Lenguajes y Compiladores - Guía 4 - Año 2013

Contenidos: Semántica denotacional del lenguaje imperativo simple. *Recordar* que utilizamos tipografía sans-serif (como en x) para variables concretas y tipografía serif (como en e) para meta-variables.

(1) Utilizar la semántica denotacional para demostrar o refutar las siguientes equivalencias:
 (a) $c; \mathbf{skip} \;\equiv\; c$
 (b) $c_1; (c_2; c_3) \equiv (c_1; c_2); c_3$
 (c) $(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1); c_2 \;\equiv\; \mathbf{if}\ b\ \mathbf{then}\ c_0; c_2\ \mathbf{else}\ c_1; c_2$
 (d) $c_2; (\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1) \;\equiv\; \mathbf{if}\ b\ \mathbf{then}\ c_2; c_0\ \mathbf{else}\ c_2; c_1$
 (e) $\mathsf{x} := \mathsf{y}\ ;\ \mathsf{z} := \mathsf{y} \;\equiv\; \mathsf{z} := \mathsf{y}\ ;\ \mathsf{x} := \mathsf{y}$

(2) Utilizar la semántica denotacional para demostrar o refutar las siguientes equivalencias:
 (a) $\mathbf{newvar}\ \mathsf{x} := e\ \mathbf{in}\ \mathbf{skip} \;\equiv\; \mathbf{skip}$
 (b) $\mathbf{newvar}\ \mathsf{x} := e\ \mathbf{in}\ \mathsf{y} := \mathsf{x} \;\equiv\; \mathsf{y} := e$
 (c) $\mathbf{newvar}\ \mathsf{x} := e_1\ \mathbf{in}\ (\mathbf{newvar}\ \mathsf{y} := e_2\ \mathbf{in}\ c) \;\equiv$
 $\mathbf{newvar}\ \mathsf{y} := e_2\ \mathbf{in}\ (\mathbf{newvar}\ \mathsf{x} := e_1\ \mathbf{in}\ c)$

(3) Teniendo en cuenta los ejercicios anteriores, discuta en grupo las siguientes afirmaciones:
 (a) El parser puede eliminar toda ocurrencia de **skip**.
 (b) El parser puede elegir inclinar las secuencias de más de dos comandos hacia la derecha o hacia la izquierda.

(4) Considere el comando **while true do** $x := x - 1$
 (a) Dar la función F que define su semántica. Calcular la expresión más sencilla que pueda para F.
 (b) Existe algún n tal que $F^n\ \perp_{\Sigma \to \Sigma_\perp}$ no sea idénticamente $\perp$?
 (c) Considere la cadena en $\Sigma \to \Sigma_\perp$ dada por

$$\omega_i\ \sigma \;=\; \begin{cases} \sigma & \text{si } 0 \le \sigma x \le i \\ \perp & \text{caso contrario} \end{cases}$$

 Es sabido que la continuidad de F garantiza la igualdad $F\left(\bigsqcup \omega_i\right) = \bigsqcup F\omega_i$. Compruebe la misma calculando cada miembro de la igualdad para el caso de la cadena dada.

(5) Calcule la semántica denotacional de los siguientes comandos:
 (a) $\mathbf{while}\ \mathsf{x} < 2\ \mathbf{do}\ \mathbf{if}\ \mathsf{x} < 0\ \mathbf{then}\ \mathsf{x} := 0\ \mathbf{else}\ \mathsf{x} := \mathsf{x} + 1$
 (b) $\mathbf{while}\ \mathsf{x} < 2\ \mathbf{do}\ \mathbf{if}\ \mathsf{y} = 0\ \mathbf{then}\ \mathsf{x} := \mathsf{x} + 1\ \mathbf{else}\ \mathbf{skip}$

(6) Suponga que $[\![\textbf{while } b \textbf{ do } c]\!]\sigma \neq \perp$.
   (a) Demuestre que existe $n \geq 0$ tal que $F^n \perp \sigma \neq \perp$.
   (b) Demuestre que si $\sigma' = [\![\textbf{while } b \textbf{ do } c]\!]\sigma$ entonces $\neg[\![b]\!]\sigma'$

(7) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:
   (a) **while false do** $c \equiv$ **skip**
   (b) **while** b **do** $c \equiv$ **while** b **do** $(c; c)$
   (c) (**while** b **do** c) ; **if** b **then** c_0 **else** $c_1 \equiv$ (**while** b **do** c) ; c_1

(8) Considerar las siguientes definiciones como syntactic sugar del comando **for** $v := e_0$ **to** e_1 **do** c:
   (a) $v := e_0;$ **while** $v \leq e_1$ **do** $c; v := v + 1$.
   (b) **newvar** $v := e_0$ **in while** $v \leq e_1$ **do** $c;\ v := v + 1$.
   (c) **newvar** $w := e_1$ **in**
   newvar $v := e_0$ **in while** $v \leq w$ **do** $c; v := v + 1$

   ¿Hay alguna que pueda considerarse satisfactoria? Justificar.

(9) Enunciar de manera completa el teorema de coincidencia y demostrar el caso **while**.

(10) Definir la sintaxis y la semántica denotacional del comando **repeat** c **until** b satisfaciendo

   repeat c **until** $b \ \equiv$
   $c;$ **if not** b **then** (**repeat** c **until** b) **else skip**

   Luego probar o refutar su equivalencia con el comando
   $c;$ **while not** b **do** c.

(11) Usando el Teorema de coincidencia para comandos probar que para todo par de comandos c_0, c_1, si

$$FV\ c_0 \cap FA\ c_1 = FV\ c_1 \cap FA\ c_0 = \emptyset,$$

entonces $[\![c_0; c_1]\!] = [\![c_1; c_0]\!]$.

(12) Considere los siguientes comandos:

$$\begin{aligned} c_0 &\doteq \textbf{newvar } \mathsf{x} := \mathsf{x} + \mathsf{y} \textbf{ in} \\ &\qquad c; \textbf{while } \mathsf{x} > 0 \textbf{ do } \mathsf{y} := \mathsf{y} + 1;\ \mathsf{x} := \mathsf{x} - 1 \\ c_1 &\doteq \textbf{newvar } \mathsf{y} := \mathsf{x} + \mathsf{z} \textbf{ in} \\ &\qquad c; \textbf{while } \mathsf{y} > 0 \textbf{ do } \mathsf{z} := \mathsf{z} + 1;\ \mathsf{y} := \mathsf{y} - 1 \end{aligned}$$

Asuma que c es un comando que satisface $(FV\ c) \cap \{\mathsf{x}, \mathsf{y}, \mathsf{z}\} = \emptyset$. Formule la relación que existe entre estos comandos (vistos como funciones que transforman estados), y pruebe tal relación sin calcular semántica.