

Funcionalidad de las fases del compilador

Que para coadyuvar en la calificación del primer parcial

Alumno

Instituto Politécnico Nacional

Agosto de 2018

Trabajo de descripción de cada una de las fases del compilador

Objetivo

Identificar y describir cada una de las fases del compilador.

Enumeración de las fases del compilador.

Las fases del compilador, son:

- 1 Análisis léxico.
- 2 Análisis sintáctico.
- 3 Análisis semántico.
- 4 Generación de código.
- 5 Optimización de código.

La fase de análisis léxico

Definición de análisis léxico

Generalmente en computación lo que el compilador debe analizar es un programa del cual debe reconocer las palabras que contiene.

Diagrama del análisis léxico

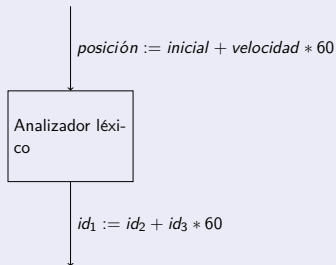


Figura: La entrada y la salida de la primera fase del compilador.

La entrada y la salida del analizador léxico

La entrada al analizador léxico

$$posición := inicial + velocidad * 60$$

La salida del analizador léxico

$$id_1 := id_2 + id_3 * 60$$

La fase de análisis sintáctico

Definición de análisis sintáctico

En la fase de análisis sintáctico, un compilador verifica si o no los tokens generados por el analizador léxico son agrupados de acuerdo a reglas sintácticas del lenguaje. Si los tokens en una cadena son agrupados de acuerdo a reglas de sintaxis del lenguaje, entonces las cadenas de tokens generados por el analizador léxico son aceptados como construcción válida del lenguaje.

Diagrama de la fase de análisis sintáctico

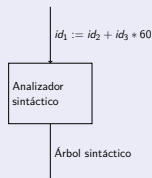


Figura: La entrada y la salida de la segunda fase del compilador.

Entrada y salida de la fase de análisis sintáctico

Entrada del analizador sintáctico

La entrada es:

$$id_1 := id_2 + id_3 * 60$$

La salida del analizador sintáctico

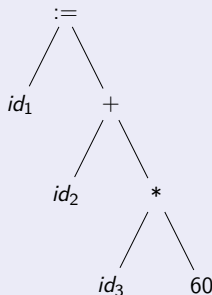


Figura: Árbol sintáctico, salida del analizador sintáctico.

La fase de análisis semántico

Definición de análisis semántico

Un programa que es gramaticalmente correcto puede contener serios errores que evitarían su compilación. Para detectar tales errores, un compilador realiza un mayor nivel de verificación que involucra considerar cada sentencia en su contexto. Estas verificaciones encuentran errores de tipo y concordancia.

Diagrama del analizador semántico

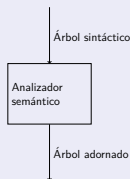


Figura: La entrada y la salida de la tercera fase del compilador.

Entrada y salida de la fase de análisis semántico

Entrada del analizador semántico

La entrada es:

$$id_1 := id_2 + id_3 * 60$$

La salida del analizador semántico

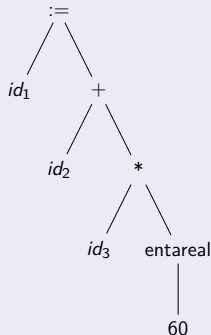


Figura: Árbol adornado, salida del analizador semántico.

La fase de generación de código

Definición de generador de código

Es el que transforma.

Diagrama del generador de código

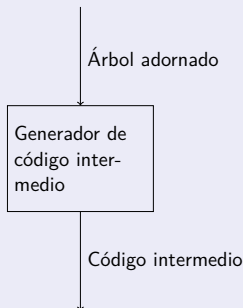


Figura: La entrada y la salida de la fase de generación de código intermedio.

La entrada y la salida del generador de código

Entrada al generador de código

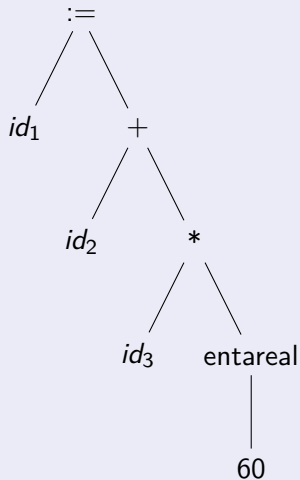


Figura: Árbol adornado, salida del analizador semántico.

La fase de optimización de código

Definición de optimización de código

Un sistema de tipos bien diseñado provee al compilador con una información detallada de cada expresión en el programa, la información que puede ofrecer se usa para producir más traducciones eficientes.

Diagrama de la optimización de código

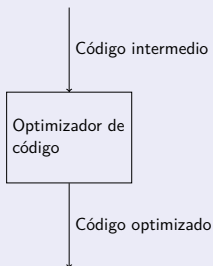


Figura: La entrada y la salida de la fase de optimización de código.