



Organización de Lenguajes y Compiladores 1

Código:	777	Créditos:	4
Escuela:	Ciencias y Sistemas	Área a la que pertenece:	Ciencias de la Computación
Pre-requisito:	771 – Introducción a la programación y Computación 2 796 – Lenguajes Formales y de Programación 962 – Matemática para Computación 2	Post-requisito:	781 – Organización de Lenguajes y Compiladores 2
Categoría:	Obligatorio	Semestre:	Primero 2015
Catedrático:	Ing. Mario Bautista	Auxiliar:	Diego Solis
Edificio:	T-1	Sección:	A
Salón del curso:	L-III-6	Salón del laboratorio:	L-III-6
Horas por semana del curso:	4	Horas por semana del laboratorio:	2
Días que se imparte el curso:	Sábado	Días que se imparte el laboratorio:	Sábado
Horario del curso:	10:50 – 14:10	Horario del laboratorio:	14:10 – 15:40

Descripción del curso:

Organización de Lenguajes y Compiladores 1 implementa los principios básicos de un compilador y/o intérprete, tomando como referencia diferentes herramientas de programación, describiendo las fases de cada proceso.

Se trata en detalle las primeras fases del proceso: análisis lexicográfico, análisis sintáctico y traducción dirigida por la sintaxis. Para poner en práctica los conceptos aprendidos se realizan varias tareas y proyectos prácticos.

Objetivo General:

Que el estudiante adquiera una base teórica fundamental para el entendimiento de la estructura interna del proceso de compilación.

Objetivos Específicos:

1. Que el estudiante aprenda en detalle las primeras fases del proceso de compilación, principalmente el análisis lexicográfico y el análisis sintáctico.
2. Capacitar y ejercitar al estudiante en los principios de análisis, diseño e implementación de compiladores.

Metodología:

Clase magistral dos veces por semana.

Resolución de dudas, problemas, tareas y autoestudio.

Tareas de investigación y prácticas para evidenciar el conocimiento adquirido.

Proyectos de programación.

Requisitos:

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene un valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

Evaluación:

Evaluaciones de rendimiento	30 pts.
Tareas, cortos e investigaciones	05 pts.
Laboratorio	40 pts.

Zona	75 pts.
Examen Final	25 pts.

Total	100 pts.

Contenido:**1. Introducción a la compilación**

1.1 Traductor de lenguajes de computación

1.2 Interpretación

1.3 Compilación

1.4 Conceptos básicos de un compilador

1.5 Estructura interna del proceso de compilación

1.6 Fases del proceso de compilación

2. Análisis Léxico

2.1 Función del analizador léxico

2.2 Especificación de los Tokens

2.2.1 Cadenas y Lenguajes

2.2.2 Operaciones y aplicaciones a lenguajes

2.2.3 Expresiones regulares

2.2.4 Definiciones regulares

2.3 Diagramas de transición

2.3.1 Implantación de diagramas de transición

2.3.2 Autómatas Finitos

2.3.3 Autómatas Finitos No Determinísticos (AFN)

2.3.4 Autómatas Finitos Determinísticos (AFD)

2.4 Construcción de Thompson – Convertir una expresión regular en un AFN

2.5 Construcción de Subconjuntos – Convertir un AFN a un AFD

2.6 Construcción de un AFD a partir de una expresión regular (Método del Árbol)

3. Análisis de Sintaxis

3.1 El papel del analizador sintáctico

3.2 Gramáticas independientes del contexto

3.3 Definiciones y notación

3.4 Derivación

3.5 Árboles de análisis sintáctico y derivaciones

3.6 Ambigüedad

3.7 Escritura de una gramática

3.8 Expresiones regulares o gramáticas independientes del contexto

3.9 Comprobación del lenguaje generado por una gramática

3.10 Reescritura de una gramática

3.10.1 Supresión de la ambigüedad

3.10.2 Eliminación de la recursividad por la izquierda

3.10.3 Factorización por la izquierda

3.11 Funciones PRIMERO y SIGUIENTE

3.12 Lenguajes dependientes del contexto

3.13 Implantación de una gramática (Análisis Semántico)

3.13.1 Representación en código de una gramática

3.13.2 Traducción dirigida por la sintaxis

3.13.2.1 Definición dirigida por la sintaxis

3.13.2.1.1 Atributos sintetizados

3.13.2.1.2 Atributos heredados

3.13.2.2 Construcción de árboles sintácticos

3.13.2.3 Definiciones con atributos por la izquierda

3.13.2.4 Esquemas de traducción

- 3.14 Análisis sintáctico descendente (Top-Down)
 - 3.14.1 Análisis sintáctico descendente recursivo
 - 3.14.2 Analizadores sintácticos predictivos
 - 3.14.3 Análisis sintáctico predictivo no recursivo
 - 3.14.4 Construcción de tablas de análisis sintáctico
 - 3.14.5 Gramáticas LL(1)
 - 3.14.6 Recuperación de errores en el parser predictivo
- 3.15 Análisis sintáctico ascendente (Bottom-Up)
 - 3.15.1 Poda de Handlers
 - 3.15.2 Análisis sintáctico por precedencia de operadores
 - 3.15.2.1 Asociatividad y precedencia
 - 3.15.2.2 Operadores unarios
 - 3.15.2.3 Funciones de precedencia
 - 3.15.3 Analizadores sintácticos LR
 - 3.15.3.1 Construcción de tablas de parsers SLR
 - 3.15.3.2 Operación cerradura
 - 3.15.3.3 Operación GOTO
 - 3.15.3.4 Construcción de conjuntos de elementos
 - 3.15.3.5 Tablas de análisis sintáctico SLR
 - 3.15.4 Construcción de tablas de parsers LR canónicos
 - 3.15.4.1 Construcción de conjuntos de elementos LR(1)
 - 3.15.5 Construcción de tablas de parsers LALR
 - 3.15.5.1 Construcción eficiente de tablas de parsers LALR
 - 3.15.5.2 Determinación de los símbolos de anticipación (LookAhead)
 - 3.15.6 Uso de gramáticas ambiguas
 - 3.15.7 Uso de precedencia y asociatividad para resolver conflictos en las acciones del análisis sintáctico

Bibliografía:

- Compiladores. Principios, técnicas y herramientas. AHO, Alfred. Segunda Edición. Pearson Educación. México, 2008.