

Informe del Compilador C-TDS

Materia: Taller de diseño de software.

Integrantes: Agustin Bauer, Alan Gonzalez, Luciano Putruele.

GLOSARIO:

- Introducción	3
- Sobre Implementación general	4
- Etapas:	
- Analizador Sintáctico y Léxico de c-tds (Scanner y Parser)	4
- Analizador Semántico de c-tds	4

Introducción:

Diseño e implementación de un compilador para el lenguaje c-tds.

En este informe, se va explicar las consideraciones tenidas en cuenta de cada etapa del compilador, además, de las decisiones de diseño elegidas para su construcción.

Por otro lado, se explicara de cada etapa las estructuras de datos diseñadas, para ser utilizadas por cada etapa.

Se suma a este informe, aclaraciones y ejemplificación de algunos algunos aspectos difíciles de entender en el código del compilador.

La organización del siguiente informe va a estar dividida por las etapas del compilador, las cuales cada una contiene una breve introducción a la etapa, las estructuras de datos utilizadas y las decisiones de diseño en la misma. Y por último, las consideraciones, aclaraciones y ejemplos.

Sobre Implementación General:

La implementación del compilador c-tds fue desarrollada usando diferentes herramientas, como CUP, Jflex, el lenguaje de programación JAVA, sistema operativo linux de 64bits, editores de texto, etc.

Etapas :

Analizador Sintáctico y Léxico de c-tds (Scanner y Parser)

Para desarrollar dicho lenguaje, durante la primera etapa se implementó en el lenguaje Jflex los token del lenguaje teniendo en cuenta las siguientes consideraciones:

- 1) Las palabras reservadas del lenguaje son: boolean, break, class, continue, else, false, float, for, if, int, return, true, void, while.
- 2) Los valores enteros son de la forma digito+, donde digito es un valor de [0,9].
- 3) Los valores reales son de la forma digito+ . digito+, donde digito es un valor de [0,9].
- 4) Los identificadores son de la forma alpha alphaNumeric, donde alpha es un valor de [a..zA..Z] y alphaNumeric es un valor de [a..zA..Z0..9_].
- 5) Los tipos de comentario que acepta el lenguaje son el comentario en linea que es de la forma "//comentario " y el comentario multilinea que es de la forma "/* comentario [/n] */".

Luego de haber definido todos los token del lenguaje, se definió la gramática (Análisis Sintáctico del lenguaje) la cual se desarrollo en CUP, estratificando las diferentes reglas para evitar conflictos reduce/reduce & shift/reduce

Analizador Semántico de c-tds:

La verificación de las reglas semánticas del lenguaje c-tds se realizó utilizando una tabla de símbolos para mantener la información de los identificadores de un programa. Dicha tabla fue implementada como una lista de listas (HashMap). Donde cada elemento del HashMap era del tipo Descriptor (definidos por nosotros).

Para el análisis semántico a cada nodo del árbol de parsing se le asoció un tipo Descriptor, el cual almacenaba la información necesaria para determinar las reglas semánticas en cada punto del programa.

Además, para realizar el analizador semántico se consideró que el lenguaje c-tds es de tipos de datos estáticos, es decir que tiene alcance estático y fuertemente tipado.

Sobre la implementación:

Para la etapa de verificación semántica se desarrollaron las siguientes estructuras de datos en Java:

* Tabla de símbolos: implementada como una lista (LinkedList) de ambientes (HashMap), donde cada ambiente está conformado por descriptores (clase abstracta) que pueden ser de tipo simple, función, o arreglo (concretas) que definen sus características.

