

Generador de código

Toma una representación intermedia del código (así como la información más importante de la tabla de símbolos) y lo mapea al lenguaje destino esto lo suele hacer partiéndolo en bloques básicos (conjuntos de instrucciones que se ejecutan juntas). Si el lenguaje destino es código máquina, los registros o locaciones de memoria son seleccionados para cada variable usada en el programa. La asignación de almacenamiento se realiza durante la optimización de código intermedio o en la generación del código.

Algunas de las técnicas usadas en la generación de código se pueden aplicar sin la necesidad de que exista una previa optimización de código.

El código final debe de ser correcto y de alta calidad por ende debe de usar perfectamente los recursos de la máquina. Es por esto por lo que esta etapa es difícil así que debemos de estar contentos con que el código sea bueno mas no óptimo.

El hecho de que no se consiga una optimización total es debido a problemas como:

- Manejo de memoria
- Selección de instrucciones
- Asignación de registros
- Evaluación del orden

El generador de código tiene tres principales tareas:

- Selección de instrucciones: elegir las instrucciones apropiadas del lenguaje destino para representar las sentencias de la representación intermedia (depende del lenguaje destino y su variedad de instrucciones y por ende de un análisis sobre que conjunto de instrucciones es mejor)
- Asignación de registros: involucra decidir en que registros almacenar que variables. Instrucciones con registros suelen ser mas rapidas y cortas que usan operadores en memoria. Complicado debido al hardware, SO y las convenciones de uso de registros
 - Selección de registro. Elegimos el registro
 - Asignación de registro. Tomamos el registro
- Ordenamiento de instrucciones: Decidir en que orden programar la ejecución de las instrucciones. El tratar de obtener el ordenamiento más optimo también es un problema por lo que se debería optar por el orden en que se encuentra el código intermedio