

## Lenguajes y Compiladores - Trabajo práctico 10 - Año 2012

**Contenidos:** *Semántica de Continuaciones, Relación entre la semántica directa y de continuaciones.*

- (1) Calcular de manera detallada la semántica de continuaciones de
  - (1) **newvar**  $x := 1$  **in skip**;  $y := x + y$
  - (2)  $x := 0$ ; **while**  $x < 2$  **do**  $x := x + 1$  **od**.
- (2) Para el lenguaje imperativo simple, se quiere demostrar por inducción estructural que para todo  $c, \sigma, \kappa, \llbracket c \rrbracket^c \kappa \sigma = \kappa_{\perp} (\llbracket c \rrbracket^d \sigma)$ . Desarrollar los casos para **skip**, **if**, asignación y **newvar**.
- (3) Dado el programa  $c \doteq$   
**while**  $x \neq 0$  **do if**  $x = 1$  **then fail** **else**  $x := x - 2$ 
  - (a) Calcular la semántica directa de  $c$ .
  - (b) Calcular la semántica de continuaciones de  $c$  en un estado  $\sigma$  tal que  $\sigma x$  es par, tomando continuaciones  $\kappa$  y  $\kappa_f$ .
  - (c) Calcular la semántica de continuaciones de  $c$  en un estado  $\sigma$  tal que  $\sigma x$  sea impar, tomado las continuaciones  $\kappa$  y  $\kappa_f$ .
  - (d) ¿Se puede determinar los estados  $\sigma$  tales que  $\llbracket c \rrbracket \kappa \kappa_f \sigma$  es distinto de  $\perp$ ?
- (4) (a) Agregar al lenguaje imperativo simple el comando  
**repeat**  $c$  **until**  $b$   
definiendo su semántica de continuaciones.  
(b) Calcular la semántica de continuaciones del programa  
**repeat if**  $x = 1$  **then fail** **else**  $x := x - 2$  **until**  $x = 0$
- (5) Calcular de manera detallada la semántica de continuaciones de los programas:
  - (a)  $?w; v := v + 1; !w$ .
  - (b)  $v := 0$ ; **while**  $v < 2$  **do**  $?w; v := v + 1; !w$  **od** .
- (6) Completar la prueba del ejercicio 2 para el caso **while**.
- (7) Agregar al lenguaje imperativo simple el comando **break**, que cuando se ejecuta fuerza la salida del cuerpo del **while** más próximo en que se encuentre (no sale de varios anidados). Dar la semántica de continuaciones. (Extra: extender la semántica de **break** de manera que salga de **repeat** y **for**.)
- (8) Adaptar la prueba de la equivalencia siguiente a la semántica de continuaciones.  
**repeat**  $c$  **until**  $b \equiv_{\text{comm}} c$ ; **while not**  $b$  **do**  $c$ .