

Compiladores 24-2

Tabla de Símbolos

Lourdes del Carmen González Huesca

luglzhuesca@ciencias.unam.mx

Facultad de Ciencias, UNAM

14 febrero 2024



Tabla de símbolos

- Estructuras para almacenar información sobre construcciones del programa fuente.
- La información es almacenada *incrementalmente* por las fases del front-end o de análisis para ser utilizada por las fases del back-end o de síntesis para generar el código objeto.

Tabla de símbolos

- Estructuras para almacenar información sobre construcciones del programa fuente.
- La información es almacenada *incrementalmente* por las fases del front-end o de análisis para ser utilizada por las fases del back-end o de síntesis para generar el código objeto.
- Relaciona nombres léxicos y sus atributos, por ejemplo de un identificador almacena su lexema, su tipo, la posición de almacenamiento (dirección relativa) y alguna otra información relevante como el tipo de dato.

Tabla de símbolos

- Estructuras para almacenar información sobre construcciones del programa fuente.
- La información es almacenada *incrementalmente* por las fases del front-end o de análisis para ser utilizada por las fases del back-end o de síntesis para generar el código objeto.
- Relaciona nombres léxicos y sus atributos, por ejemplo de un identificador almacena su lexema, su tipo, la posición de almacenamiento (dirección relativa) y alguna otra información relevante como el tipo de dato.
- El lexer creará e inicializará la tabla ya sea con los lexemas o sólo con un apuntador al identificador.

Tabla de símbolos

- Estructuras para almacenar información sobre construcciones del programa fuente.
- La información es almacenada *incrementalmente* por las fases del front-end o de análisis para ser utilizada por las fases del back-end o de síntesis para generar el código objeto.
- Relaciona nombres léxicos y sus atributos, por ejemplo de un identificador almacena su lexema, su tipo, la posición de almacenamiento (dirección relativa) y alguna otra información relevante como el tipo de dato.
- El lexer creará e inicializará la tabla ya sea con los lexemas o sólo con un apuntador al identificador.
- Los tokens generados por el lexer son pares que contienen el nombre y un valor atributo; este último es justo un apuntador a la tabla de símbolos.

Tabla de símbolos

- Estructuras para almacenar información sobre construcciones del programa fuente.
- La información es almacenada *incrementalmente* por las fases del front-end o de análisis para ser utilizada por las fases del back-end o de síntesis para generar el código objeto.
- Relaciona nombres léxicos y sus atributos, por ejemplo de un identificador almacena su lexema, su tipo, la posición de almacenamiento (dirección relativa) y alguna otra información relevante como el tipo de dato.
- El lexer creará e inicializará la tabla ya sea con los lexemas o sólo con un apuntador al identificador.
- Los tokens generados por el lexer son pares que contienen el nombre y un valor atributo; este último es justo un apuntador a la tabla de símbolos.
- Las palabras reservadas, operadores e identificadores también se almacenan en la tabla.

Front-end y la Tabla de Símbolos

Las tablas de símbolos son estructuras de datos que se utilizan para almacenar información sobre construcciones del programa fuente, relaciona nombres léxicos y sus atributos, éstos son información relevante que es recolectada *incrementalmente* por las fases del front-end:

Front-end y la Tabla de Símbolos

Las tablas de símbolos son estructuras de datos que se utilizan para almacenar información sobre construcciones del programa fuente, relaciona nombres léxicos y sus atributos, éstos son información relevante que es recolectada *incrementalmente* por las fases del front-end:

- el lexer creará e inicializará la tabla ya sea con los lexemas o sólo con un apuntador al identificador;
- el parser podrá distinguir las diferentes declaraciones de un mismo identificador, también puede redefinir la tabla para algún símbolo si es necesario;
- el analizador semántico podrá incluir más información sobre cada identificador para completar la tabla.

Características de una Tabla de Símbolos

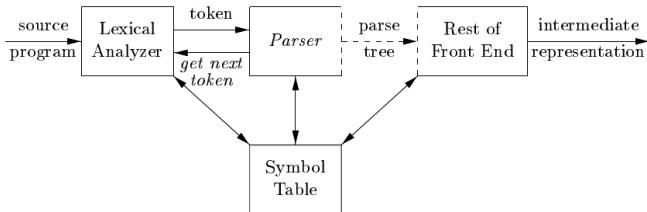
- Será utilizada por las fases del back-end o las fases de síntesis para generar el código objeto.

Características de una Tabla de Símbolos

- Será utilizada por las fases del back-end o las fases de síntesis para generar el código objeto.
- Debe estar diseñada para que el compilador pueda almacenar y recuperar de forma rápida la información sobre identificadores, variables, declaraciones, nombres de procedimientos o funciones, etc.

Características de una Tabla de Símbolos

- Será utilizada por las fases del back-end o las fases de síntesis para generar el código objeto.
- Debe estar diseñada para que el compilador pueda almacenar y recuperar de forma rápida la información sobre identificadores, variables, declaraciones, nombres de procedimientos o funciones, etc.
- Almacenará información sobre cada identificador del programa tales como su lexema, su tipo, la posición de almacenamiento (dirección relativa) y alguna otra información relevante.



Organización y propiedades de una Tabla de Símbolos

- Es una estructura que sólo “vive” durante la compilación y almacena tokens reconocidos con información de ellos.
- Se puede ver como un diccionario:
mapea nombres a información que el compilador ya conoce.

Organización y propiedades de una Tabla de Símbolos

- Es una estructura que sólo “vive” durante la compilación y almacena tokens reconocidos con información de ellos.
- Se puede ver como un diccionario:
mapea nombres a información que el compilador ya conoce.
- Debe soportar las acciones de inserción de nueva información, búsqueda de un identificador (*lookup*), manejo de alcances (*enter_scope* y *leave_scope*).

Organización y propiedades de una Tabla de Símbolos

- Es una estructura que sólo “vive” durante la compilación y almacena tokens reconocidos con información de ellos.
- Se puede ver como un diccionario:
mapea nombres a información que el compilador ya conoce.
- Debe soportar las acciones de inserción de nueva información, búsqueda de un identificador (*lookup*), manejo de alcances (*enter_scope* y *leave_scope*).
- Nunca se borra información ni se elimina la tabla durante la compilación.

Tabla de Símbolos

en compilación

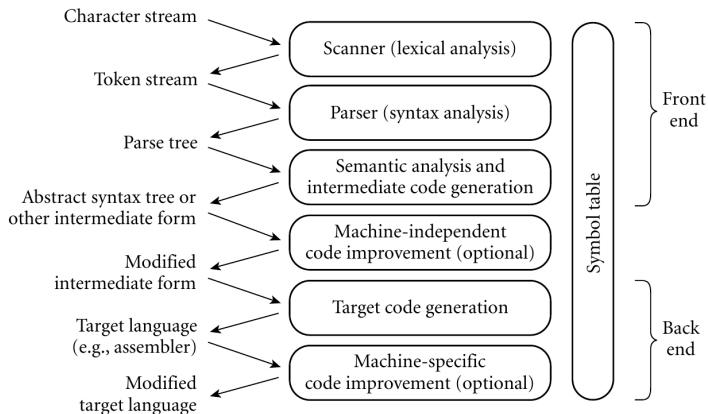
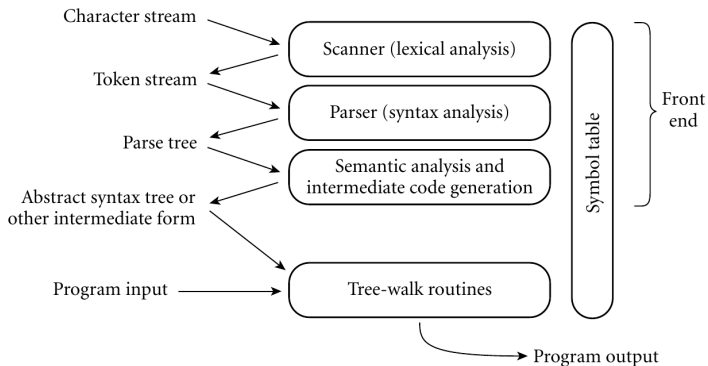


Tabla de Símbolos

en interpretación



Organización de una Tabla de Símbolos

Implementaciones

Lista Lineal

- *lookup* de orden lineal $O(n)$
- fácil de expandir ya que no tiene un tamaño fijo
- una asignación por inserción

Lista ordenada

- *lookup* de orden logarítmico $O(\log_2 n)$
- por cada inserción se debe reorganizar la lista

Árbol binario

- *lookup* lineal si el árbol no es balanceado y logarítmica si es balanceado
- fácil de expandir
- una asignación por inserción

Tabla Hash

- *lookup* constante en promedio
- expansión de la tabla depende del esquema usado

Referencias

Imágenes tomadas del libro [4].

- [1] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman.
Compilers, Principles, Techniques and Tools.
Pearson Education Inc., Second edition, 2007.
- [2] H. R. Nielson and F. Nielson.
Semantics with Applications: An Appetizer (Undergraduate Topics in Computer Science).
Springer-Verlag, Berlin, Heidelberg, 2007.
- [3] F. Pfenning.
Notas del curso (15-411) Compiler Design.
<https://www.cs.cmu.edu/~fp/courses/15411-f14/>, 2014.
- [4] M. L. Scott.
Programming Language Pragmatics.
Morgan-Kaufman Publishers, Third edition, 2009.
- [5] Y. Su and S. Y. Yan.
Principles of Compilers, A New Approach to Compilers Including the Algebraic Method.
Springer-Verlag, Berlin Heidelberg, 2011.
- [6] T. Teitelbaum.
Introduction to compilers.
<http://www.cs.cornell.edu/courses/cs412/2008sp/>, 2008.
- [7] L. Torczon and K. Cooper.
Engineering A Compiler.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2011.
- [8] S. Zdancewic.
Notas del curso (CIS 341) - Compilers, Universidad de Pennsylvania, Estados Unidos.
<https://www.cis.upenn.edu/~cis341/current/>, 2018.