

Especialista en contenido:	PROF. ALBERTO CASTILLO	AUTORIZADO POR VICE RECTORADO ACADÉMICO (FIRMA Y SELLO)
Fecha de elaboración:	MARZO, 1993	
Elaborado por:	PROF. ALBERTO CASTILLO	

DESCRIPCIÓN DEL PROGRAMA

Las ciencias e ingeniería teórica de la computación tienen sus raíces en áreas tales como la lógica formal, computacional y la parte de la lógica conocida como teoría de funciones recursivas. Preguntas acerca de la computabilidad y decibilidad se centran en las limitaciones de las técnicas matemáticas formales que antecedieron la construcción de los computadores. El desarrollo de diseños de sistemas computacionales complejos y aplicaciones resultantes estimularon el estudio de las limitaciones de procesadores y algoritmos; tres aspectos importantes de estas teorías se estudian en este curso: los que tienen que ver con limitaciones de software (teoría de los lenguajes formales) y sobre la teoría de la complejidad que tiene que ver con la posibilidad de tratar o no un problema para alcanzar su solución con los medios y recursos de procesadores, tiempo y tipo de algoritmos disponibles para su solución.

MODALIDAD Y ESTRATEGIA DE ENSEÑANZA

Es la misma que la empleada en la enseñanza de las matemáticas o del curso de lógica computacional que le precede y es pre-requisito: definición de términos y conceptos, proposiciones básicas, demostraciones de nuevas proposiciones que son teoremas. Se hará énfasis entre la teoría y sus aplicaciones.

El material sobre aplicaciones será presentado después de cada una de las tres áreas que componen el curso: Teoría de Autómata, Teoría de los Lenguajes Formales, y Teoría de la Complejidad.

ESTRATEGIA DE ESTUDIO

El estudiante deberá repetir las demostraciones y resolver nuevas demostraciones de problemas de igual complejidad a los dados en clase, luego hará ejercicios y resolverá más problemas sobre los aspectos de aplicaciones será presentado después de cada una de las tres áreas que componen el curso: Teoría de los lenguajes formales, y Teoría de la complejidad.

OBJETIVO TERMINAL DE LA ENSEÑANZA

Al finalizar el curso el estudiante estará en capacidad de conocer los fundamentos matemáticos de la computabilidad, las limitaciones de procesadores y algoritmos, los elementos de autómatas lenguajes formales y complejidad, que dan base para el diseño de compiladores y lenguaje de programación y ponen límites a lo que puede o no resolverse con un computador.

UNIDAD I	OBJETIVO TERMINAL DE UNIDAD
AUTÓMATA DE ESTADOS FINITOS	PARA INTRODUCIR AL ESTUDIANTE A LOS CONCEPTOS BÁSICOS DE AUTÓMATAS, COMENZANDO CON LA REPRESENTACIÓN Y PROPIEDADES DE LOS AUTÓMATAS DE ESTADOS FINITOS.
DURACIÓN	
3 SEMANAS	
EVALUACIÓN	
20%	
ESTRATEGIA DE INSTRUCCIÓN	
Exposición tutorial; prácticas y examen al finalizar la unidad	
OBJETIVOS ESPECÍFICOS	
- Se estudiarán autómatas determinísticos y no determinísticos y sus representaciones gráficos y tablas de transición. - Se estudiarán los conjuntos y expresiones regulares que los sustentan y teoremas más importantes.	
CONTENIDOS	
- Expresiones Regulares – Autómatas Finitos – Grafos de Transición - Teoremas de Kleene. El Teorema de Equivalencia.	

UNIDAD II	OBJETIVO TERMINAL DE UNIDAD
MÁQUINAS DE TURING	EL ESTUDIO DE LAS PROPIEDADES DE LAS MÁQUINAS DE TURING Y DE LAS FUNCIONES RECURSIVAS.
DURACIÓN	
5 SEMANAS	
EVALUACIÓN	
30%	
ESTRATEGIA DE INSTRUCCIÓN	
Se presentarán formalmente las máquinas de Turing y su relación con la compatibilidad de funciones.	
OBJETIVOS ESPECÍFICOS	
1. Conceptos sobre máquinas de Turing y los problemas de la limitación. El problema de Parada y la Undecidibilidad de ciertos predicados. La relación de algoritmos y máquinas de Turing. 2. Las funciones recursivas totales, regulares y parciales y su lenguaje de representación. La relación de algoritmos y funciones recursivas. 3. Funciones computables, números de Godel y la prueba de Godel y la tesis de Churc Turing.	
CONTENIDOS	
• Máquinas de Turing. El problema de Parada y la Indecibilidad. Problemas de Decisión y ejemplos. Descripciones Instantáneas. Uso del lenguaje para representar funciones recursivas. • Recursión Primitiva y ejemplos. • Funciones recursivas totales, regulares y parciales. • Minimización sin límites. Funciones recursivas y funciones recursivas parciales. Numeración de godel. Funciones Computables: Computabilidad de Turing. • Equivalencia entre funciones computables vía Turing y vía funciones recursivas. Tesis de church.	

UNIDAD III	OBJETIVO TERMINAL DE UNIDAD
LENGUAJES FORMALES	CUANDO TERMINE ESTA UNIDAD EL ESTUDIANTE CONOCERÁ EL TIPO DE REPRESENTACIÓN Y LAS PROPIEDADES DE LOS LENGUAJES DE GRAMÁTICAS LIBRES DE CONTEXTO Y SU RELACIÓN CON LOS AUTÓMATAS DE “PUSHDOWN”
DURACIÓN	
5 SEMANAS	
EVALUACIÓN	
30%	
ESTRATEGIA DE INSTRUCCIÓN	
Se establecerá formalmente la definición y operadores de una gramática de contexto libre y se asociaran sus teoremas con los de autómatas pushdown.	
OBJETIVOS ESPECÍFICOS	
- Proveer las características y representación de gramáticas libres de contexto tales como los sistemas de producción, la forma normal de Chomsky, la de Greibach y otras. - Relacionar éstas con autómatas de Pushdown.	
CONTENIDOS	
- Gramáticas libres de contexto. - Sistemas de Producción. Formal Normal de Chomsky. Formal Normal de Greibach. P autocontenidas. - Autómatas Pushdown-Autómatas no determinísticas y gramáticas de contexto libre. Técnicas de Parsing. Requerimientos de espacio y tiempo. Lenguajes sensitivos al contexto. Autómatas limitaas linealmentw y su relación con los conjuntos recursivos. Propiedades de cerradura.	

UNIDAD IV	OBJETIVO TERMINAL DE UNIDAD
INTRODUCCIÓN A LA TEORÍA DE LA COMPLEJIDAD	ESTUDIAR LOS FACTORES QUE AFECTAN EL DISEÑO DE LOS ALGORITMOS Y LOS LÍMITES DE LA COMPUTACIÓN.
DURACIÓN	
3 SEMANAS	
EVALUACIÓN	
20%	
ESTRATEGIA DE INSTRUCCIÓN	
Se establecerán los teoremas de limitación en relación al diseño en tiempo y memoria de ejecución de algoritmos.	
OBJETIVOS ESPECÍFICOS	
1. Establecer conceptos y propiedades de la complejidad en tiempo. 2. En espacio. 3. En clases.	
CONTENIDOS	
• Complejidad en tiempo-Complejidad en clase	

BIBLIOGRAFÍA

Both T.L. (1967). Sequential Machines y Autómatas Theory

Brainerd J.R. (1974). Theory of Computation. Wiley.

Hopcroft, J.L. and Ullman J.D. (1979). Introduccion to Autómata Theory, Language and Computation. Nuevas Ediciones Wiley.

Otras Ediciones (1979). Machines Language and Computation. Prentice-Hall.

Zohar Manna (1989). Matematical Theory of Computation. Mc Graw-Hill