

Unidad de Aprendizaje: Compiladores

UNIDAD TEMÁTICA I

Arquitectura de los compiladores e intérpretes

UNIDAD DE COMPETENCIA

Determina la estructura general de los compiladores intérpretes con base en la funcionalidad de sus etapas y fases.

- 1.1** Definición de compiladores e intérpretes.
- 1.2** Arquitectura de los compiladores e intérpretes.
- 1.3** Máquinas virtuales.
- 1.4** Funcionalidad de las fases de un compilador.
 - 1.4.1** Análisis léxico.
 - 1.4.2** Análisis sintáctico.
 - 1.4.3** Análisis semántico.
 - 1.4.4** Generación de código intermedio.
 - 1.4.5** Optimización de código.
 - 1.4.6** Generación de código objeto.

UNIDAD TEMÁTICA II

Construcción de analizadores léxicos

UNIDAD DE COMPETENCIA

Implementa el proceso de análisis léxico, con base en autómatas finitos deterministas y no deterministas.

- 2.1** Definición de clases léxicas por medio de expresiones regulares.
- 2.2** Creación de AFN (Autómatas finitos no deterministas) por medio de la construcción de Thompson.
- 2.3** Conversión de AFN a AFD (Autómatas finitos deterministas).
- 2.4** Modificación del histograma.
- 2.5** Minimización de estados de un AFD.
- 2.6** Construcción de analizadores léxicos definidos por medio de expresiones regulares.
- 2.7** Generadores de analizadores léxicos (FLEX, JFLEX, JACCIE, GPLEX, etc.).

UNIDAD TEMÁTICA III

Construcción de analizadores sintácticos

UNIDAD DE COMPETENCIA

Implementa analizadores sintácticos, con base en gramáticas libres de contexto y herramientas automatizadas.

3.1 Analizadores sintácticos.

3.1.1 Gramáticas ambiguas.

3.1.2 Eliminación de recursión izquierda en las gramáticas Árboles de sintaxis.

3.1.3 Definición de lenguajes libres del contexto.

3.1.4 Analizadores Sintácticos Descendentes.

3.2 Construcción de analizadores sintácticos por descenso recursivo.

3.2.1 Construcción de analizadores sintácticos LL(1).

3.3 Analizadores Sintácticos Ascendentes.

3.3.1 Analizadores LR(0).

3.3.2 Analizadores LR(1).

3.3.3 Analizadores LALR.

3.4 Generadores de analizadores sintácticos (YACC, JACC, JACCIE, GPPG).

UNIDAD TEMÁTICA IV

Análisis semántico y generación de código intermedio

UNIDAD DE COMPETENCIA

Implementa una aplicación de análisis semántico con base en atributos sintetizados y heredados.

4.1 Atributos.

4.1.1. Atributos heredados.

4.1.2. Atributos sintetizados.

4.2 Diseño de tabla de símbolos para soportar llamadas a funciones y/o procedimientos y ámbito de variables.

4.3 Generación de código de 3 y 4 direcciones.

4.4 Generación de código intermedio para estructuras iterativas y de control (for, while, if-else, switch).

4.5 Generación de código intermedio para el llamado a funciones y procedimientos.

4.6 Manejo del Polimorfismo en un compilador.

UNIDAD TEMÁTICA V

Optimización de código

UNIDAD DE COMPETENCIA

Implementa los generadores de código en el proceso de construcción de compiladores, con base en algoritmos de optimización.

5.1. Introducción a la optimización de código.

5.2 Análisis de flujo de datos.

5.3 Propagación de constantes.

5.4 Eliminación de redundancia parcial.

5.5 Ciclos en grafos de flujos.

5.6 Análisis basado en regiones.

Bibliografía Básica

Autor(es)	Año	Título del documento	Editorial / ISBN
Aho, A. V., Sethi, R., y Ullman, J. D.	2008	<i>Compiladores: Principios, Técnicas y Herramientas (2ª Edición)</i>	Pearson-Addison Wesley. ISBN: 978-970-26-1133-2
Alfonseca, M.	2006	<i>Compiladores e intérpretes: Teoría y práctica (1ª edición)</i>	Pearson Alhambra ISBN: 8420550310
Grune, D., E. Bal. H.	2007	<i>Diseño de compiladores modernos (1ª edición)</i>	Mc Graw Hill ISBN: 978-84-481-5656-5
I. Holub Allen	1990	<i>Compiler Design in C (2ª edición)</i>	Prentice Hall ISBN: 0131550454
Ruiz, J.	2010	<i>Compiladores: Teoría e Implementación</i>	Alfaomega Grupo Editor ISBN: 978-607-7854-68-5