Lenguajes y Compiladores - Trabajo práctico 6 - Año 2012

Contenidos: Semantica de Transiciones, Relación entre la semántica operacional y denotacional.

(1) Dados los programas
   (1) $\mathsf{y} := \mathsf{x} + \mathsf{y}$; **if** $\mathsf{y} > 0$ **then** $\mathsf{x} := \mathsf{x} - 1$ **else skip**
   (2) **while** $\mathsf{x} > 0$ **do**
   $\mathsf{y} := \mathsf{x} + \mathsf{y}$; **if** $\mathsf{y} > 0$ **then** $\mathsf{x} := \mathsf{x} - 1$ **else skip**

   Computar la semántica operacional en σ tal que $\sigma\mathsf{x} = 2, \sigma\mathsf{y} = 1$.

(2) Dado el programa $P \doteq$

newvar $\mathsf{x} := 2$ **in**
   while $\mathsf{x} > 0$ **do**
   $\mathsf{y} := \mathsf{x} + \mathsf{y}$; **if** $\mathsf{y} > 0$ **then** $\mathsf{x} := \mathsf{x} - 1$ **else skip**

   Computar la semántica operacional de P en un estado arbitrario.

(3) Pruebe:
   (a) Si $\langle c_0, \sigma\rangle \to^* \sigma'$, entonces $\langle c_0; c_1, \sigma\rangle \to^* \langle c_1, \sigma'\rangle$
   (b) Si $\langle c, [\sigma|\mathsf{x} : [\![e]\!]\sigma]\rangle \to^* \sigma'$, entonces

$$\langle \textbf{newvar } \mathsf{x} := e \textbf{ in } c, \sigma\rangle \to^* [\sigma'|\mathsf{x} : \sigma\mathsf{x}].$$

   (c) Si $\langle c, [\sigma|\mathsf{x} : [\![e]\!]\sigma]\rangle \to^* \langle c', \sigma'\rangle$, entonces

$\langle \textbf{newvar } \mathsf{x} := e \textbf{ in } c, \sigma\rangle \to^* \langle \textbf{newvar } \mathsf{x} := \sigma' x \textbf{ in } c', [\sigma'|\mathsf{x} : \sigma x]\rangle$

(4) Dado el programa $P \equiv$

newvar $\mathsf{x} :=$ y + x **in**
   while $\mathsf{x} > 0$ **do**
   ?x;
   y:= x+y;
   !(y-1);
   if $\mathsf{x} > 0$ **then fail else skip**

   (a) Determinar (sin calcular la semántica) estados σ y σ' tales que P sea equivalente a **skip** en σ y diferente a **skip** en σ'.
   (b) Computar la semántica operacional de P en ambos estados.

(5) Demostrar la corrección de la semántica operacional one-step (incluyendo **fail**) respecto de la semántica denotacional, demostrando simultáneamente
   (a) $\langle c, \sigma\rangle \to \sigma' \Longrightarrow [\![c]\!]\sigma = \sigma'$.

(b) $\langle c, \sigma\rangle \to \langle \mathbf{abort}, \sigma'\rangle \Longrightarrow [\![c]\!]\sigma = \langle \mathbf{abort}, \sigma'\rangle$.
(c) $\langle c, \sigma\rangle \to \langle c', \sigma'\rangle \Longrightarrow [\![c]\!]\sigma = [\![c']\!]\sigma'$.
Use inducción en la derivación.

(6) Demostrar simultáneamente:
(a) $\langle c, \sigma\rangle \to^* \sigma' \Longleftarrow [\![c]\!]\sigma = \sigma'$.
(b) $\langle c, \sigma\rangle \to^* \langle \mathbf{abort}, \sigma'\rangle \Longleftarrow [\![c]\!]\sigma = \langle \mathbf{abort}, \sigma'\rangle$.
Use inducción en la estructura de c. Tendrá que usar los resultados probados en el ejercicio 3. El caso **while** requiere a su vez una inducción en el mínimo n tal que $F^n(\bot)\sigma \neq \bot$.

(7) Demostrar la equivalencia entre la semántica denotacional y la operacional del lenguaje imperativo simple incluyendo **fail**. Debe salir inmediatamente de los dos ejercicios anteriores.

NOTA: La extensión del lenguaje imperativo con las construcciones **fail** l *y* **catch** l **in** c_0 **with** c_1 *es definida en los ejercicios 5.3 y 5.4 del Reynolds.*

(8) Para el lenguaje con **while**, input, output, **fail** l y **catch** l **in** c_0 **with** c_1, dar la semántica de transiciones. Para esto previamente tendrá que cambiar el conjunto de configuraciones terminales.