

Lenguajes y Compiladores - Trabajo práctico 7 - Año 2011

Temas: Semántica de Transiciones, Relación entre la semántica operacional y denotacional.

(1) Dados los programas

(1) $y := x + y$; **if** $y > 0$ **then** $x := x - 1$ **else skip**

(2) **while** $x > 0$ **do**

$y := x + y$; **if** $y > 0$ **then** $x := x - 1$ **else skip**

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

(2) Dado el programa $P \doteq$

newvar $x := 2$ **in**

while $x > 0$ **do**

$y := x + y$; **if** $y > 0$ **then** $x := x - 1$ **else skip**

Computar la semántica operacional de P en un estado arbitrario.

(3) Pruebe:

(a) Si $\langle c_0, \sigma \rangle \rightarrow^* \sigma'$, entonces $\langle c_0; c_1, \sigma \rangle \rightarrow^* \langle c_1, \sigma' \rangle$

(b) Si $\langle c, [\sigma|x : \llbracket e \rrbracket \sigma] \rangle \rightarrow^* \sigma'$, entonces

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

(c) Si $\langle c, [\sigma|x : \llbracket e \rrbracket \sigma] \rangle \rightarrow^* \langle c', \sigma' \rangle$, entonces

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

(4) Dado el programa $P \equiv$

newvar $x := y + x$ **in**

while $x > 0$ **do**

? x ;

$y := x + y$;

!($y - 1$);

if $x > 0$ **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) $\langle c, \sigma \rangle \rightarrow \sigma' \implies \llbracket c \rrbracket \sigma = \sigma'$.

- (b) $\langle c, \sigma \rangle \rightarrow \langle \mathbf{abort}, \sigma' \rangle \implies \llbracket c \rrbracket \sigma = \langle \mathbf{abort}, \sigma' \rangle$.
 (c) $\langle c, \sigma \rangle \rightarrow \langle c', \sigma' \rangle \implies \llbracket c \rrbracket \sigma = \llbracket c' \rrbracket \sigma'$.

Use inducción en la derivación.

- (6) Demostrar simultáneamente:
 (a) $\langle c, \sigma \rangle \rightarrow^* \sigma' \iff \llbracket c \rrbracket \sigma = \sigma'$.
 (b) $\langle c, \sigma \rangle \rightarrow^* \langle \mathbf{abort}, \sigma' \rangle \iff \llbracket c \rrbracket \sigma = \langle \mathbf{abort}, \sigma' \rangle$.
 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 $F^n(\perp)\sigma \neq \perp$.
- (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.
- (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. (Tal extensión fue definida en los ejercicios 5.3 y 5.4 del Reynolds, a los cuales nos referimos en el práctico anterior.)
- (9) Agregar a este lenguaje el comando **repeat** definiendo su semántica de transiciones.