

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: FACULTAD DE CIENCIAS.
2. Programa (s) de estudio: Licenciatura en Ciencias Computacionales 3. Vigencia del plan:
4. Nombre de la Asignatura: Teoría de Autómatas 5. Clave:
6. HC: 1 HL 2 HT 2 HPC HCL HE 1 CR 6
7. Etapa de formación a la que pertenece: Disciplinaria
8. Carácter de la Asignatura: Obligatoria X Optativa
9. Requisitos para cursar la asignatura:

Formuló: Dra. María Victoria Meza Kubo

Vo. Bo. Dr. Alberto Leopoldo Morán y Solares

Fecha: Agosto de 2016

Cargo: Subdirector

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

Esta asignatura 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.

Los fundamentos matemáticos de la teoría de autómatas serán de gran utilidad para el profesionalista en ciencias computacionales ya que le permitirá sustentar de manera formal los problemas que se le presenten relacionados a los lenguajes y la abstracción formal de máquinas de información.

La asignatura es obligatoria y se encuentra en la etapa disciplinaria y se recomienda haber cursado las asignaturas básicas de matemáticas, como Álgebra superior y Matemáticas Discretas, así como las asignaturas 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.

Esta asignatura está seriada con la asignatura subsecuente de 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, como base para la construcción de traductores y compiladores, con actitud crítica y propositiva.

IV. EVIDENCIA (S) DE DESEMPEÑO

Entrega compendio de prácticas de laboratorio con implementación de autómatas solicitadas durante el curso, cada una de ellas en código fuente y ejecutable, acompañadas de la documentación correspondiente.

V. DESARROLLO POR UNIDADES

Competencia

Utilizar 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

1. Introducción a los autómatas finitos

1.1 Introducción

1.2 Autómatas y complejidad

1.3 Introducción a las demostraciones

1.4 Conceptos fundamentales de la teoría de autómatas

Duración

2 horas

Competencia

Identificar las diferentes formas y estructuras de las gramáticas independientes al contexto, a través del desarrollo de ejercicios y programas de computadora, para desarrollar gramáticas eficientes, con creatividad.

Contenido temático

2. Lenguajes y gramáticas formales

2.1 Gramáticas formales

2.2 Árboles de derivación

2.3 Aplicaciones de gramáticas independientes del contexto

2.4 Ambigüedad en gramáticas y lenguajes

Duración

4 horas

<p>Competencia Diseñar aceptadores de estado finito analizando su estructura para entender la relación que guardan con gramáticas regulares, con creatividad y responsabilidad.</p>	
<p>Contenido temático 3. Autómatas finitos 3.1 Introducción a los autómatas finitos 3.2 Autómata finito determinístico 3.3 Autómata finito no determinístico 3.4 Autómatas finitos con transiciones-λ</p>	<p>Duración 4 horas</p>
<p>Competencia Diseñar máquinas abstractas, lenguajes generados por gramáticas formales y expresiones regulares, mediante la solución de ejercicios de conversión, para analizar las relaciones que guardan entre ellos, con responsabilidad y eficiencia.</p>	
<p>Contenido temático 4. Lenguajes y expresiones regulares 4.1 Expresiones regulares 4.2 Autómatas finitos y expresiones regulares 4.3 Aplicaciones de las expresiones regulares</p>	<p>Duración 4 hrs.</p>
<p>Competencia Desarrollar aplicaciones de automatas Pushdown, mediante el desarrollo de los algoritmos que dejarán las bases para la construcción de compiladores y traductores, para analizar el alcance de los analizadores de sintaxis Pushdown con relación a los lenguajes independientes al contexto, con responsabilidad y eficiencia.</p>	
<p>Contenido temático 5. Propiedades de los lenguajes regulares 5.1 Demostraciones de lenguajes no regulares 5.2 Propiedades de clausura y decisión de los lenguajes regulares 5.3 Equivalencia y minimización</p>	<p>Duración 2 horas.</p>

IV. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1 (Taller)	Analizar los conceptos y notaciones de máquinas y lenguajes abstractos mediante su estudio y desarrollo de ejercicios como base para su diseño y construcción, con actitud responsable.	Resolver problemas de conjuntos, igualdades y operaciones de lenguajes.	Papel, lápiz, pizarrón y plumones	2 horas
2 (Taller)	Realizar demostraciones de los teoremas básicos para pruebas de cerradura aplicando los principales teoremas con actitud crítica y propositiva.	Realizar demostraciones en conjunto de los principales teoremas vistos en clase	Papel y lápiz.	4 horas
3 (Taller)	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, con actitud crítica y responsable.	Diseñar gramáticas y hará su clasificación de acuerdo a la jerarquía de Chomsky.	Papel, lápiz, pizarrón y plumones.	6 horas
4 (Laboratorio)	Diseñar programas utilizando la manipulación y clasificación de gramáticas formales para entender su comportamiento, con actitud crítica y responsable.	Desarrollar 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	8 horas
5 (Taller)	Diseñar máquinas de estado finito a través de la solución de ejercicios, para identificar su alcance en la	Diseñar máquinas de estado finito que cumplan con las propiedades solicitadas.	Apuntes, papel, lápiz, pizarrón y	4 horas

	abstracción formal de problemas de lenguajes, de manera creativa		plumones.	
6 (Laboratorio)	Diseñar programas de computadora para la manipulación de máquinas de estado finito, para identificar su alcance en la abstracción formal de problemas de lenguajes, de manera creativa	Desarrollar 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.	Computadora , apuntes	8 horas
7 (Taller)	Diseñar aceptadores de estado finito con la solución de ejercicios, para identificar su alcance en la abstracción formal de problemas de lenguajes, de manera creativa	Diseñar aceptadores de estado finito que cumplan con las propiedades solicitadas.	Apuntes, papel, lápiz, pizarrón y plumones.	4 horas
8 (Laboratorio)	Diseñar programas de computadora a través de la manipulación de aceptadores de estado finito, para identificar su alcance en la abstracción formal de problemas de lenguajes, de manera creativa.	Desarrollar en equipo un programa de computadora que reciba de entrada un aceptador de estado finito No determinístico y lo convierta a determinístico.	Apuntes, Computadora	8 horas
9 (Taller)	Diseñar aceptadores de estado finito a partir de expresiones regulares, con la solución de ejercicios, para identificar su alcance en la abstracción formal de problemas de lenguajes, de manera creativa.	Diseñar aceptadores de estado finito que cumplan con las propiedades de una expresión regular.	Apuntes, papel, lápiz, pizarrón y plumones.	4 horas
10 (Taller)	Diseñar aceptadores de estado finito a partir de gramáticas regulares, con la solución de ejercicios, para identificar la relación que guardan en la solución de problemas de lenguajes formales, de manera creativa.	Diseñar aceptadores de estado finito a partir de gramáticas regulares y viceversa.	Apuntes, papel, lápiz, pizarrón y plumones.	4 horas
11 (Laboratorio)	Diseñar programas de computadora para la conversión de aceptadores de estado finito a gramáticas	Desarrollar en equipo un programa de computadora que reciba de entrada un aceptador de estado finito No determinístico	Apuntes, Computadora	8 horas

	regulares, identificando la relación que guardan en la solución de problemas de lenguajes formales, de manera creativa.	y lo convierta a gramática y viceversa.		
12 (Taller)	Diseñar aceptadores de estado finito a partir de expresiones regulares, con la solución de ejercicios, para identificar la relación que guardan en la solución de problemas de lenguajes formales, de manera creativa.	Diseñar aceptadores de estado finito a partir de una expresión regular.	Apuntes, papel, lápiz, pizarrón y plumones.	4 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 consiste básicamente en asignar problemas para la cual deberán organizarse y resolverlos.

- 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.

- Prácticas de laboratorio

Se solicitarán prácticas de laboratorio de acuerdo a los temas que se vayan abordando durante el periodo. El trabajo de las prácticas se entrega de forma individual, pero pueden trabajarse por equipo. Se requerirá que los estudiantes realicen horas extraclase para la resolución de los ejercicios y prácticas.

VIII. CRITERIOS DE EVALUACIÓN

Criterio de calificación

■ 4 Exámenes parciales	50%
■ Tareas	10%
■ Prácticas	40%
TOTAL	100%

Criterio de acreditación

- Resolver cuatro exámenes parciales en tiempo y forma.
- Cumplir con las prácticas de laboratorio en tiempo y forma.
- Para la acreditación del curso se atenderá al Estatuto Escolar Vigente, artículos 70-71, por lo que el estudiante deberá contar un mínimo de 80% de asistencias en el periodo. Tener un mínimo aprobatorio de 60 en su calificación final.

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.

IX. BIBLIOGRAFÍA

Básica

- Denning, P. J., Dennis, J. B., & Qualitz, J. E. (1978). *Machines, languages and computation*. Prentice Hall Professional Technical Reference. (CLÁSICO)
- Hopcroft, J. E., Ullman, J. D., & Motwani, R. (2007). *Introducción a la teoría de autómatas, lenguajes y computación* 3ra ed. (CLÁSICO)
- Viso Gurovich, Elisa (2010). *Introducción a la teoría de la computación: Autómatas y lenguajes Formales*. UNAM. Fac. de Ciencias.

Complementaria

- Jurado Málaga, Elena (2008). *Teoría de autómatas y lenguajes formales*. Universidad de Extremadura. (CLÁSICO)
- Meduna, A. (2012). *Automata and languages: theory and applications*. Springer Science & Business Media.
- <http://www.jalc.de/>
- <http://www.journals.elsevier.com/theoretical-computer-science/>

X. PERFIL DEL DOCENTE

Profesionista del área de matemáticas o computación con conocimientos en el manejo de lenguajes y teoría de la computación y experiencia docente.