

Tarea III (15 puntos)

A continuación encontrará 4 preguntas, cada una compuesta de diferentes sub-preguntas. El valor de cada pregunta (y sub-pregunta) estará expresado entre paréntesis al inicio de las mismas. La entrega se realizará por correo electrónico a rmonascal@gmail.com hasta las 11:59pm. VET del Jueves 17 de Julio de 2025.

1. (4 puntos) Considere el siguiente programa:

```
for (i = 0 ; i < n ; i++) {  
    t1 = foo(0, 1);  
    t2 = bar(t1);  
    t3 = foo(t2, 1);  
    X[i] = t1 * (t2 + t3)  
}  
  
int foo(int a, int b) {  
    return a + b * bar(a);  
}  
  
int bar(int a) {  
    return a * 2 + 1;  
}
```

- a) (1 punto) Construya el grafo de llamadas para el programa mostrado.
 - b) (1.5 puntos) Construya un programa equivalente usando análisis de contexto *por clonación* y vuelva a crear el grafo de llamadas para este nuevo programa.
 - c) (1.5 puntos) Usando el último programa, realice propagación de constantes en su programa de forma que obtenga una versión mejorada. Muestre los resultados de cada iteración.
2. (3 puntos) Considere el problema de análisis de flujo definido en la pregunta 1 de la Tarea II, en el que queríamos saber qué variables en un programa son potencialmente iguales a cero y, a partir de ahí, calcular cuáles expresiones tienen peligro de estar indefinidas por ser una división con denominador igual a cero.
Se desea que reescriba las reglas planteadas como un programa de Datalog.

3. (6 puntos) Considere el modelo de lenguaje de programación imperativo IMP discutido en el libro de texto “*The Formal Semantics of Programming Languages*” de Winskel, con la categoría sintáctica de instrucciones

$$\begin{array}{lcl}
 I & \leftarrow & skip \\
 & | & id := E \\
 & | & I_0; I_1 \\
 & | & \text{if } B \text{ then } I_0 \text{ else } I_1 \\
 & | & \text{while } B \text{ do } I_0
 \end{array}$$

donde E corresponde a la categoría sintáctica de las expresiones aritméticas y B corresponde a la categoría sintáctica de las expresiones booleanas.

Suponga que se crea un nuevo tipo de expresión condicional: el *switch-case*, con la siguiente forma:

$$\begin{array}{lcl}
 I & \leftarrow & \dots \\
 & | & \text{switch } E \langle \text{case } E_i \text{ do } I_i \mid 0 \leq i < n \rangle
 \end{array}$$

donde n es la cantidad de posibles opciones dentro de la instrucción y $\langle \rangle$ es notación de secuencias.

Informalmente, la semántica de dicha instrucción consta de evaluar la expresión asociada al *switch* y ejecutar la instrucción I_i de la primera instrucción E_i (leídas de izquierda a derecha) cuya evaluación sea igual a la de E .

- (2 puntos) Defina reglas para Semántica Operacional de la nueva instrucción usando “*pasos largos*”.
- (2 puntos) Defina reglas para Semántica Operacional de la nueva instrucción usando “*pasos cortos*”.
- (2 puntos) Defina una Semántica Denotacional de la nueva instrucción (puede usar notación de conjuntos/relaciones o funcional).

Puede suponer que ya las expresiones e instrucciones del lenguaje IMP tienen definidas sus semánticas, tanto de pasos largos como cortos. Puede usar cuantificadores universales y existenciales como componentes de su inferencia, si le es conveniente.

4. (2 puntos) Última pregunta de la última tarea del trimestre. ¿Realmente hace falta un enunciado?



Construya **un buen meme** que tenga que ver con el contenido del curso.

Puede ser, por ejemplo, sobre alguno de los temas que vimos o sobre la experiencia en general de la materia. Lo que sea que los inspire y les parezca cómico.