

Apellido y Nombre:

email:

nota
------

1	2	3	4	5	6
---	---	---	---	---	---

Lenguajes y Compiladores

Parcial 1

15/4/2010

1. Demostrar la propiedad de composicionalidad para el cálculo de predicados. Esta propiedad dice que se pueden reemplazar subfrases por otras de igual significado. En símbolos:

Si para toda  $w \in FV(\varphi)$  se tiene  $\llbracket \delta w \rrbracket = \llbracket \delta' w \rrbracket$ , entonces  $\llbracket \varphi / \delta \rrbracket = \llbracket \varphi / \delta' \rrbracket$

2. Considere el programa

**while**  $x \neq 1$  **do if**  $x > 0$  **then**  $x := x - 2$  **else**  $x := -x$

- a) Dé explícitamente la  $F$  asociada al while.
  - b) Calcule  $F^k \perp$  para  $k = 0, 1, 2, 3$ .
  - c) Dé explícitamente  $F^k \perp$ .
  - d) Calcule la semántica denotacional del programa.
3. Demuestre:
    - a)  $f \in (P \rightarrow D)$  es continua si y sólo si  $f_{\perp} \in (P_{\perp} \rightarrow D)$  es continua.
    - b) Si  $f : P' \rightarrow P''$  es continua entonces

$$f \circ \_ \in (P \rightarrow P') \rightarrow (P \rightarrow P'')$$

también es continua.

4. Demostrar o refutar

- a) **newvar**  $x := e$  **in**  $?x; z := x; !e \equiv ?z; !e$ .
- b) **newvar**  $x := e$  **in**  $?x; z := x \equiv ?z$ .

5. Se agregan los comandos **catch**  $\langle \text{var} \rangle$  **in**  $\langle \text{comm} \rangle$  **with**  $\langle \text{comm} \rangle$  y **fail**  $\langle \text{var} \rangle$ . Definir la semántica de **catch**  $l$  **in**  $c_0$  **with**  $c_1$  y **fail**  $l$  para el lenguaje imperativo simple con fallas. El comando **fail**  $l$  causa una falla con label  $l$ . El comando **catch**  $l$  **in**  $c_0$  **with**  $c_1$  causa la ejecución de  $c_0$  y en el caso de que  $c_0$  produzca una falla con label  $l$ , causa la ejecución de  $c_1$ . Modificar adecuadamente el dominio semántico.

6. Probar que para todo  $\sigma$  tal que  $\neg \llbracket b \rrbracket \sigma$  se tiene la igualdad:

$$\llbracket \text{while } b \text{ do } c \rrbracket \sigma = \sigma$$