



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD CIENCIAS DE LA COMPUTACIÓN

PROGRAMA DE LA MATERIA CORRESPONDIENTE A LA INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN.

Coordinación: Área de Software de Base

NOMBRE DE LA MATERIA:	Compiladores
-----------------------	--------------

Clave: CCO 511

Créditos: 10

Modalidad: Escolarizada

Nivel de Ubicación: Formativo

Tipo de Materia: Optativa

PRE-REQUISITOS:	Nivel Básico
-----------------	--------------

MATERIA CONSECUENTE:	Ninguna
----------------------	---------

TIEMPO TOTAL ASIGNADO:	80 hrs.
------------------------	---------

PRIMAVERA - OTOÑO

HRS. TEÓRICAS/SEM:	5	HRS. PRÁCTICAS/SEM:	0
--------------------	---	---------------------	---

VERANO

HRS. TEÓRICAS/SEM:	10	HRS. PRÁCTICAS/SEM:	0
--------------------	----	---------------------	---

AUTOR(ES) DEL PROGRAMA:	
-------------------------	--

Pedro Bello López	Jesús García Fernández
-------------------	------------------------

Hilda Castillo Zacatelco	
--------------------------	--

Yalú Galicia Hernández	
------------------------	--

REVISADO POR:	Área de Software de Base
---------------	--------------------------

APROBADO POR:	Academia
---------------	----------

AUTORIZADO POR:	Docencia
-----------------	----------

FECHA DE ELABORACIÓN/REVISIÓN:	Julio 2001 / Junio 2003 / Julio 2004
--------------------------------	--------------------------------------

VIGENCIA:	Julio de 2001
-----------	---------------

JUSTIFICACIÓN:

En la creación o construcción de compiladores para algún lenguaje de programación es importante tomar en cuenta las diferentes técnicas que existen para ello, además el desarrollador puede sacar

provecho de tales técnicas para desarrollar software de propósito general.

OBJETIVO GENERAL DE LA MATERIA:

Que el alumno diseñe un compilador utilizando las técnicas vistas en el curso, e identifique lo importante que es la separación del compilador en fases.

CONTRIBUCIÓN DE LA SIGNATURA AL PERFIL DE EGRESO:

El alumno al concluir sus estudios estará preparado para seleccionar técnicas de análisis o síntesis para el diseño de software de propósito general.

CONTENIDO TEMÁTICO

UNIDAD: 1				TÍTULO: INTRODUCCIÓN		
OBJETIVO ESPECÍFICO: El estudiante identificará y analizará los elementos principales requeridos en el diseño de compiladores Bibliografía: [1,3,4 y 5]						
CONTENIDO DE LA UNIDAD		Tiempo de impartición (hrs).		Actividades de Aprendizaje	Técnicas	Recursos Necesarios
		HT	HP			
1.1	Elementos de un compilador, herramientas, utilerías y algoritmos necesarios de	2	0	Actividades de aplicación del conocimiento y desarrollo de capacidades de análisis y síntesis	Exposición del profesor, discusión grupal, uso de analogías.	Salón, pizarrón, plumones, proyector de acetatos o cañón y laptop
1.2	Diseño de una máquina virtual	3	0	Actividades para el desarrollo de capacidades de análisis y síntesis, de reflexión, integración y generalización y para el desarrollo de capacidades de valoración y creatividad	Exposición del profesor, planteamiento de un problema, discusión grupal, lluvias de ideas	Idem
HORAS TOTALES:		5	0			

UNIDAD: 2

TÍTULO: ANÁLISIS LÉXICO Y SINTÁCTICO

OBJETIVO ESPECÍFICO:

El alumno experimentará con alguna técnica de análisis léxico y sintáctico, para la creación de un programa que analice léxica y sintácticamente un programa fuente, además identificará los nexos que existen con la teoría de lenguajes formales.

Bibliografía [1,2,3 ,4 y 5]

CONTENIDO DE LA UNIDAD		Tiempo de impartición (hrs).		Actividades de Aprendizaje	Técnicas	Recursos Necesarios
		HT	HP			
2.1	El papel del análisis léxico y sintáctico y su correspondiente problema en los compiladores?	4		Identificar el papel de análisis léxico y sintáctico durante la compilación de un programa. Discutir el porqué de su separación.	Exposición por parte del profesor, discusión y lluvia de ideas.	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo
2.2	Expresiones LEX y tablas de análisis léxico 2.2.1 Conversión de expresiones regulares a AFN 2.2.2 Conversión de AFN a AFD 2.2.3 Conversión de AFD a AFD mínimo	16		Identificar técnicas para la conversión de autómatas, analizar problemas que puedan ser resueltos con autómatas y construir el autómata. Aplicar las técnicas de conversión de autómatas para llevar un AFN a un AFD.	Exposición por parte del profesor, resolver problemas con autómatas y aplicar técnicas de conversión. Trabajo por equipo para resolver problemas.	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo
2.3	Gramáticas libres y reconocedores determinísticos 2.3.1 Autómata de pila 2.3.2 Descendencia recursiva 2.3.3 Precedencia 2.3.4 LL 2.3.5 LR(0), LR(1) y LALR	37		Identificar técnicas para representar la sintaxis de un lenguaje mediante una GLC. Crear una gramática para construir un lenguaje de programación sencillo. Construir un programa prototipo que analice léxica y	Exposición por parte del profesor; en equipos de alumnos aplicar técnicas para crear gramáticas GLC, aplicar técnicas para obtener una Gramática LL(1). Construir un	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo

				sintácticamente un programa fuente.	analizador léxico y sintáctico.	
		HORAS TOTALES:	57			

UNIDAD: 3	TÍTULO: TRADUCCIÓN DIRIGIDA POR SINTAXIS
------------------	---

OBJETIVO ESPECÍFICO:

El alumno examinará la traducción de lenguajes guiada por gramáticas independientes del contexto asociando atributos a los símbolos de la gramática calculados mediante "reglas semánticas", además identificar las diversas formas de crear código intermedio.

Bibliografía [1,3,4 y 5]

CONTENIDO DE LA UNIDAD		Tiempo de impartición (hrs).		Actividades de Aprendizaje	Técnicas	Recursos Necesarios
		HT	HP			
3.1	Diversos códigos intermedios (triples, cuádruples, árbol, notación sufija, digráficas acíclicas)	2		Identificar los diversos tipos de códigos intermedios para el proceso de creación de un compilador.	Exposición por parte del profesor, discusión y lluvia de ideas.	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo
3.2	Esquema abstracto de traducción	4		Aplicar un esquema de traducción a una gramática independiente del contexto asociando atributos con los símbolos gramaticales por medio de acciones semánticas, los esquemas de traducción pueden ser tanto atributos sintetizados como heredados.	Exposición por parte del profesor, resolver problemas aplicando esquemas de traducción simples y menos simples con una gramática independiente del contexto	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo

3.3	Acciones semánticas. 3.3.1. Expresiones aritméticas, expresiones booleanas, expresiones mixtas. 3.3.2. Generación de código y ejecución en línea (interprete). 3.3.3. Proposiciones de control (if-then else, do-while, case). 3.3.4. Proposiciones declarativas, registros, arreglos, referencia a arreglos, referencia a registros, caso de PL/I. 3.3.5. Invocación y paso de parámetros.	4		Aplicara acciones semánticas a expresiones aritméticas, booleanas y mixtas, así mismo identificara las formas de crear código a los diferentes tipos de proposiciones, paso de parámetros en funciones que existen en un lenguaje de programación.	Exposición por parte del profesor; en equipos de alumnos aplicar acciones semánticas a su proyecto de compiladores.	Pizarrón, borrador, proyector de acetatos, cañón y equipo de computo
HORAS TOTALES:		10				

UNIDAD: 4	TÍTULO: OPTIMIZACIÓN DE CÓDIGO Y GENERACIÓN DE CÓDIGO OBJETO
------------------	---

OBJETIVO ESPECÍFICO:

El estudiante reconocerá y analizará las diferentes técnicas de generación y optimización de códigos objeto y su aplicación.
Bibliografía: [1,3,4 y 5]

CONTENIDO DE LA UNIDAD	Tiempo de impartición (hrs).		Actividades de Aprendizaje	Técnicas	Recursos Necesarios
	HT	HP			
4.1 Bloques	1	0	Actividades de introducción, de comprensión y de reflexión, integración y generalización.	Exposición del profesor, discusión grupal y asociación con conocimientos previos.	Salón, pizarrón, plumones, proyector de acetatos o cañón y laptop
4.2 Determinación de ciclos	2	0	Actividades para el desarrollo de capacidades de análisis y síntesis, de reflexión,	Idem	Idem

				integración y generalización y para el desarrollo de capacidades de valoración y creatividad.		
4.3	Inducción de variables	2	0	Idem	Idem	Idem
4.4	Generación de código objeto, uso de DAGS, coloración	3	0	Idem	Exposición del profesor, discusión grupal, asociación con conocimientos previos y optimizar una solución.	Idem
HORAS TOTALES:		8	0			

PRACTICAS

UNIDAD	NOMBRE DE LA PRACTICA	OBJETIVO	HORAS

CRITERIOS DE EVALUACIÓN

PARCIALES DEPARTAMENTALES

Parcial	Contenido a evaluar	Periodos
1	Unidad 1 y parte de Unidad 2	6ª. Semana
2	Unidad 2	11ª. Semana
3	Unidad 3 y Unidad 4	16ª. Semana

	%
Asistencias:	A criterio del profesor
Proyecto Final:	A criterio del profesor

Tareas:	A criterio del profesor
Trabajos de Investigación:	A criterio del profesor
Prácticas de Laboratorio:	A criterio del profesor
TOTAL:	100

REQUISITOS DE ACREDITACIÓN:

A criterio del profesor

FOMENTO DE VALORES:

Crear en el estudiante el hábito del análisis y la búsqueda de soluciones a problemas de cómputo.

BIBLIOGRAFÍA:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, "Compiladores. Principios, técnicas y herramientas.". Pearson Educación, 1998. (B)
2. Dean Kelley, "Teoría de Autómatas y lenguajes formales", Prentice may, 1995. (C)
3. Jean-Paul Tremblay, Paul G. Sorenson, "The theory and practice of compiler writing", McGraw-Hill Computer science series, 1985. (C)
4. Karen A. Lemone, "Fundamentos de Compiladores", Compañía Editorial Continental S.A, de C.V. (CECSA), 1996. (B)
5. Thomas A. Sudkamp, "lenguages and Machines", second edition, Addison Wesley, 1997. (C)

B: Básico

C: Complementario