

Lenguajes y Compiladores - Trabajo práctico 5 - Año 2013

Temas: Fallas, Output, Dominios Recursivos e Input.

Recordar que los símbolos en cursiva representan metavariables. Por ejemplo, si en un ejercicio involucra los símbolos v y w , debe considerarse el caso en que ambas metavariables denoten la misma variable concreta.

- (1) Considerando el lenguaje con fallas y output, de un programa para cada posible comportamiento:
- (a) cantidad finita de output y luego divergencia,
 - (b) cantidad finita de output y luego falla,
 - (c) cantidad finita de output y luego terminación,
 - (d) cantidad infinita de output.

- (2) Dado el programa $P \equiv$

```
newvar  $x := y + x$  in  
  while  $x > 0$  do ! $x$ ; if  $x > 0$  then skip else fail
```

Caracterice (sin calcular la semántica) los estados σ en los cuales P se comporta como **skip**.

- (3) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:

- (a) $c; \mathbf{while\ true\ do\ skip} \equiv \mathbf{while\ true\ do\ skip}$.
- (b) $c; \mathbf{fail} \equiv \mathbf{fail}$.
- (c) $\mathbf{newvar\ } v := e \mathbf{\ in\ } v := v + 1; \mathbf{fail} \equiv$
 $\mathbf{newvar\ } w := e \mathbf{\ in\ } w := w + 1; \mathbf{fail}$.
- (d) $\mathbf{while\ } b \mathbf{\ do\ fail} \equiv \mathbf{if\ } b \mathbf{\ then\ fail\ else\ skip}$.

Considere el punto (a) para el lenguaje sin fallas: ¿la respuesta es la misma que para el lenguaje con fallas?

- (4) Dado el programa $\mathbf{while\ } x > 0 \mathbf{\ do\ } !x; c$, calcule su semántica denotacional, considerando los casos

- (a) $c \equiv \mathbf{if\ } x > 0 \mathbf{\ then\ skip\ else\ fail}$
- (b) $c \equiv \mathbf{if\ } x > 0 \mathbf{\ then\ fail\ else\ skip}$

- (5) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:

- (a) $?x; ?y \equiv ?y; ?x$.
- (b) $?x; z := x \equiv ?z$.
- (c) $\mathbf{newvar\ } x := e \mathbf{\ in\ } (?x; z := x) \equiv ?z$.

(6) Describa mediante un diagrama de Hasse las relaciones de orden que se establecen entre los siguientes elementos de Ω :

- $\iota_{in}(\lambda n. \iota_{out}(n, \perp))$
- $\iota_{out}(0, \perp)$,
- $\iota_{in}(\lambda n. \perp)$,
- $\iota_{in}(f)$, donde $f n = \begin{cases} \perp & \text{si } n < 0 \\ \iota_{out}(n, \perp) & \text{caso contrario} \end{cases}$

(7) Dé un programa, y justifique su elección, cuya semántica sea el supremo de la cadena:

$$w_0 = \perp, \quad w_{i+1} = \iota_{in}(\lambda n. \iota_{out}(n, w_i))$$

(8) Considere los programas de la forma **while true do** ($?x; c$). La cadena $F^i \perp \sigma$ de la semántica del **while**, ¿será siempre una cadena interesante en Ω ? Justifique su respuesta.

(9) Dado el programa $P \equiv$

```
newvar x := x + 1 in
  while x > 0 do ?x; if y > 0 then fail else !x
```

(a) Calcular la semántica denotacional de P , en un estado σ tal que $\sigma y > 0$.

(b) Considere el caso $\sigma y \leq 0$. Calcule $F \perp$ y $F^2 \perp$. Puede dar una expresión general para $\mathbf{Y}_{\Sigma \rightarrow \Sigma \perp} F$.

(10) Demostrar o refutar las siguientes equivalencias usando semántica denotacional

(a) $?x; \mathbf{while} \ b \ \mathbf{do} \ !x; ?x \ \mathbf{od}; !x \equiv \mathbf{while} \ b \ \mathbf{do} \ ?x; !x \ \mathbf{od}.$

(b) $?x; \mathbf{while} \ b \ \mathbf{do} \ !x; ?x \ \mathbf{od}; !x \equiv ?x; !x; \mathbf{while} \ b \ \mathbf{do} \ ?x; !x \ \mathbf{od}.$