

# CONCEPTION DE LANGAGE

## Examen – Fév. 2008

### Affectation multiple

On souhaite disposer d'une affectation de la forme

$$\mathbf{x}_1, \dots, \mathbf{x}_n := \mathbf{e}_1, \dots, \mathbf{e}_n$$

où  $\mathbf{x}_1, \dots, \mathbf{x}_n$  sont des variables et  $\mathbf{e}_1, \dots, \mathbf{e}_n$  des expressions. On se place dans le cadre où la fonction sémantique pour les instruction a pour signature :

$$\mathbf{S} : \text{STAT} \rightarrow \text{Env} \rightarrow \text{Mem} \rightarrow \text{Mem}$$

(ie. pas de continuation).

1. donner une définition de la fonction sémantique  $\mathbf{S}$  pour ce nouveau cas :

$$\mathbf{S}[[\mathbf{x}_1, \dots, \mathbf{x}_n := \mathbf{e}_1, \dots, \mathbf{e}_n]](r, m) = \dots$$

2. en utilisant *votre définition* de la fonction sémantique  $\mathbf{S}$ , donnez les résultats de l'évaluation (valeurs  $m$  pour  $\mathbf{x}$  et  $\mathbf{y}$ ) dans les cas suivants :

(a)  $\mathbf{x} := 1; \mathbf{y} := 2; \mathbf{x}, \mathbf{y} := \mathbf{y}, \mathbf{x}$

(b)  $\mathbf{x} := 1; \mathbf{y} := 2; \mathbf{x}, \mathbf{y} := 3, \mathbf{x}$

(c)  $\mathbf{x} := 1; \mathbf{x}, \mathbf{x} := 2, 3$

(d)  $\mathbf{x} := 1; \mathbf{x}, \mathbf{x} := 2, \mathbf{x}$

(e)  $\mathbf{x} := 1; \mathbf{x}, \mathbf{x} := \mathbf{x}, 2$

### Une boucle

Soit l'équation sémantique (toujours sans continuation)

$$\begin{aligned} \mathbf{S}[[\text{loop on } e \text{ with } s \text{ done}]](r, m) = & \\ (!w. \lambda m'. & \\ \text{case } \mathbf{E}[[e]](r, m') : & \\ \text{inBool(true)} \rightarrow m' & \\ | \text{inBool(false)} \rightarrow w (\mathbf{S}[[s]](r, m')) & \\ m) & \end{aligned}$$

où `loop on with` et `done` sont des mots clef,  $e$  est une expression booléenne et  $s$  une instruction.

1. donnez le résultat de l'évaluation (valeur de  $m$  en  $\mathbf{x}$  et  $\mathbf{y}$ ) dans les cas suivants

(a)  $\mathbf{x}:=1; \mathbf{y}:=3; \text{loop on } (\mathbf{x}=\mathbf{y}) \text{ with } \mathbf{x}:=\mathbf{x}+1; \mathbf{y}:=\mathbf{y}-1 \text{ done}$

(b)  $\mathbf{x}:=3; \mathbf{y}:=1; \text{loop on } (\mathbf{x}=\mathbf{y}) \text{ with } \mathbf{x}:=\mathbf{x}+1; \mathbf{y}:=\mathbf{y}-1 \text{ done}$

## Langage de gardes

On définit le langage de programmation réactive suivant.

Les programmes manipulent deux sortes de variables :

- des variables de contrôles qui peuvent avoir deux états : actif ou non ;
- des variables de calcul destinées à contenir des valeurs (pour faire simple, des entiers)

Un programme est constitué de deux parties

1. une suite de déclarations, avec initialisation des variables de contrôle et de calcul ;
2. une boucle infinie dont le corps contient une suite «d'actions gardées».

Une «action gardée» est essentiellement constituée d'une variable de contrôle et d'une suite d'affectations.

Il y a deux sortes d'affectations : celles modifiant l'état d'une variable de contrôle et celles modifiant la valeur d'une variable de calcul.

Le comportement attendu de la boucle des programmes réactifs est la suivante : à chaque itération de la boucle **toutes** les actions dont la variable de contrôle est active sont exécutées. On demande que le contexte de déclenchement et d'exécution d'une action soit celui présent au début de la boucle.

Exemple : à l'initialisation  $c1$  est faux,  $c2$  et  $c3$  sont vrais,  $i$  vaut 0 et les valeurs de  $x$  et  $y$  sont indifférentes. La boucle effectue les actions gardées suivantes :

$$\begin{aligned}c1 &\Rightarrow z := x; x := y; y := x; i := 0; c1 := false; c2 := true; c3 := true \\c2 &\Rightarrow i := i + 1 \\c3 &\Rightarrow c1 := (i = 10); c2 := false\end{aligned}$$

Ce programme échange les valeurs de  $x$  et  $y$  tous les 10 tours de boucle.

1. définissez une syntaxe pour ce langage
2. définissez sa sémantique