Parseo y Generación de Código – 2do semestre 2024
Licenciatura en Informática con Orientación en Desarrollo de Software
Universidad Nacional de Quilmes

Segundo parcial

Nota: este parcial es a libro abierto. 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. En este ejercicio supondremos que un **polinomio** está representado con una lista de coeficientes, de tal modo que el n-ésimo coeficiente de la lista acompaña al término de grado n. Por ejemplo, la lista `[3, 2, -5, 0, 9, 7]` representa el polinomio $7X^5 + 9X^4 - 5X^2 + 2X + 3$. Escribir una función:

```
compilarPolinomio :: [Int] -> [Op]
```

que recibe una lista de coeficientes que representan un polinomio y produce código de tres direcciones para calcular el valor del polinomio *usando sólo dos registros* y almacenando el resultado en el registro r_0. Las operaciones son las siguientes:

```
type Reg = Int              -- los registros se identifican con un número entero (r0, r1, r2, ...)

data Op = Mov Reg Int       -- ri := c       : almacena una constante numérica en ri
        | Add Reg Reg Reg   -- ri := rj + rk : suma dos registros rj y rk, almacenando el resultado en ri
        | MulX Reg Reg      -- ri := rj * X  : multiplica rj por X y almacena el resultado en ri
```

Importante: se deben usar sólo dos registros (digamos `r0` y `r1`).

Ejercicio 2. Una variable se considera **insegura** en un punto del programa si su valor puede depender de datos provistos por el usuario. Los datos ingresados por el usuario se leen con la función `read()`. Consideraremos una variante del análisis de flujo de datos para aproximar el **conjunto de variables inseguras**. Notar que el análisis es de tipo *may–forward*. En una asignación de la forma "$x :=$ `read()`" la variable x pasa a ser insegura. En una asignación de la forma "$x := y \otimes z$" la variable x pasa a ser insegura si y es insegura o si z es insegura; en caso contrario x deja de ser insegura. Calcular el conjunto de variables inseguras en cada punto del siguiente programa:

```
1  x := read()    4  label_A:      7  z := read()     10  z := x * x
2  y := 2         5  y := y + z    8  jump label_A    11  jump label_B
3  z := y         6  label_B:      9  x := 3
```

Ejercicio 3. Considerar el cálculo-λ extendido con una operación $\mu x.t$ (conocida como operador de punto fijo). Las reglas de tipado se extienden del siguiente modo:

$$\frac{\Gamma, x : A \vdash t : A}{\Gamma \vdash \mu x.t : A}\ \text{T-FIX}$$

Suponiendo que hay un tipo `Int` y una constante `F : Int -> Int -> Int`.

a. Proponer un tipo para completar el agujero **?1** y dar una derivación para el juicio: $\vdash \mu x.\lambda y.\, \mathtt{F}\, y\, (x\, y) : \boxed{\textbf{?1}}$.

b. Proponer un tipo para completar el agujero **?2** y dar una derivación para el juicio: $\vdash \lambda y.(\mu x.y\, (\mathtt{F}\, x)) : \boxed{\textbf{?2}}$.

Justificar todas las respuestas.