

README de la Practica 4

Escamilla Soto Cristopher Alejandro, 314309253

Montiel Manriquez Ricardo, 314332662

30 de Abril del 2022

Para el ejercicio 1, 2 y 3 no creemos que sea necesaria una explicación detallada, dado que en clase de laboratorio se nos dio una explicación de lo que había que hacer en estos ejercicios.

En el ejercicio 1 currificamos las expresiones `let` y `letrec`.

En el ejercicio 2 reemplazamos `let` por `letrec`.

En el ejercicio 3 definimos un nuevo lenguaje con el nombre `L8` que contara con el constructor `letfun` y este recibe una lambda, siendo esta el valor y cuerpo de `letfun`, esta parte la definimos en `un-anonymous`.

Ejercicio 4

Este proceso funciona como verificador de la sintaxis de las expresiones y consiste en verificar que el número de parámetros corresponde con la aridad de las primitivas.

Si corresponde, se regresa la misma expresión en caso contrario se lanza un error, especificando que la expresión está mal construida.

Para este ejercicio usamos una función auxiliar:

Predicado auxiliar `aridad?` : regresa si la aridad del operador es correcta concorde al operador.

En esta función contemplamos operadores que no están en la especificación sin embargo creemos que hacen mas completo al lenguaje.

La función hace uso del lenguaje 8 (`L8`) y devuelve el resultado en este mismo lenguaje o un mensaje de error si la aridad es incorrecta.

Ejercicio 5

Este proceso funciona como verificador de la sintaxis de las expresiones y consiste en verificar que la expresión no tenga variables libres, de existir variables libres se regresa un error en caso contrario

la salida es la misma expresión.

La función trabaja sobre un lenguaje L8 y lo devuelve en este mismo lenguaje.

Nos apoyamos en la función `memq` para buscar una variable dentro de un ambiente que esta representado por una lista, de esta forma sabremos si es libre o no.

Nota:

En las lineas 183 y 209 que corresponden al ejemplo de los ejercicios 4 y 5, las lineas están comentadas, la razón fue que como generar errores al momento de hacer la ejecución del programa este finaliza y no deja ver el resto, simplemente para probar el ejemplo descomentamos la linea que queramos ver en ejecución.