

Recuperatorio del segundo parcial

NOTA: este parcial es a libro abierto. Se permite tener cualquier material manuscrito o impreso, pero no se permite el uso de dispositivos electrónicos. El parcial se califica con una nota numérica de 1 a 10. Se requiere ≥ 4 en ambos parciales para aprobar la materia. Para promocionar se requiere nota ≥ 6 en ambos parciales y promedio ≥ 7 .

Ejercicio 1. Calcular el número de Ershov de la siguiente expresión E :

$$(a * (b * c)) + ((d - e) / (f - g))$$

Escribir código para una máquina de registros para evaluar la expresión E utilizando la mínima cantidad de registros posible, de acuerdo con el número de Ershov calculado.

Ejercicio 2. Considerar una máquina de pila con las siguientes instrucciones:

```
type Programa = [Instruccion]
data Instruccion = Push Int
                  | Add
                  | Label String
                  | JumpIfZero String
```

con la siguiente semántica:

- La instrucción **Push** n mete la constante numérica n en la pila.
- La instrucción **Add** saca los dos elementos del tope de la pila y mete la suma en la pila.
- La instrucción **Label** e corresponde a la declaración de una etiqueta e , que no tiene ningún efecto sobre la pila.
- La instrucción **JumpIfZero** e saca el elemento del tope de la pila. Si dicho número es 0, salta a la instrucción donde se encuentra definida la etiqueta e . De lo contrario, el control sigue hacia la siguiente instrucción. Suponemos que siempre que hay una instrucción **JumpIfZero** e hay una (y solamente una) declaración de la etiqueta e en el programa.

Si la pila está vacía, se considera que el número en el tope es 0. Definir la función:

```
eval :: Programa -> Int
```

que ejecuta el programa en una pila vacía y devuelve el número que queda en el tope de la pila después de ejecutar el programa.

Ejercicio 3. Considerar la siguiente implementación del algoritmo de búsqueda binaria:

```
a = 0
b = n
while a + 1 < b {
  c = (a + b) / 2
  if y[c] <= x {
    a = c
  } else {
    b = c
  }
}
```

- Compilarlo para una máquina de registros. Usar la instrucción `load(r_1 , r_2 , r_3)` para cargar en el registro r_1 el contenido del arreglo apuntado por el registro r_2 en el índice r_3 , es decir, $r_1 := r_2[r_3]$.
- Calcular el grafo de flujo de control.
- Aproximar las definiciones de alcance en todos los puntos del programa, usando análisis de flujo de datos.

Justificar todas las respuestas.