdash xx \Rightarrow \mu'', r$.

5. Montrer qu'il n'existe aucune dérivation de $\rho, \mu \vdash (\texttt{fun}\ x \to 3)\Omega \Rightarrow \mu', V$ en sémantique en appel par valeur. On admettra le résultat rappelé ci-dessus selon lequel il n'existe aucune dérivation de $\rho, \mu \vdash \Omega \Rightarrow \mu', V$ en sémantique en appel par valeur.

   S'il en existait une, elle serait de la forme :

$$\dfrac{\dfrac{}{\rho, \mu \vdash \texttt{fun}\ x \to 3 \Rightarrow \mu, (x, 3, \rho)}\,(Fun) \qquad \begin{array}{c}\vdots\ \pi \\ \rho, \mu \vdash \Omega \Rightarrow \mu', V'\end{array} \qquad \dfrac{}{\rho[x := V'], \mu' \vdash 3 \Rightarrow \mu', 3}\,(Cst)}{\rho, \mu \vdash (\texttt{fun}\ x \to 3)\Omega \Rightarrow \mu', 3}\,(App)$$

   mais la dérivation π n'existe pas.

6. Montrer qu'il existe par contre une dérivation de $\varrho, \mu \vdash (\texttt{fun}\ x \to 3)\Omega \Rightarrow \mu', r$ en sémantique en appel par nom. Que vaut le résultat r ? Commenter.

$$\dfrac{\dfrac{}{\varrho, \mu \vdash \texttt{fun}\ x \to 3 \Rightarrow \mu, (\texttt{fun}\ x \to 3, \varrho)}\,(Fun') \qquad \dfrac{}{\varrho[x := (\Omega, \varrho)], \mu \vdash 3 \Rightarrow \mu, 3}\,(Cst')}{\varrho, \mu \vdash (\texttt{fun}\ x \to 3)\Omega \Rightarrow \mu, 3}\,(App')$$

   Le résultat r est donc 3. En d'autres termes, l'appel par nom a permis de ne pas calculer l'argument Ω, dont la valeur n'aurait pas été utilisée de toute façon.

7. Démontrer que dans toute dérivation $\rho, \mu \vdash M \Rightarrow \mu', V$ en sémantique en appel par valeur, et dans toute dérivation $\varrho, \mu \vdash M \Rightarrow \mu', r$ en sémantique en appel par nom, on a $\mu = \mu'$. En déduire une simplification des deux sémantiques.

   Les deux démonstrations se font par récurrence sur la taille de la dérivation. Les cas de base sont (Var), (Cst), (Fun) d'une part, et (Var'), (Cst'), (Fun') d'autre part, et le cas des applications consiste en trois, respectivement deux, appels à l'hypothèse de récurrence.

   On peut donc réécrire toutes les règles en supprimant purement et simplement la partie parlant des mémoires tant à gauche du $\vdash$ qu'à droite du $\Rightarrow$.

8. Le résultat de la question précédente, et la simplification afférente, seraient-ils toujours possibles dans le langage mini-Caml complet tel que vu dans le cours ? Donner un exemple explicite.

   Non, car les effets de bord modifient la mémoire. Notamment, on a toujours $\rho, \mu \vdash \texttt{ref}\ M \Rightarrow \mu', V$ avec $\mu' \neq \mu$.