

Práctica III: Análisis semántica y generación de código intermedio

Francisco Javier Roig
Jose Miguel Avellana

El trabajo siguiente tiene como objetivo integrar las herramientas de soporte al diseño de traductores presentadas durante el curso, entre estas se dan uso de yacc, bison y c.

Sobre las diferentes etapas cubiertas del trabajo:

1. Diseño del lenguaje fuente.
 - a. Las variables pueden ser locales o globales, en caso de ser globales se denota tal que así:
global int a = 1;
 - b. Se cubren los tipos básicos: carácter, entero, real y booleano.
 - c. El compilador acepta operadores aritméticos y lógicos.
 - d. Estructura algorítmica if/else/fi.
 - e. Es posible crear métodos dentro del programa con begin/end, pero no puede ser asignado.
 - f. El hilo principal del programa se denota por un begin/end general.
2. Diseño e implementación del analizador léxico. Se definen las palabras clave del lenguaje: tipos, visibilidad y estructuras algorítmicas.
3. Diseño e implementación del analizador sintáctico.
4. Diseño e implementación del analizador semántico para la visibilidad de las variables. Cuando detecta una variable global la añade , además, en la table de variables globales.
5. No se ha llevado a cabo ningún tipo de implementación sobre la generación de código intermedio.