

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

COORDINACIÓN DE FORMACIÓN BÁSICA COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA PROGRAMA DE UNIDADES DE APRENDIZAJE POR COMPETENCIAS

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: FACULTAD DE CIENCIAS.
2. Programa (s) de estudio: Licenciatura en Ciencias Computacionales
3. Vigencia del plan: 2008-1
4. Nombre de la Asignatura: Teoría de la computación
5. Clave: 9832
6. HC: 2 HL 2 HT 2 HPC HCL HE 2 CR 8
7. Ciclo Escolar: 2009-1
8. Etapa de formación a la que pertenece: Disciplinaria
9. Carácter de la Asignatura: Obligatoria X Optativa
10. Requisitos para cursar la asignatura: Se recomienda haber cursado Introducción a las matemáticas y Matemáticas Discretas

Formuló: M.C. María Victoria Meza Kubo
L.C.C. Pedro Perez

VoBo. Marcelo Rodríguez Meraz.
Cargo: Subdirector

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Este curso muestra la construcción matemática de la teoría de la computación, mediante el diseño de autómatas y lenguajes abstractos como base para la construcción de traductores y compiladores.

El curso es obligatorio y se encuentra al final de la etapa disciplinaria una vez que se han cursado los cursos básicos de matemáticas, como son Introducción a las matemáticas y Matemáticas Discretas, así como cursos disciplinarios de programación, como Introducción a la programación, Estructuras de datos y Algoritmos, Programación Orientada a Objetos, ya que aquí se combinarán ambas áreas para la construcción de máquinas abstractas.

Las asignaturas subsecuentes relacionadas con esta son: Teoría de la computación II (optativa) y Compiladores.

III. COMPETENCIA (S) DE LA UNIDAD DE APRENDIZAJE

Construir autómatas de estado finito y lenguajes regulares, aplicando los conocimientos formales que sustentan el modelo teórico y conceptual de las computadoras y del quehacer computacional en su sentido más amplio, como base para la construcción de traductores y compiladores, abstrayendo comportamientos naturales a modelos teóricos.

IV. EVIDENCIA (S) DE DESEMPEÑO

1. El alumno diseñará y construirá gramáticas, autómatas de estado finito y expresiones regulares, mediante la realización de ejercicios en papel y el desarrollo de programas de computadora que muestren gráficamente los ejercicios realizados en clase.
2. Presentación y reporte de un proyecto final realizado en equipos que incluya las prácticas de laboratorio integradas como un solo sistema. Este sistema deberá mostrar por tablas y gráficamente las relaciones que guardan las gramáticas regulares, autómatas de estado finito y lenguajes regulares.

V. DESARROLLO POR UNIDADES

Competencia

Comprender los conceptos y notaciones de máquinas y lenguajes abstractos mediante su estudio y desarrollo de ejercicios como base para su diseño y construcción tomando una actitud de abstracción de los conceptos y operaciones básicos.

Contenido temático

I. Introducción

- 1.1 Máquinas de información
- 1.2 Máquinas abstractas y algoritmos
- 1.3 Lenguajes abstractos
- 1.4 Aceptadores, generadores y traductores
- 1.5 Cadenas de lenguajes
- 1.6 Cadenas y sus operaciones
- 1.7 Operaciones sobre lenguajes

Duración

2 horas

Competencia

Comparar las principales clases de gramáticas, construyendo gramáticas generadoras de lenguajes para conocer la relación que guardan con las clases de lenguajes de la jerarquía de Chomsky, realizando una síntesis de sus diferencias claves.

Contenido temático

II. Gramáticas formales

- 2.1 Representación de los lenguajes
- 2.2 Conceptos básicos de gramáticas
- 2.3 Gramáticas formales
- 2.4 Tipos de gramáticas
 - 2.4.1 Gramáticas sin restricción
 - 2.4.2 Gramáticas de contexto sensitivo
 - 2.4.3 Gramáticas libres del contexto
 - 2.4.4 Gramáticas regulares
- 2.5 Árboles y diagramas de derivación
- 2.6 Ambigüedad

Duración

8 horas

Competencia

Construir autómatas de estado finito con la solución de ejercicios de manera creativa y transformando máquinas equivalentes como base para la construcción de automatas más complejos con el uso de abstracciones del lenguaje.

<p>Contenido temático</p> <p>III. Máquinas de estado finito</p> <ul style="list-style-type: none"> 3.1 Propiedades de las máquinas de estado finito <ul style="list-style-type: none"> 3.1.1 Máquinas de transición 3.1.2 Máquinas de estado 3.2 Conversión entre máquinas de transición y estados 3.3 Equivalencia de máquinas de estado finito 3.4 Estados equivalentes 3.5 Reducción de estados y prueba de equivalencia 3.6 Historia de máquinas y memoria finita 	<p>Duración</p> <p>12 horas</p>
<p>Competencia</p> <p>Diseñar máquinas abstractas, lenguajes generados por gramáticas formales y expresiones regulares mediante la solución de ejercicios para analizar las relaciones que guardan entre ellos comprendiendo la abstracción de los lenguajes.</p>	
<p>Contenido temático</p> <p>IV. Lenguajes de estado finito</p> <ul style="list-style-type: none"> 4.1 Aceptadores de estado finito 4.2 Aceptadores de estado finito y gramáticas regulares 4.3 Expresiones regulares y aceptadores de estado finito 4.4 Propiedades de los lenguajes de estado finito 	<p>Duración</p> <p>16 hrs.</p>
<p>Competencia</p> <p>Analizar el alcance de los analizadores de sintaxis Pushdown con relación a los lenguajes independientes al contexto mediante el desarrollo de los algoritmos que dejarán las bases para la construcción de compiladores y traductores, realizando síntesis de problemas dados.</p>	

<p>Contenido temático V. Autómatas Pushdown 5.1 Introducción 5.1.1 Autómatas de cinta 5.2 Aceptadores "pushdown". 5.3 Aceptadores propios y sin ciclos. 5.4 Aceptadores "pushdown" para lenguajes independientes al contexto. 5.4.1 Análisis de sintaxis. 5.4.2 Construcción de analizadores de sintaxis "pushdown". 5.4.3 Prueba de la construcción. 5.5 Gramáticas independientes al contexto para aceptadores "pushdown". 5.6 Ambigüedad.</p>	<p>Duración 8 horas.</p>
<p>Competencia Estudiar las diferentes formas y estructuras de las gramáticas independientes al contexto para desarrollar gramáticas eficientes.</p>	
<p>Contenido temático VI. Lenguajes independientes del contexto 6.1 Transformaciones elementales de gramáticas. 6.1.1 Equivalencia de gramáticas. 6.1.2 Transformación de gramáticas mediante substitución y expansión 6.1.3 Eliminación de producciones inútiles. 6.1.4 Reemplazo de producciones no generativas. 6.1.5 Gramáticas bien formadas. 6.2 Formas canónicas de gramáticas. 6.2.1 Gramáticas en forma normal. 6.2.2 Gramáticas en forma estándar. 6.3 Estructura de los lenguajes independientes al contexto.</p>	<p>Duración 12 horas</p>

IV. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Analizar los conceptos y notaciones de máquinas y lenguajes abstractos mediante su estudio y desarrollo de ejercicios como base para su diseño y construcción.	El alumno resolverá problemas de conjuntos, igualdades y operaciones de lenguajes.	Papel, lápiz, pizarrón y plumones	2 horas
2	Comparar las principales clases de gramáticas, construyendo gramáticas generadoras de lenguajes para conocer la relación que guardan con las clases de lenguajes de la jerarquía de Chomsky.	El alumno diseñará gramáticas, de acuerdo a la jerarquía de Chomsky, que generen los lenguajes solicitados	Papel y lápiz.	2 horas
3	Comparar las principales clases de gramáticas, construyendo gramáticas generadoras de lenguajes para conocer la relación que guardan con las clases de lenguajes de la jerarquía de Chomsky.	El alumno desarrollará en equipo un programa que reciba una gramática de entrada y que de forma automática la clasifique conforme a la jerarquía de Chomsky, y permita la generación de cadenas terminales, ilustrando cada paso de derivación.	Computadora.	4 horas

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
4	Estudiar la formulación matemática del autómata más simple: máquina de estado finito con la solución de ejercicios de manera creativa y transformando máquinas equivalentes.	El alumno diseñará máquinas de estado finito que cumplan con las propiedades solicitadas.	Apuntes, papel, lápiz, pizarrón y plumones.	2 horas
5	Estudiar la formulación matemática del autómata más simple: máquina de estado finito con la solución de ejercicios de manera creativa y transformando máquinas equivalentes.	El alumno desarrollará en equipo un programa de computadora que reciba de entrada una máquina de estado finito con salidas en las transiciones o los estados y obtenga una máquina de estados o transiciones equivalente, respectivamente. Mostrar los resultados por tablas y en forma gráfica.	Computadora, apuntes	6 horas
6	Estudiar la formulación matemática del autómata más simple: máquina de estado finito con la solución de ejercicios de manera creativa y transformando máquinas equivalentes.	El alumno desarrollará en equipo un programa de computadora que reciba de entrada una máquina de estado finito, la particione y obtenga una máquina reducida y conectada equivalente. Mostrar todos los pasos.	Apuntes, Computadora	4 horas

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
7	Diseñar máquinas abstractas, lenguajes generados por gramáticas formales y expresiones regulares mediante la solución de ejercicios para analizar las relaciones que guardan entre ellos.	El alumno desarrollará en equipo un programa de computadora que reciba un aceptador de estado finito y obtenga su equivalente determinístico	Apuntes, Computadora.	4 horas
8	Diseñar máquinas abstractas, lenguajes generados por gramáticas formales y expresiones regulares mediante la solución de ejercicios para analizar las relaciones que guardan entre ellos.	El alumno desarrollará en equipo un programa de computadora que reciba de entrada un aceptador de estado finito, M, y obtenga una gramática regular, G, tal que $L(G)=L(M)$ y viceversa.	Apuntes, computadora	6 horas
9	Diseñar máquinas abstractas, lenguajes generados por gramáticas formales y expresiones regulares mediante la solución de ejercicios para analizar las relaciones que guardan entre ellos.	El alumno desarrollará en equipo un programa de computadora que reciba un aceptador λ y obtenga su aceptador de estado finito equivalente.	Apuntes, computadora	4 horas
10	Analizar el alcance de los autómatas de cinta con relación a los lenguajes independientes al contexto.	El alumno desarrollará en equipo un programa de computadora que reciba un autómata de cinta y dada una cadena de entrada imprima la respuesta correspondiente.	Apuntes, computadora	2 horas
11	Analizar el alcance de los autómatas de cinta con relación a los lenguajes independientes al contexto	El alumno desarrollará en equipo un programa de computadora que reciba un autómata de cinta y de un aceptador λ equivalente.	Apuntes, computadora	2 horas

VII. METODOLOGÍA DE TRABAJO

■ Aprendizaje participativo

Durante la clase se aplicará esta metodología en la que el estudiante juega un papel activo al intervenir propositivamente en la planeación, realización y evolución del proceso de aprendizaje. Consiste básicamente en asignar problemas, el cual primeramente es analizado en forma individual, posteriormente en equipo, después se resuelven los problemas en clase. La participación del maestro en la aplicación de esta metodología es de mediador.

■ Trabajo en equipo

A lo largo del semestre se estará trabajando en equipo, tanto para los trabajos en clase, así como para las prácticas y el proyecto final, y consiste básicamente en asignar problemas para la cual deberán organizarse y resolverlos. El maestro en esta metodología juega el papel de mediador al igual que en la anterior.

■ Clase expositiva

Esta en el caso del alumno, será aplicada sobretodo en la exposición de la solución de los problemas asignados, en el caso del maestro se aplicará en la impartición de conceptos básicos, explicación de ejercicios, así como en la conclusión de temas.

Se requerirá que los estudiantes realicen extraclase la resolución de los ejercicios que se dejen, así como que utilicen horas de cómputo extra para la realización de las prácticas, por lo menos requerirán 4 horas a la semana extra de uso de computadora cuando sean asignadas prácticas de laboratorio.

VIII. CRITERIOS DE EVALUACIÓN

Criterio de calificación

■ Exámenes parciales	60%
■ Prácticas	30%
■ Proyecto final	10%
TOTAL	100%

Criterio de acreditación

- Resolver cuatro exámenes parciales en tiempo y forma.
- Cumplir con las prácticas de laboratorio en tiempo y forma.
- Cumplir con la presentación y reporte de un proyecto en tiempo y forma.
- Cumplir con el 80% de asistencia como está establecido en el estatuto, este es un requisito incluso para tener derecho a examen parcial.

Criterio de evaluación

- Tanto para el caso de las tareas como el de los exámenes, ambos serán resueltos en clase posterior para retroalimentar el desarrollo del curso.
- Deberán entregarse el 70% de prácticas de laboratorio para tener derecho a examen final.
- En el caso del proyecto final por equipo, la evaluación se dividirá en dos: reporte y desarrollo, en el primer caso los puntos a evaluar serán, contenido, limpieza, así como ortografía; para el segundo caso los puntos a evaluar serán, número de prácticas, fácil de utilizar e integración de las prácticas.
- Se busca con el proyecto en equipo formar valores de responsabilidad, búsqueda de la calidad, sentido de justicia, así como valores de síntesis y atracción.

IX. BIBLIOGRAFÍA

Básica

Complementaria

Denning, Dennis, qualitz. Machines, Languajes, And Computation.
Prentice may, 1978.

Hofcroft J. E. Ullman J. D. Introduction to Autómata Theory,
Languajes and Computation.
Addison Wesley Pub. Co. 1979.

Harrison M. A. Introduction to Formal languajes theory.
Addison Wesley Pub. 1978.