

**Investigación Tabla de símbolos**

**Nombre: Alexander Morgan**

**Cedula: 8-859-2236**

**Materia: Compiladores**

**Profesor: Leonardo Ezqueda**

**Problemas de alcance**

La fase de análisis semántico obtiene su nombre por requerir información relativa al significado del lenguaje, que está fuera del alcance de la representatividad de las gramáticas libres de contexto y los principales algoritmos existentes de análisis; es por ello por lo que se dice que captura la parte de la fase de análisis considerada fuera del ámbito de la sintaxis.



**Relación entre estructuras entre tiempo de ejecución y la tabla de símbolos**

La tabla de símbolos puede iniciarse con cierta información útil, tal como:

- Constantes: PI, E, etc.

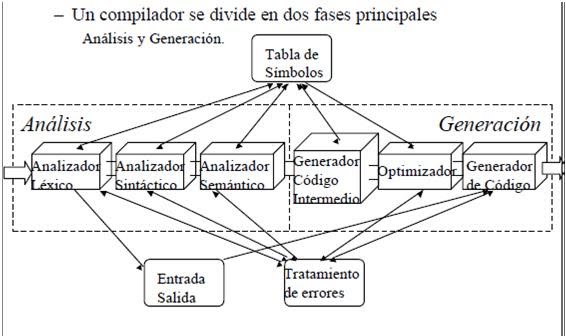
- Funciones de librería: EXP, LOG, etc.

- Palabras reservadas. Esto facilita el trabajo al lexicográfico, que tras reconocer un identificador lo busca en la tabla de símbolos, y si es palabra reservada devuelve un token asociado. Bien estructurado puede ser una alternativa más eficiente al lex tal y como lo hemos visto (hash perfecto).

Conforme van apareciendo nuevas declaraciones de identificadores, el analizador léxico, o el analizador sintáctico según la estrategia que sigamos, insertará nuevas entradas en la tabla de símbolos, evitando siempre la existencia de entradas repetidas.

El analizador semántico efectúa las comprobaciones sensibles al contexto gracias a la tabla de símbolos, y el generador de código intermedio usa las direcciones de memoria, asociadas a cada identificador en la tabla de símbolos, al igual que el generador de código.

El optimizador de código no necesita hacer uso de ella



**Las etapas de la construcción en la tabla de símbolos**

Declaración antes de uso, es una regla común utilizada en C y Pascal, que requiere que se declare un nombre en el texto de un programa antes de referenciarlo.

Permite construir la tabla de símbolos a medida que continúa el análisis sintáctico y que las búsquedas se realicen tan pronto como se encuentra una referencia de nombre en el código. Si falla la búsqueda es que no se ha declarado la variable .Facilita la compilación de una sola pasada.

Modula-2 no requiere de la declaración antes del uso, paso por separado para la construcción de la tabla de símbolos. No es posible la compilación de una sola pasada.

Estructura de bloques, es una propiedad común de los lenguajes modernos, pero... ¿Qué es un bloque? Es cualquier construcción que pueda contener declaraciones. ¿Cuándo un lenguaje está estructurado en bloques? Si permite la anidación de bloques dentro de otros bloques.

Ámbito léxico o estático: la tabla de símbolos se construye de manera estática siguiendo el orden de escritura del programa, tal y como hemos visto hasta ahora. En los lenguajes de bloques, como Pascal o C, el ámbito es el procedimiento, la función o el registro.

Ámbito dinámico (LISP): cuando la tabla de símbolos se construye siguiendo el orden de ejecución del programa.

Esto implica que se realicen operaciones de inserción y eliminación a medida que los ámbitos se introducen y extraen en tiempo de ejecución.

La tabla debe ser parte del ambiente de ejecución.

Se compromete la legibilidad de los programas porque las referencias no locales no pueden resolverse sin simular la ejecución del programa.

Es incompatible con la verificación de tipos estático, ya que los tipos de los datos de las variables deben mantenerse por medio de la tabla de símbolos. (Ej. Si en el main i =double).