

Escuela Politécnica Nacional

Nombre: Tomando Elías Guila Villegas

La fase del compilador que traduce el árbol de sintaxis del programa en una representación intermedia que sirve como puente entre el análisis y la generación de código máquina. Esta etapa es clave para permitir optimizaciones y facilitar la portabilidad del compilador.

Se representan diferentes formas de representar el programa en un nivel más bajo que el código fuente pero aún independiente de la máquina:

- **Árbol de sintaxis interna (AST)**: representación jerárquica que omite detalles gramaticales, útil para evaluaciones estructurales.
- **Código de tres direcciones (TAC, Three Address)**: Serie de instrucciones simples de tipo $x = y \text{ op } z$ adecuadas para representar operaciones básicas.
- **Código tipo pila**: usado en máquinas abstractas basadas en pila, más compacto pero menos eficiente para optimizaciones.

El código de tres direcciones se compone de instrucciones que operan con a lo sumo tres operaciones / operandos usando temporales para almacenar resultados intermedios.

Se utilizan árboles o DAGS (grafos acíclicos dirigidos) para identificar subexpresiones comunes.

lexema: instancia concreta del programa fuente que coincide con un patrón.

El analizador léxico también puede asociar atributos a los tokens para que los fases posteriores puedan operar con mayor precisión.

Uso de búferes de entrada.

Para mejorar la eficiencia de la lectura, se utiliza un esquema de doble búfer. Cada búfer tiene un tamaño fijo (como un bloque de disco) y se llena alternadamente. Esto permite leer bloques completos de caracteres en lugar de uno por uno, reduciendo el número de llamadas al sistema. Además, se emplean centinelas (caracteres especiales como `\0`) al final de los búferes para evitar verificaciones constantes de fin de bloque.

Errores léxicos y recuperación:

El analizador debe manejar casos donde no se puede identificar un token válido. Estrategias comunes incluyen:

- Modo de pánico: descartar caracteres hasta encontrar un token válido.
- Corrección Simple: insertar, eliminar o sustituir caracteres de forma de una coincidencia válida.
- Transformaciones más complejas: aunque posible, son costosas y poco prácticas.

Jonathan Pulgar