Índice

Marina de la Cruz
Alfonso Ortega

- Definición de la información de los elementos de la tabla de símbolos
  - Motivación
  - Selección de la información para el lenguaje ALFA
- Diseño de la tabla de símbolos

Definición de la información de los elementos de la tabla de símbolos

Motivación

- En la tabla de símbolos se almacena información de
  - las **variables (globales del programa principal y locales de las funciones)**
  - los **parámetros de las funciones**
  - las **funciones**

- Hay que definir la información contenida en los elementos de la tabla de símbolos. Las decisiones tomadas en esta fase de desarrollo **dependen de las características del lenguaje fuente** y de las **prestaciones** que debe ofrecer el compilador.

- Habitualmente, un lenguaje complejo/extenso requiere almacenar más información en la tabla de símbolos que un lenguaje sencillo/reducido.

- Lógicamente, un compilador de altas prestaciones, exige más cantidad de información en la tabla de símbolos que un compilador con pocas prestaciones.

Motivación

- Ejemplos:
  - Si el lenguaje dispone de vectores, para controlar que las operaciones de indexación se realizan dentro del rango correcto, es necesario almacenar en la tabla el tamaño de cada vector definido.
  - Si el compilador ofrece la prestación de controlar que cada variable sea **inicializada** antes de usarla, es necesario almacenar en la tabla de símbolos información acerca de la inicialización de las variables. De esta manera, cada vez que se usa una variable, se consulta la tabla de símbolos para comprobar si ha sido inicializada.
  - Se puede comprobar que el número de parámetros en la llamada a una función coincide con el número de parámetros que aparecen en la definición de la función, almacenando en la tabla de símbolos, para cada función, el número de parámetros con el que se define.
  - Para comprobar que la asignación de un valor a una variable es correcta desde el punto de vista de la compatibilidad de tipos, es necesario almacenar en la tabla de símbolos el tipo de cada variable.

Selección de la información para el lenguaje ALFA

- **Clave de acceso a la tabla:**
  - **Identificador (lexema)** de la variable, parámetro o función.

- **Categoría del elemento:**
  - Se distinguen tres categorías de elementos almacenados en la tabla de símbolos: **variable**, **parámetro de función** y **función**.
  - Define la categoría del identificador que está almacenado en la tabla de símbolos.
  - Se recomienda el uso de macros del tipo:
    - `#define VARIABLE 1`
    - `#define PARAMETRO 2`
    - `#define FUNCION 3`

- **Tipo básico de dato:**
  - Define el tipo básico del identificador.
  - ALFA dispone de dos tipos básicos: **`boolean`** e **`int`.**
  - Se recomienda el uso de macros del tipo:
    - `#define BOOLEAN 1`
    - `#define INT 2`
  - Si el identificador corresponde a una variable o a un parámetro de función, esta información se refiere al tipo básico de la variable o parámetro.

Selección de la información para el lenguaje ALFA

- Si el identificador corresponde a un vector, esta información se refiere al tipo de dato de los elementos del vector.
- Si el identificador corresponde a una función, esta información se refiere al tipo del valor de retorno de la función.

- **Categoría:**
  - Identifica la estructura de la información asociada al identificador de una variable o un parámetro (aunque en principio los parámetros sólo pueden ser de clase escalar).
  - Se pueden distinguir dos clases: **escalar** y **vector**.
  - Se recomienda el uso de macros del tipo:
    - `#define ESCALAR 1`
    - `#define VECTOR 2`

- **Tamaño:**
  - Sólo para identificadores de vectores.
  - Representa el número de filas.
  - Su valor está comprendido entre 1 y 64.

Selección de la información para el lenguaje ALFA

- **Número de parámetros:**
  - Sólo para elementos de tipo función.
  - Representa el número de parámetros formales de la función.
- **Posición del parámetro:**
  - Sólo para elementos del tipo parámetro de función.
  - Representa la posición del mismo dentro de la lista de parámetros (comenzando en 0).
- **Número de variables locales:**
  - Sólo para elementos de tipo función.
  - Representa el número de variables locales de la función.
- **Posición de la variable local:**
  - Sólo para elementos del tipo variable local de función.
  - Representa la posición de la misma dentro de la sección de declaraciones de la función (comenzando en 1).

Diseño de la tabla de símbolos

- La tabla de símbolos de un lenguaje en el que únicamente hay un ámbito para todas las variables del programa, puede consistir en una única estructura que almacene información de todas las variables.

- La tabla de símbolos de un lenguaje en el que se contemplan ámbitos anidados tiene que implementar de alguna manera la gestión de dichos ámbitos para gestionar correctamente la visibilidad de las variables/parámetros/funciones.

- En un lenguaje como ALFA se cumplen las siguientes condiciones:
  - En un módulo de programa existen variables "globales" (situadas al comienzo del módulo). Estas variables se visualizan en las sentencias "globales" del módulo y dentro de todas las funciones.
  - Las funciones se sitúan después de las variables globales y sin anidamiento (dentro de una función no se puede declarar otra función) Dentro de una función se visualizan las variables globales, las variables locales y los parámetros de la función y las funciones declaradas previamente.
  - Las sentencias del módulo se sitúan después de las funciones. En estas sentencias se visualizan las variables globales y las funciones.

Diseño de la tabla de símbolos

- Para implementar la tabla de símbolos de un lenguaje como ALFA con las reglas de visibilidad descritas anteriormente es suficiente:
  - Utilizar dos estructuras de almacenamiento de los identificadores. Una estructura para las variables globales y las funciones, que llamaremos ***tabla global*** y otra estructura que almacena las variables locales y los parámetros de las funciones que llamaremos ***tabla local.*** La tabla de símbolos está formada por esas dos tablas con unas reglas de inicialización de las tablas y de inserción y búsqueda de identificadores en ambas tablas.
  - Inicialmente, se inicializa la tabla global y se insertan en ella todos los identificadores de las variables globales. Dado que no puede haber identificadores duplicados, la inserción falla si ya existe el identificador en la tabla global.
  - Cuando aparece una declaración de función, se inserta su identificador en la tabla global. Además, se inicializa la tabla local y se insertan en ella, el propio identificador de la función, los identificadores de las variables locales de la función y los parámetros de la misma. Dado que no puede haber identificadores duplicados, la inserción falla si ya existe el identificador en la tabla local.
  - En la parte de sentencias de la función, la aparición de un identificador desencadena la búsqueda del mismo, primero en la tabla local y después en la global (este orden en la búsqueda implementa el ocultamiento de una variable global por una local del mismo nombre).