Práctica Final: Traductor de C a Lisp

Procesadores del lenguaje

Mayo 2023

Grado en Ingeniería Informática

Pablo Hidalgo Delgado. NIA: 100451225
César López Navarro. NIA: 100451326

5 y 6.

Diseñamos la gramática de manera que al comienzo del código se puedan definir tantas sentencias globales como se desee. Sin embargo, una vez definida alguna función, ya sea main() u otra cualquiera, se deben de seguir definiendo funciones, no se podrá volver a definir una sentencia global. Además, también controlamos que la función main() exista y sea la última función que se define.
En nuestro conjunto de pruebas tenemos algunos archivos que simulan estos comportamientos: *no_main.c* (no existe la función main()), *solo_main.c* (solo existe la función main), *no_main_final.c* (la función main() no es la última función) y *sentencias_globales_en_medio.c* (existen sentencias globales en medio de definición de funciones).

7.

Esto se utiliza en todas las pruebas por lo que no hemos realizado pruebas específicas para este punto.

8.

Esto se utiliza en numerosas pruebas por lo que no hemos realizado pruebas específicas para este punto.

10.

Cabe destacar que, para que se pueda imprimir la cadena entre comillas, en la semántica utilizamos el carácter '\' para indicar que queremos que las comillas sean interpretadas como un carácter literal.
Esto se utiliza en numerosas pruebas por lo que no hemos realizado pruebas específicas para este punto.

11.

Esto se utiliza en numerosas pruebas por lo que no hemos realizado pruebas específicas para este punto.

12.
Los incluimos dentro de la producción expresión ya que en C los operadores lógicos dan como resultado un entero 1 (representa True) o 2 (representa False). Por tanto, se podrían tratar como expresiones. Es responsabilidad del programador utilizarlos adecuadamente.
Esto se utiliza en numerosas pruebas por lo que no hemos realizado pruebas específicas para este punto.

13. WHILE
Lo incluimos en la producción sentencia_local.
Lo probamos en nuestro conjunto de pruebas en el archivo *while.c*

14. IF
if.c en este archivo probamos todos los casos: if dentro de otro if, if con una sentencia, if con varias sentencias, if con una sentencia y else con varias sentencias, if else con 1 sentencia…

15. FOR
También lo incluimos en sentencia local. Controlamos que la inicialización de la variable de control se haya realizado antes y que el incremento se haga al final por medio de la semántica.

16. Vectores

No controlamos que las variables hayan sido previamente declaradas ya que no se nos pide.