Práctica 4: Inferencia de tipos

Lenguajes de Programación I

López Soto Ramses Antonio
Quintero Villeda Erik

27 de septiembre de 2019

Introducción

Objetivo

Integrar las funciones que nos ayudaran a inferir los tipos de las expresiones.

Desarrollo

La práctica se realizó con el lenguaje de programación *Haskell*.

Teniendo la práctica 3 resuelta, sólo se agregaron cosas a distintos modulos

Y extendimos el el lenguaje con nuevas fuciones y se agregaron algunas otras.

- 1
  - – *Función que devuelve el conjunto de variables de tipo.*
    vars :: Expr -> [Identifier]
  - - *Función que aplica la sustitución a un tipo dado.* -
    subst :: Identifier -> Identifier
  - – *Función que realiza la composición de dos sustituciones*
    comp :: Expr -> Substitution -> Expr
  - – *Función que elimina sustituciones redundantes (T 0 := T 0) en una sustitución*
    simpl :: Expr -> Expr
- 2

- *- Función que busca el tipo de una variable en un con- texto dado. Devuelve el tipo en caso de encontrarlo y Nothing en otro caso -*

  find :: Expr -> Expr

- *- Dado un conjunto de variables de tipo, obtiene una variable de tipo fresca, es decir, que no aparece en este conjunto -*

  fresh :: Expr -> Expr

- *Dada una expresión, infiere su tipo implementando las reglas descritas anteriormente. Devolviendo el contexto y el conjunto de restricciones donde es válido. Utiliza el conjunto de variables de tipo para crear variables de tipo frecas durante la ejecución*

  infer' :: Expr -> Expr

Conclusión

Algunos de los problemas a los cuales nos enfrentamos fue a la ejecución mediante make, pues nos regresaba errores que no entendíamos muy bien y, además, el módulo Semantic nos tomó más tiempo de lo necesario, sin mencionar que no pudimos terminar la funcion infer' debido a falta de tiempo y a una mala organizacion.