

Autómatas, Teoría de Lenguajes y Compiladores

TPE: Sprache

Integrantes:

* Luca La Mattina (57093)
* Rodrigo Fera (58079)

Segundo Cuatrimestre 2020

Índice

Idea y Consideraciones.....……………………………………………….3

Desarrollo………………………………………………………………………..3

Gramática………………………………………………………………………..3

Dificultades………………………………………………………………………4

Extensiones……………………………………………………………………..4

Referencias………………………………………………………………………4

Idea y Consideraciones

El alemán es una persona muy orgullosa de su idioma. Si se tiene la posibilidad de visitar Alemania (o algún otro país de habla alemana como Austria, parte de Suiza o Liechtenstein) uno puede notar que en el cine las películas están en alemán, los videojuegos son jugados con el texto y voces en alemán y si una página web o aplicación no tiene una versión de la misma en alemán muchos ciudadanos evitaran usarla ya que les genera desconfianza. Dado que ambos integrantes de este grupo estudiamos alemán queríamos crear un lenguaje en este idioma para aplicar nuestro conocimiento y ofrecer una introducción al mundo de la programación más amigable y familiar a las deutschsprachige Menschen (gente de habla alemana). Nuestra elección de lenguaje fue C dado que fue el primer lenguaje de programación visto por ambos integrantes del grupo por lo que creemos que es una buena elección para alguien que quiere aprender a programar. Decidimos reemplazar la mayor cantidad de símbolos por su palabra en alemán para maximizar la localización del lenguaje.

Desarrollo

Utilizamos las herramientas vistas en clase Yacc y Lex. Para el Lex creamos el archivo Übersezter.l que contiene todas las palabras del lenguaje. Luego se creó la estructura que implementa el árbol de sintaxis, para esto usamos una librería detallada en la sección de referencias que adaptamos a nuestras necesidades. Finalmente se creó la gramática en un archivo Yacc llamado Grammatik.y el cual usa el árbol mencionado anteriormente.

Gramática

La gramática creada incluye lo siguiente:

Tipos de datos: int y string, booleans

Operadores aritméticos: +, -, \*, /, %

Operadores relacionales: <, >, <=, >=, ==, !=

Bloques: if, while

Operadores lógicos: &&, ||

Para ver su traducción se puede consultar el archivo Übersetzer.l

Dificultades

La parte más difícil del trabajo fue la primera, no estábamos seguros de cómo empezar a codear ya que todo esto era nuevo para nosotros y no teníamos experiencia con Lex y Yacc. Otra dificultad fue que no estábamos seguros como hacer debugging, en un momento recibíamos sintax error y tuvimos que recurrir al método de descarte para saber que parte de nuestro código estaba fallando.

Extensiones

Como extensiones se podrían agregar muchísimas funcionalidades de C no soportadas soporte de estructuras con malloc, free, etc esto sin embargo seria complejo ya que hay que interactuar con el administrador de memoria. También se podría implementar un garbage collector aunque C no tenga uno. Finalmente se podría implementar una forma que facilite el debugging que es una parte fundamental de cualquier proyecto de programación.

Referencias

Librería de árboles utilizada ya incluida en el tp la cual fue modificada para acatar nuestras necesidades: posix ninja

Se puede encontrar en el siguiente link: <https://github.com/Chronic-Dev/libcnary>