

# Éléments de correction du TD10 - Sémantique Dynamique

## Exercice 1

La structure des preuves en sémantique dynamique est très similaire à celle de sémantique statique pour le même programme, sauf qu'il faut calculer la valeur de la mémoire à tous stades du programme.

## 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 :

$$\frac{\begin{array}{c} \text{Branche 1} \\ d \xrightarrow{d} \eta \end{array} \quad \begin{array}{c} \text{Branche 2} \\ \langle c, \eta, \emptyset \rangle \xrightarrow{c} \sigma \end{array}}{d; c \rightarrow \sigma} \quad [\text{prog}]$$

Traitions ensuite le cas de la branche 1 (déclarations) :

$$\frac{\begin{array}{c} @1 = \text{newadr}() \\ \hline \text{var } x_1 \xrightarrow{d} \eta_1 = [x_1 \mapsto @1] \end{array} \quad \begin{array}{c} @2 = \text{newadr}() \\ \hline \text{var } x_2 \xrightarrow{d} \eta_2 = [x_2 \mapsto @2] \end{array}}{(\text{var } x_1; \text{var } x_2) \xrightarrow{d} \eta = \eta_1 \cup \eta_2} \quad [\text{seq}]$$

Toutes les feuilles sont des axiomes et les applications des règles sont correctes ssi  $\eta = [x_1 \mapsto @1, x_2 \mapsto @2]$ . Traitions maintenant le cas des commandes avec  $\eta$  comme environnement.

$$\begin{array}{c} \text{axiome ssi } v_1 = 3 \\ \langle 3, \eta, \emptyset \rangle \xrightarrow{e} v_1 \end{array} \quad \begin{array}{c} \text{axiome ssi } v_2 = 1 \\ \langle 1, \eta, \sigma'_0 \rangle \xrightarrow{e} v_2 \\ \hline \langle x_2 := 1, \eta, \sigma'_0 \rangle \xrightarrow{c} \sigma'_1 \end{array} \quad \begin{array}{c} \text{Branche 2b} \\ \langle \text{tq } ec, \eta, \sigma'_1 \rangle \xrightarrow{c} \sigma'_2 \\ \hline \langle \text{tq } ec; \dots, \eta, \sigma'_1 \rangle \xrightarrow{c} \sigma \end{array} \quad \begin{array}{c} \text{Branche 2c} \\ \langle x_2 := 2, \eta, \sigma'_2 \rangle \xrightarrow{c} \sigma \end{array} \quad [\text{seq}]$$

$$\frac{\begin{array}{c} \text{[aff]} \quad \langle x_1 := 3, \eta, \emptyset \rangle \xrightarrow{c} \sigma'_0 \end{array} \quad \begin{array}{c} \langle x_2 := 1; \dots, \eta, \sigma'_0 \rangle \xrightarrow{c} \sigma \end{array}}{\langle x_1 := 3; x_2 := 1; \text{tq } ec; x_2 := 2, \eta, \emptyset \rangle \xrightarrow{c} \sigma} \quad [\text{seq}]$$

On écrit dans un coin au fur et à mesure les indications que l'on récupère avec les applications des règles :

- $\sigma'_0 = \emptyset[@1 \mapsto v_1]$  ;
- $v_1 = 3$ , donc  $\sigma'_0 = [@1 \mapsto 3]$  ;
- $\sigma'_1 = \sigma'_0[@2 \mapsto v_2] = [@1 \mapsto 3, @2 \mapsto 1]$  ;

Réalisons maintenant la branche 2b (la sémantique du test  $x_1 > 0$  n'est pas dans le cours, mais on fait avec!) :

$$\begin{array}{c}
\frac{\langle x_1, \eta, \sigma'_1 \rangle \xrightarrow{e} 3 \quad \langle 2, \eta, \sigma'_1 \rangle \xrightarrow{e} 2}{\langle x_1 - 2, \eta, \sigma'_1 \rangle \xrightarrow{e} v_4} \text{ [moins]} \\
\frac{\text{OK car } \sigma'_1(\eta(x_2)) = 1}{\langle x_2 > 2, \eta, \sigma'_1 \rangle \xrightarrow{e} \text{ff}} \quad \frac{\langle x_1 := x_1 - 2, \eta, \sigma'_1 \rangle \xrightarrow{c} \sigma'_3}{\text{Branche2balpha}} \text{ [aff]} \\
\text{OK car } \sigma'_1(\eta(x_1)) > 0 \quad \frac{\langle x_1 > 0, \eta, \sigma'_1 \rangle \xrightarrow{e} \text{tt}}{\langle \text{si } \dots \text{ alors } \dots \text{ sinon } \dots, \eta, \sigma'_1 \rangle \xrightarrow{c} \sigma'_3} \quad \frac{\langle \text{tq } x_1 > 0 \text{ si } \dots \text{ alors } \dots \text{ sinon } \dots, \eta, \sigma'_3 \rangle \xrightarrow{c} \sigma'_2}{\text{[while]}} \\
\hline
\langle \text{tq } x_1 > 0 \text{ si } \dots \text{ alors } \dots \text{ sinon } \dots, \eta, \sigma'_1 \rangle \xrightarrow{c} \sigma'_2
\end{array}$$

La branche 2balpha est laissée au lecteur. A ce stade, on a accumulé les indications intermédiaires :

- $v_4 = 3 \text{ minus } 2 = 1$  ;
- $\sigma'_3 = \sigma'_1[\text{@}1 \mapsto v_4 = 1]$  ;
- $\sigma'_2 = \sigma'_3[\text{@}1 \mapsto 1 - 2 = -1]$ .

Il reste maintenant la branche 2c :

$$\frac{\langle 2, \eta, \sigma'_2 \rangle \xrightarrow{e} 2}{\langle x_2 := 2, \eta, \sigma'_2 \rangle \xrightarrow{c} \sigma} \text{ [ aff ]},$$

avec l'information suivante :  $\sigma = \sigma'_2[\eta(x_2) \mapsto 2]$ . Et finalement on trouve :  $\sigma \circ \eta : [x_2 \mapsto 2, x_1 \mapsto -1]$ .

## Programme 2

Le déroulement des règles est similaire, sauf que l'on ne peut pas construire d'arbre de preuve fini.

## Exercice 2

1. Expressions : on veut pouvoir effectuer une évaluation paresseuse de  $e_1 \wedge e_2$ , d'où le **etpuis** :

$$\frac{\langle e_1, \eta, \sigma \rangle \xrightarrow{e} \text{ff}}{\langle e_1 \text{ etpuis } e_2, \eta, \sigma \rangle \xrightarrow{e} \text{ff}} \quad \frac{\langle e_1, \eta, \sigma \rangle \xrightarrow{e} \text{tt} \quad \langle e_2, \eta, \sigma \rangle \xrightarrow{e} b}{\langle e_1 \text{ etpuis } e_2, \eta, \sigma \rangle \xrightarrow{e} b}$$

2. Commandes : on fait une première fois la commande  $c$ , et ensuite on teste si on doit continuer (valeur de l'expression  $e$  dans  $\sigma'$ ) :

$$\frac{\langle c, \eta, \sigma \rangle \xrightarrow{c} \sigma' \quad \langle e, \eta, \sigma' \rangle \xrightarrow{e} \text{tt}}{\langle \text{repete } c \text{ jusqu'a } e, \eta, \sigma \rangle \xrightarrow{c} \sigma'} \\
\frac{\langle c, \eta, \sigma \rangle \xrightarrow{c} \sigma' \quad \langle e, \eta, \sigma' \rangle \xrightarrow{e} \text{ff} \quad \langle \text{repete } c \text{ jusqu'a } e, \eta, \sigma' \rangle \xrightarrow{c} \sigma''}{\langle \text{repete } c \text{ jusqu'a } e, \eta, \sigma \rangle \xrightarrow{c} \sigma''}$$

3. Déclarations : Il faut calculer ce que vaut l'expression  $e$  dans l'environnement courant, et ensuite mettre cette expression à la nouvelle adresse, pour cela on s'aperçoit que l'on doit maintenant se rappeler de la mémoire courante, et de l'environnement aussi, et on modifie à la fois la valeur de l'environnement et la valeur de la mémoire :

$$\frac{ad = \text{newadr}() \quad \langle e, \eta, \sigma \rangle \xrightarrow{e} v}{(\text{var } x : \text{Entier} := e, \eta, \sigma) \xrightarrow{d} ([x \mapsto ad], \sigma[ad \mapsto v])}$$

Il faudrait donc changer la configuration des déclarations pour prendre en compte cette nouvelle sémantique, et aussi prendre en compte ceci lors de la règle des blocs/ règle du programme.

## Exercice 3

Flemme du correcteur, on trouve à la fin :  $\sigma \circ \eta : [x_2 \mapsto 1, x_1 \mapsto 4]$ . Attention à la sortie du bloc les variables sont restaurées.