

README de la Practica 6

Escamilla Soto Cristopher Alejandro, 314309253

Montiel Manriquez Ricardo, 314332662

27 de Mayo del 2022

Definimos el lenguaje L10

Definimos los predicados para los terminales según el lenguaje una variable es cualquier símbolo que no sea una primitiva

Primitivas del lenguaje son: + - * / length car cdr.

Constantes pueden ser: Integers Chars Booleans

Tipos que nuestro lenguaje usara son:

Bool | Int | Char | List | Lambda | ($\langle \text{type} \rangle \rightarrow \langle \text{type} \rangle$) | (List of $\langle \text{type} \rangle$)

Asi mismo definimos el parser de nuestro lenguaje L10

Ejercicio 1:

La función uncurry es la encargada de descurricular las expresiones lambda de nuestro lenguaje.

Primero se definió un nuevo lenguaje L11 en el cual el constructor lambda vuelve a ser multiparametrico.

Funciones Auxiliares Ejercicio 1:

body-lambda :: Función que nos regresa el cuerpo de una expresión lambda

lst-lambda :: Función que obtiene la lista de asignaciones que se tienen en las expresiones lambdas.

Ejercicio 2:

La función symbol-table-var que genera la tabla de símbolos de una expresión del lenguaje. Para modelar la tabla de símbolos usaremos una estructura Hash Table del lenguaje Racket.

aux :: Función encargada de generar la tabla de símbolos según el caso de la expresión.

Ejercicio 3:

Definimos un nuevo lenguaje L12 en el que los constructores de asignación solo reciben una variable.

Definimos el parser de L12.

Creamos la función `assignment` que modifica los constructores `let`, `letrec` y `letfun`, eliminando el valor asociado a los identificadores y el tipo correspondiente.