Éléments de correction du TD8 - Sémantique Statique

Exercice 1

Programme 1

On commence par la règle du programme, qui fournit une branche pour la déclaration des variables, une branche pour l'utilisation de ces variables :

$$\frac{\overset{\text{Branche 1}}{D \xrightarrow{d} \rho} \qquad\qquad \overset{\text{Branche 2}}{< C, \rho > \xrightarrow{c} \text{Void}}}{DC \rightarrow \text{Void}}\ [\text{prog}]$$

Traitons ensuite le cas de la branche 1 :

$$\frac{\dfrac{\rho_1 : [x_1 \mapsto \text{Entier}]}{\text{var } x_1\text{Entier} \xrightarrow{d} \rho_1} \quad \dfrac{\dfrac{\rho_3 : [x_2 \mapsto \text{Entier}]}{\text{var } x_2\text{Entier} \xrightarrow{d} \rho_3} \quad \dfrac{\rho_4 : [x_3 \mapsto \text{Booléen}]}{\text{var } x_3\text{Booléen} \xrightarrow{d} \rho_4} \quad Dom(\rho_3) \cap Dom(\rho_4) = \emptyset}{(\text{var } x_2\text{Entier}; D_3) \xrightarrow{d} \rho_2}\ [\text{seq}]\ \rho_2 = \rho_3 \cup \rho_4 \quad Dom(\rho_1) \cap Dom(\rho_2) = \emptyset}{(\text{var } x_1\text{Entier}; D_2) \xrightarrow{d} \rho}\ [\text{seq}]\quad \rho = \rho_1 \cup \rho_2$$

Toutes les feuilles sont des axiomes et les applications des règles sont correctes ssi :

- $\rho_1 : [x_1 \mapsto \text{Entier}]$, $\rho_3 : [x_2 \mapsto \text{Entier}]$, $\rho_4 : [x_3 \mapsto \text{Booléen}]$,
- $\rho_2 : [x_2 \mapsto \text{Entier}, x_3 \mapsto \text{Booléen}]$,
- $\rho : [x_1 \mapsto \text{Entier}, x_2 \mapsto \text{Entier}, x_3 \mapsto \text{Booléen}]$,

Donc la déclaration des variables est "correcte". Traitons maintenant le cas des commandes avec ρ comme fonction de type (branche 2) :

$$\frac{[\text{aff.}]\ \dfrac{\overset{\text{OK}}{x \in Dom(\rho)} \quad \overset{\text{OK}}{\rho(x) = Entier} \quad \overset{\text{axiome}}{< 3, \rho > \xrightarrow{e} \text{Entier}}}{< x := 3, \rho > \xrightarrow{c} \text{Void}} \qquad \dfrac{\dfrac{\dfrac{\overset{\text{axiome car } \rho(x_3) = \text{Booléen}}{< x_3, \rho > \xrightarrow{e} \rho(x_3)}}{< x_3, \rho > \xrightarrow{e} \text{Booléen}}}{< \text{not } x_3, \rho > \xrightarrow{e} \text{Booléen}} \quad \overset{\text{Branche 2b}}{< C, \rho > \xrightarrow{c} \text{Void}}}{< \text{tq } EC, \rho > \xrightarrow{c} \text{Void}}\ [\text{tq}]}{< x := 3; \text{tq } EC, \rho > \xrightarrow{c} \text{Void}}\ [\text{seq}]$$

La sous-branche "test" n'a pas posé de problème, maintenant essayons de finir la branche "commande" :

$$\frac{\dfrac{x_1 \in Dom(\rho) \quad \rho(x_1) = \text{Entier} \quad \dfrac{...+entier}{< x_2 + 1, \rho > \xrightarrow{e} \text{Entier}}}{< x_1 := x_2 + 1, \rho > \xrightarrow{c} \text{Void}} \qquad \dfrac{x_3 \in Dom(\rho)\, \rho(x_3) = \text{Booléen} \quad \dfrac{...ou\ boolen}{< x_3 \text{ et } true, \rho > \xrightarrow{e} \text{Booléen}}}{< x_3 := x_3 \text{ et } true, \rho > \xrightarrow{c} \text{Void}}}{< x_1 := x_2 + 1; x_3 := x_3 \text{ et } true, \rho > \xrightarrow{c} \text{Void}}\ [\text{seq}]$$

Les dernières sous-branches ne sont pas difficiles, et permettent de clore l'arbre, le programme est donc bien typé.

Exercice 2

1. Opérateurs de comparaison
   - On rajoute dans les expressions :

   $$E ::= \ldots \mid E = E \mid E < E$$

   - Maintenant, voyons les règles à rajouter. On décide de ne faire la comparaison que sur les entiers (pas de = booléen) :

   $$\frac{<E_1, \rho> \xrightarrow{e} \text{ Entier} \quad <E_2, \rho> \xrightarrow{e} \text{ Entier}}{<E_1 = E_2> \xrightarrow{e} \text{ Booléen}}$$

   $$\frac{<E_1, \rho> \xrightarrow{e} \text{ Entier} \quad <E_2, \rho> \xrightarrow{e} \text{ Entier}}{<E_1 < E_2> \xrightarrow{e} \text{ Booléen}}$$

2. If-expressions :
   - Syntaxe abstraite

   $$E ::= \ldots \mid E \text{ si } E \text{ alors } E \text{ sinon } E$$

   - Sémantique : on décide que `e2` et `e3` doivent être de même type, qui sera le type final de l'expression :

   $$\frac{<E_1, \rho> \xrightarrow{e} \text{ Booléen} \quad <E_2, \rho> \xrightarrow{e} t \quad <E_3, \rho> \xrightarrow{e} t \quad t \in \{\text{Entier}, \text{Booléen}\}}{< \text{ si } E_1 \text{ alors } E_2 \text{ sinon } E_3, \rho> \xrightarrow{e} t}$$

Exercice 3

Ici on enrichit le langage des commandes :

1. `repeat` :

   $$C ::= \ldots \mid \text{repeat } C \text{ until } E$$

   Pour être correctement typée, cette commande doit avoir la condition de type booléen, et la commande intérieure du `repeat` correctement typée, donc :

   $$\frac{<C, \rho) \xrightarrow{c} \text{ Void} \quad <E, \rho> \xrightarrow{e} \text{ Booléen}}{<\text{repeat } C \text{ until } E, \rho> \xrightarrow{c} \text{ Void}}$$

2. `for` : on suppose que c'est un `for` sur les entiers (on ne traite pas le cas du type énuméré Ada) :

   $$C ::= \ldots \mid \text{for } (i, E_1, E_2, c)$$

   Dans les deux cas, il faut vérifier que E_2 et E_2 sont bien de type Entier, et c est bien de type Void.
   - Cas où la variable i est déclarée dans le for : pour l'évaluation de la commande, il faut prendre en compte ce i entier :

   $$\frac{<E_1, \rho> \xrightarrow{e} \text{ Entier} \quad <E_2, \rho> \xrightarrow{e} \text{ Entier} \quad <c, \rho \cup [i \mapsto \text{Entier}]> \xrightarrow{c} \text{ Void} \quad i \notin Dom(\rho)}{<\text{for } (i, E_1, E_2, c), \rho> \xrightarrow{c} Void}$$

   - Cas où la variable i n'est pas déclarée dans le for : on doit vérifier que i a bien été déclaré et bien de type Entier :

   $$\frac{<E_1, \rho> \xrightarrow{e} \text{ Entier} \quad <E_2, \rho> \xrightarrow{e} \text{ Entier} \quad <c, \rho> \xrightarrow{c} \text{ Void} \quad i \in Dom(\rho) \quad \rho(i) = \text{Entier}}{<\text{for } (i, E_1, E_2, c), \rho> \xrightarrow{c} Void}$$