

1.6 Generación de la tabla de símbolo y de direcciones

Las tablas de símbolos (también llamadas tablas de identificadores y tablas de nombres), realizan dos importantes funciones en el proceso de traducción: verificar que la semántica sea correcta y ayudar en la generación apropiada de código. Ambas funciones se realizan insertando o recuperando desde la tabla de símbolos los atributos de las variables usadas en el programa fuente. Estos atributos, tales como: el nombre, tipo, dirección de almacenamiento y dimensión de una variable, usualmente se encuentran explícitamente en las declaraciones o más implícitamente a través del contexto en que aparecen los nombres de variables en el programa.

Una de las estructuras de datos que se encuentran relacionadas con las fases del proceso de compilación es la tabla de símbolos, la cual tiene como propósito registrar información que se comparte entre varias etapas y que permite administrar los recursos asociados a las entidades que manipulará el programa. La tabla de símbolos tiene típicamente la siguiente estructura:

Una tabla de símbolos puede conceptualizarse como una serie de renglones, cada uno de los cuales contiene una lista de valores de atributos que son asociados con una variable en particular. Las clases de los atributos que aparecen en una tabla de símbolos dependen en algún grado de la naturaleza del lenguaje de programación para el cual se escribe el compilador.

Por ejemplo, un lenguaje puede ser sin tipos, y por lo tanto el atributo tipo no necesita aparecer en la tabla. Similarmente, la organización de la tabla de símbolos variará dependiendo de las limitaciones de memoria y tiempo de acceso.

Contenido de la tabla de símbolos.

Esencialmente la información que aparece en la tabla de símbolos es de dos tipos:

- x El propio símbolo, y
- x Los atributos necesarios para definir el símbolo a nivel semántico y de generación de código.

Los atributos requeridos para cada símbolo dependen a nivel general si:

- x del tipo de gestión de memoria,
- x el lenguaje está, o no, estructurado en bloques,
- x el símbolo es, o no, parámetros de un procedimiento o función.

CONSTRUCCIÓN DE LA TABLA DE SÍMBOLOS.

1. El analizador de léxico deberá:

- x Insertar los símbolo detectados en la tabla de símbolos,
- x Crear la tabla de símbolos parcialmente,
- x Señalar la línea del programa fuente en donde aparecen.

2. El analizador semántico:

- x Añadir los tipos, si procede, a los símbolos que aparecen en la tabla de símbolos.

Cuando se hace una implantación de una única pasada, entonces los símbolos son insertados y calificados, a nivel semántica y de generación de código, por el analizador sintáctico y en el mismo instante que son

detectados por el analizador léxico y que son pasados al analizador sintáctico.

Operaciones sobre la tabla de símbolos.

x INSERTAR

x CONSULTAR

x MODIFICAR (añadir atributos nuevos)

El CUANDO y el CÓMO se usan estas operaciones dependen del tipo de lenguaje:

Lenguajes con DECLARACIONES DE VARIABLES:

Explícitas:

- ☐ Declaraciones: sólo INSERTAR.
- ☐ Referencia: sólo CONSULTAR.

x Implícitas:

- ☐ CONSULTAR si no está ya incluida.
- ☐ INSERTAR, en caso contrario.
- ☐ Lenguajes con estructura de BLOQUE : CREAR SUBTABLAS.

IMPLEMENTACIÓN DE LA TABLA DE SÍMBOLOS

La distribución de la información de la tabla de símbolos dependerá de las características del lenguaje y de las restricciones establecidas para los símbolos.

Campo dedicado para el símbolo

x Formato fijo

Apropiado cuando se establece límite en el número de caracteres que forman los símbolos y, además, sea pequeño.

En este caso sólo se dispone de un área fija en la tabla para almacenar el símbolo.

x Formato variable

Se dispone de la tabla de símbolos y de un área auxiliar en donde se introducen los símbolos de modo consecutivo. En la TS se sustituye el campo dedicado para el nombre del símbolo por un puntero al área auxiliar y un entero que indica la longitud del mismo.