Ejercicio Semanal 2: Cálculo Lambda

Lenguajes de Programación I

López Soto Ramses Antonio

21 de agosto de 2019

Objetivo

Simular el comportamiento del *Cálculo Lambda* sin tipos.

Introducción

El llamado *Cálculo Lambda* es considerado el lenguaje de programación universal más pequeño, pues es conformado por una regla de transformación simple y un esquema para definir funciones. Es equivalente a las *Máquinas de Turing* porque es capaz de expresar y evaluar cualquier función computable.

El cálculo Lambda puro, λ^U se define con la siguiente sintaxis concreta:

$$e ::= x|\lambda x.e|ee$$

Una función se representa:

$$f(x) = e =_{def} \lambda x.e$$

Desarrollo

Las funciones que se muestran a continuación fueron implementadas en el lenguaje de programación *Haskell*.

```
data Expr = V Identifier
          | L Identifier Expr
          | App Expr Expr deriving(Eq)

type Identifier = String

type Substitution = (Identifier, Expr)
```

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

frVars :: Expr -> [Identifier]

- *Función que se encarga de incrementar variables en una unidad* -

incrVar :: Identifier -> Identifier

- *Función que se encarga de devolver una expresión equivalente* -

alphaExpr :: Expr -> Expr

- *Función que se encarga de realizar una sustitución dada* -

subst :: Expr -> Substitution -> Expr

- *Función que se encarga de implementar una beta reducción* -

beta :: Expr -> Expr

- *Función que determina si una expresión es normal* -

normal :: Expr -> Bool

- *Función que se encarga de evaluar una expresión lambda* -

eval :: Expr -> Expr

Conclusión

Algunos de los problemascon los cuales se tuvo que lidiar: fueron con la implementación de las funciones *normal y eval*, pues las ideas de cómo resolver algunos conflictos tomó tiempo.

En esta ejercicío práctico quedó más claro como es que se comporta y funciona el cálculo Lambda sin tipos.