



NOMBRE DEL CURSO: Lenguajes Formales y de Programación

CODIGO: 796	CREDITOS: 3
ESCUELA: Ciencias y Sistemas	AREA A LA QUE PERTENECE: Ciencias de la Computación
PRE REQUISITOS: 770 – Introducción a la Programación 1 795 – Lógica de sistemas 960 – Matemática de cómputo 1	POST REQUISITO: Organización de lenguajes y compiladores 1 [777] Organización computacional[964]
CATEGORIA: Obligatorio	SEMESTRE: 2do. 2014
CATEDRÁTICO (A): Ing. Otto Amílcar Rodríguez Acosta	AUXILIAR: Mario Antonio Guzmán Garzona
EDIFICIO: T-3	SECCIÓN: A+
SALON DEL CURSO: 216	SALON DEL LABORATORIO: 211 T-3
HORAS POR SEMANA DEL CURSO: 2 HORAS	POR SEMANA DEL LABORATORIO: 2
DÍAS QUE SE IMPARTE EL CURSO: Martes	DIAS QUE SE IMPARTE EL LABORATORIO: Miércoles
HORARIO DEL CURSO: 07:10AM – 08:50AM	HORARIO DEL LABORATORIO: 08:00AM – 09:40AM

Descripción del curso:

El laboratorio tiene como propósito el poner en práctica los contenidos y las diversas áreas cubiertas en el curso de lenguajes formales y de programación, así como también el complementar los mismos. Durante el laboratorio el estudiante aprenderá a implementar de forma correcta componentes básicos de un compilador y las fases de análisis léxico, sintáctico y semántico. El estudiante deberá de diseñar gramáticas para representar lenguajes y utilizar diversas técnicas de programación y métodos para resolverlas.

OBJETIVOS:

Objetivo General

- Introducir al estudiante al conocimiento y desarrollo de las funciones básicas de los compiladores.

Objetivos Específicos

- Implementar un analizador léxico.
- Implementar un analizador sintáctico.
- Usar de manera eficiente la tabla de símbolos.
- Conocer herramientas que faciliten la creación de un compilador.

Metodología

El laboratorio consiste de clases presenciales en las cuales se complementara el contenido del curso y se utilizarán los conocimientos adquiridos, adicionalmente se realizaran prácticas, proyectos, tareas, exámenes cortos y hojas de trabajo con el objetivo de mejorar las habilidades de los estudiantes en el diseño y construcción de compiladores.

- Copias parciales o totales en las tareas de textos para las tareas, investigaciones, etc., serán sancionadas.
- Copias en los proyectos y prácticas serán sancionadas y reportadas a la escuela.
- Las tareas, investigaciones, etc., enviadas deben de poseer el formato establecido y en la fecha establecida.

Evaluación

La nota final estará compuesta de la siguiente manera:

Tareas y Cortos.....	10 puntos
Práctica 1.....	10 puntos
Práctica 2.....	15 puntos
Primer Proyecto	20 puntos
Segundo Proyecto	35 puntos
Examen Final.....	10 puntos

Total: 100 puntos

Observaciones

- El laboratorio se debe de aprobar con una nota mínima de 61 puntos.
- Solo se calificaran exámenes, proyectos y demás actividades, a estudiantes asignados.
- No se guardan notas para semestres posteriores, no se pasan notas de semestres anteriores, y no se aceptan estudiantes con problemas de prerrequisitos.

Contenido

1. Lenguajes

- 1.1 Lenguajes Naturales
- 1.2 Características de lenguajes naturales
- 1.3 Lenguajes Formales
- 1.4 Características de lenguajes formales

2. Introducción a compiladores

- 2.1 ¿Qué es un compilador?
- 2.2 Ejemplos de compiladores
- 2.3 Partes del compilador
- 2.4 Etapa de análisis
- 2.5 Etapa síntesis

3. Análisis léxico

- 3.1 Análisis lexicográfico.
- 3.2 Token
- 3.3 Lexema

3.4 Patrón

4. Lenguajes regulares

4.1 Lenguajes regulares

4.2 Expresiones regulares

4.3 Ejemplos.

5. Métodos para Calcular DFAs

5.1 Método del árbol

5.2 Construcción de Thomson y minimización de estados

5.3 Ejemplos y ejercicios

6. Autómatas con pila

6.1 Definición

6.2 Formalización

6.3 Traducción

6.4 Ciclos en los AFS

6.5 AFS de un estado

6. Implementación de un analizador léxico

6.1 Tabla de símbolos

6.2 Manejo de errores

7. Lenguajes libres de contexto

7.1 Lenguajes Independientes del contexto

7.2 Gramáticas Tipo 2

7.3 Diseño de gramáticas independientes del contexto

7.3.1 Ejemplos y ejercicios de gramáticas libres de contexto

7.4 Recursividad

7.4.1 Por la izquierda

7.4.2 Por la derecha

7.5 Gramáticas ambiguas

7.6 Parser recursivo descendente

7.7 Ejemplos y ejercicios

7.8 Método para generación de tabla de análisis descendente.

8. Análisis semántico

8.1 Introducción a conceptos de la fase de análisis semántico

8. Herramientas para la fase de análisis

8.1 Introducción a JLex

8.2 Introducción a Cup

BIBLIOGRAFÍA:

- Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas y herramientas. Addison-Wesley.
- Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana.
- Appleby, Doris y Vandekopple, Julius J. Lenguajes de Programación - Paradigma y práctica. Editorial McGraw Hill. 3. Louden, Kenneth C. Construcción de compiladores/ Construction of compilers: Principios y práctica. 2004.
- 4. Eliza Viso G. Teoría de Computación. UNAM 2002

LISTA DE CATEDRÁTICOS

CURSO	SEC	EDI	SALON	INICIO	FIN	L	M	MI	J	V	CATEDRATICOS
LENGUAJES FORMALES Y DE PROGRAMACIÓN	A+	T-3	211	9:10	10:50	-	-	-	-	X	MARIO GUZMÁN