

NOMBRE DE LA INSTITUCIÓN	Universidad de Sonora
DIVISIÓN ACADÉMICA	División Ciencias Exactas y Naturales
DEPARTAMENTO QUE IMPARTE LA MATERIA	Departamento de Matemáticas
LICENCIATURAS USUARIAS	Ciencias de la Computación
NOMBRE DE LA MATERIA	Teoría de la Computación
CLAVE	9446
EJE FORMATIVO	Profesional
REQUISITOS	Matemáticas Discretas
CARÁCTER	Obligatorio
VALOR EN CRÉDITOS	8 (3 teoría/2 taller)
CLAVE	

Introducción:

En el diseño lógico de sistemas digitales, codificadores y robots, pasando por el funcionamiento de autómatas empleados en la industria y en las comunicaciones, hasta las máquinas inteligentes y la inteligencia artificial; actúa toda una maquinaria lógico-matemática: gramáticas formales, modelos matemáticos, teoremas y conceptos teóricos específicos. En este curso se estudiará de manera precisa la teoría de autómatas y sus lenguajes formales.

Objetivo general:

Conozca los elementos principales en que se sustenta la teoría de la computación y sea capaz de comprender y aplicar herramientas empleadas en diseño de lenguajes de programación

Objetivos específicos:

Introducir al estudiante a la teoría de computabilidad, resultados importantes del estudio de autómatas y lenguajes formales.

Discutir nociones de decidibilidad e indecidibilidad y la relación con el análisis de complejidad.

Presentar y analizar los enfoques a la semántica de programas, llevando esto a una breve introducción al tema de verificación formal de programas.

Contenido

1. Introducción.

- 1.1. Procesadores de lenguajes.
- 1.2. Cadenas, alfabetos y lenguajes.
- 1.3. Modelos matemáticos de traducción.

2. Máquinas con un número finito de estados.

- 2.1. Definición.
- 2.2. Aplicación e instrumentación. Diseño de autómatas finitos.
- 2.3. Equivalencia de autómatas finitos.
- 2.4. Minimización de autómatas finitos.
- 2.5. Relación entre modelos de autómatas finitos: determinísticos, no determinísticos, de Mealy y de Moore.

3. Gramáticas formales y lenguajes formales.

- 3.1. Introducción.
- 3.2. Conceptos básicos de gramática.
- 3.3. Gramáticas formales.
- 3.4. Clasificación de gramáticas: no restringidas, sensitivas al contexto, libres del contexto y regulares.
- 3.5. Árboles de derivación. Ambigüedad.

4. Autómatas de estados finitos y lenguajes tipo 3.

- 4.1. Introducción.
- 4.2. Propiedades de los lenguajes regulares.
- 4.3. Expresiones regulares. Equivalencia entre autómatas finitos y expresiones regulares.

- 4.4. Algoritmos de decisión para lenguajes regulares: Ambigüedad; vacuidad, finitud e infinitud; equivalencia.
- 4.5. Aplicaciones de autómatas finitos y lenguajes regulares.
- 5. Autómatas con snack o pila.**
 - 5.1. Definición.
 - 5.2. Formalización
 - 5.3. Traducción con autómatas de stack.
 - 5.4. Ciclos en los autómatas de stack determinísticos
- 6. Lenguajes libres del Contexto**
 - 6.1. Presentación.
 - 6.2. Simplificación de gramáticas libres del contexto. Producciones vacías o Producciones-E.
 - 6.3. Formas normales: Chomsky y Greibach.
 - 6.4. Equivalencia entre autómatas de stack y lenguajes libres del contexto.
- 7. Propiedades de los lenguajes libres del contexto.**
 - 7.1. Caracterización del complemento de los lenguajes libres del contexto: Lemas del bombeo y de Orden.
 - 7.2. Propiedades de cerradura de los lenguajes libres del contexto: Operaciones. booleanas Utilización de las propiedades de cerradura.
 - 7.3. Deducibilidad en lenguajes libres del contexto, membresía
- 8. Máquina de Turing.**
 - 8.1. Motivación, definiciones y notación.
 - 8.2. Técnicas para la construcción de máquinas de Turing.
 - 8.3. La Máquina de Turing como un procedimiento.
 - 8.4. Distintos tipos de Máquina de Turing.
 - 8.5. Computabilidad e Indecibilidad.
 - 8.6. Modelos de funciones computables a través de máquinas de Turing, funciones recursivas (parciales), cálculo lambda, lenguajes imperativos; tesis de church
 - 8.7. Máquinas universales (Máquina de Turing).
 - 8.8. Problemas de decisión; problemas recursivos y recursivamente enúmeros.
 - 8.9. Problemas indecibles como por ejemplo, El problema de paro (halting problem).
- 9. Semántica de los lenguajes de programación.**
 - 9.1. Semántica informal.
 - 9.2. Semántica formal (axiomática, denotacional, operativa).

Estrategias Didácticas

- En general, promover la participación activa de los estudiantes poniendo especial atención al desarrollo de habilidades de comunicación oral y escrita.
- Presentar demostraciones formales de algunos resultados relacionados con los temas del curso.
- Incorpore el empleo de recursos computaciones, en las actividades cotidianas del curso.

Estrategias de Evaluación

Para la evaluación de los estudiantes, el profesor tomará en cuenta:

- Resultados de los exámenes parciales aplicados (se sugiere que sean al menos tres),
- Tareas, trabajos de investigación, presentaciones.
- Participación individual y colectiva en las actividades cotidianas.

Los porcentajes serán acordados al inicio del semestre.

Bibliografía

- Linz, P. *An Introduction To Formal Languages And Automata*. D.C. Heath and Company, 1990.
- Ullamn, J.D. y Hopcroft, J. E. *Introduction to Automata Theory, Language and Computations*. Addison Wesley Publishing Company, 1979.
- Gough, K. J. *Syntax Análisis and Tools*. Addison-Wesley Publishing Company, 1988.
- Gries, D.; Editor. *Programming In The 1990, An Introduction To Calculation Of Programs*. Springer Verlag, 1990.
- Dijkstra, E. W.; editor. *Formal Development Of Programs And Proofs*. Addison-Wesley Publishing Company, 1990.

Manna, Z. & Waldinger, R. *The Logical Bases For Computing Programming*. Addison Wesley Publishing Company, 1993.

Kfoury, A. J.; Moll, R. N. & Arbib, M. A. *A Programming Approach Computability*. Springer-Verlag, 1982.

Perfil Académico Deseable del Maestro

Se recomienda que el profesor tenga las siguientes características:

- Formación matemática sólida en el área de Ciencias de la Computación o área afín de forma que sea capaz de dar un panorama general del uso de los temas expuestos en clase en las distintas áreas de las Ciencias de la Computación.