Lenguajes y Compiladores. Guía del 25/04/2017

Objetivos. Comprender las construcciones necesarias para dar semántica a output e input. Entender el orden subyacente a productos de dominios y uniones disjuntas de predominios. Poder determinar si dado un elemento d del dominio semántico existe un programa cuya denotación sea d.

Repaso.

(1) Decida si la función $H\colon (\Sigma \to \Sigma'_\perp) \to (\Sigma \to \Sigma'_\perp)$ es monótona y si es continua.

$$H\, f\, \sigma = \begin{cases} \sigma & \text{si } f = \perp \\ \perp & \text{en caso contrario} \end{cases}$$

(2) Considere la función $G\colon (\Sigma \to \Sigma'_\perp) \to (\Sigma \to \Sigma'_\perp)$ definida por

$$G\, f\, \sigma = \begin{cases} \perp & \text{si } f \text{ es parcial} \\ \sigma & \text{en caso contrario} \end{cases}$$

   (a) Decida si G es continua o no.
   (b) Decida si existe un comando **while** b **do** c, tal que $G = F$, donde F es el funcional asociado al **while**.

Ejercicios.

(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 **while** $\mathsf{x} > 0$ **do** $!\mathsf{x}; c$, calcule su semántica denotacional, considerando los casos
   (a) $c \equiv$ **if** $\mathsf{x} > 0$ **then skip else fail**
   (b) $c \equiv$ **if** $\mathsf{x} > 0$ **then fail else skip**

(3) Demostrar o refutar las siguientes equivalencias usando semántica denotacional:
   (a) $?x; ?y \equiv ?y; ?x$.
   (b) $?x; z := x \equiv ?z$.
   (c) **newvar** $x := e$ **in** $(?x; z := x) \equiv ?z$.

(4) Sea c un progama que no incluya fallas, outputs, ni inputs tal que $\{\mathsf{x}, \mathsf{y}\} \cap FV(c) = \emptyset$. Determine si es válida la siguiente igualdad:

$$?\,\mathsf{x}\,;\, c\,;\, !\mathsf{x} \equiv ?\,\mathsf{y}\,;\, c\,;\, !\mathsf{y}$$

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

- $\iota_{in}(\lambda n.\ \iota_{out}(n, \perp))$

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

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

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

(7) 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.

(8) 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 $\mathbf{Y}_{\Sigma \to \Sigma_\bot} F$.

(9) Demostrar o refutar las siguientes equivalencias usando semántica denotacional
(a) ?x; **while** b **do** !x; ?x **od**; !x $\equiv$ **while** b **do** ?x; !x **od**.
(b) ?x; **while** b **do** !x; ?x **od**; !x $\equiv$?x; !x; **while** b **do** ?x; !x **od**.