

Ejercicio Semanal 1: El lenguaje EAB (Syntaxis)

Lenguajes de Programación I

López Soto Ramses Antonio

28 de agosto de 2019

Introducción

Objetivo

Implementar la sintaxis del lenguaje de expresiones aritmético-booleanas.

$$\begin{aligned} e ::= & x \mid n \mid \text{true} \mid \text{false} \\ & \mid e + e \mid e - e \mid \text{succ } e \mid \text{pred } e \\ & \mid \text{not } e \mid \text{and } e \ e \mid \text{or } e \ e \\ & \mid \text{lt } e \ e \mid \text{gt } e \ e \mid \text{eq } e \ e \\ & \mid \text{if } e \text{ then } e \text{ else } e \\ & \mid \text{let } x = e \text{ in } e \text{ end} \end{aligned}$$

Desarrollo

La implementación de este ejercicio se realizó utilizando el lenguaje de programación *Haskell*

implementación

Para la implementación se hizo uso de lo siguiente:

```
data Expr = V Identifier | I Int | B Bool
          | Add Expr Expr | Mul Expr Expr | Succ Expr | Pred Expr
          | Not Expr | And Expr Expr | Or Expr Expr
          | Lt Expr Expr | Gt Expr Expr | EQQ Expr Expr
          | If Expr Expr Expr
          | Let Identifier Expr Expr deriving (Eq)

type Identifier = String
```

```
type Substitution = (Identifier, Expr)
```

Completamos la función Show:

```
instance Show Expr where
  show e = case e of
    (V x) -> "V[" ++ x ++ "]"
    (I n) -> "N[" ++ (show n) ++ "]"
    (B t) -> "B[" ++ (show t) ++ "]"
    ...
```

Lo anterior lo utilizamos para completar las siguientes funciones:

- *Función que devuelve las variables libres de una expresión*

```
frVars :: Expr -> [Identifier]
```

- *Función encargada de realizar una sustitución a una expresión en caso de ser posible*

```
subst :: Expr -> Substitution -> Expr
```

- *Función que se encarga de verificar si dos expresiones son alfa-equivalentes*

```
alphaEq :: Expr -> Expr -> Bool
```

Conclusiones

La implementación, la principio fue sencilla, pues los primeros casos de cada función no hubo complejidad alguna. Los problemas comenzaron a surgir al momento de tratar de implementar el caso de Let en cada una de las tres funciones. Fue el más complicado.

En conclusion, el objetivo de la práctica fue cubierto, pues el fin de la misma fue implementar la sintaxis de las expresiones aritmético-booleanas en *Haskell* está completada.