

CARRERA	ASIGNATURA	Año	Régimen ¹	Plan	Total Horas
Ingeniería en Sistemas	Teoría de la Computación	4°	Anual	071/08	60

EQUIPO DOCENTE:

PROFESOR	CATEGORÍA
	Titular
	Asociado
José Daniel Texier Ramírez	Adjunto
	Jefe de Trabajos Prácticos
	Ayudante de 1 ^{ra}
	Ayudante de 2 ^{da}

1. CONTENIDOS MÍNIMOS²:

Lenguajes formales y autómatas. Minimización de autómatas. Lenguaje y Gramática. Clasificación de los lenguajes. Lenguajes y gramáticas regulares. Expresiones regulares. Lenguaje y gramáticas libres de contexto. Autómatas aceptores de lenguaje. Lenguajes estructurados en frases. Jerarquía de Chomsky. Máquina de Turing. Gramáticas e Isomorfismos. Teoría de Computabilidad y Complejidad. Problemas computables y no computables. Problemas de la parada. Problemas tratables e intratables. Balance entre Tiempo y Espacio en los Algoritmos. Análisis de Complejidad de Algoritmos. Funciones Recursivas. Verificación de programas. Semántica axiomática. Semántica operacional. Semántica denotacional. Compiladores.

2. CONTENIDOS DE LA ASIGNATURA³:

Unidad N° 1: TEORÍA DE AUTÓMATAS

Contenidos:

Objetivo: Al finalizar la unidad, el alumno definirá, construirá y analizará expresiones regulares y lenguajes regulares. Construirá autómatas finitas deterministas y no deterministas; podrá comprobar si un lenguaje dado es o no regular.

- Alfabetos.
- Lenguajes.
- Operaciones con cadenas.
- Operaciones con lenguajes
- Expresiones regulares
- Autómatas finitas.

¹ Anual, Primer Cuatrimestre ó Segundo Cuatrimestre

² Se deberán consignar los mismos, tal como se encuentran aprobados en el Plan de Estudios aprobado por Resolución Rectoral.

³ Cada Unidad Temática estará identificada por un nombre que describa claramente una unidad de conocimientos coherentes, la descripción de los mismos, la bibliografía específica para la misma (puede ser la misma en varias unidades o tener cada una de ellas diferencias con otras) y la manera en que serán evaluados esos contenidos.

- Grafos de transición
- Teorema de Kleene
- Equivalencia entre autómatas finitos y grafos de transición.
- Equivalencia entre grafos de transición y expresiones regulares.
- Equivalencia entre expresiones regulares y autómatas finitos
- Autómatas finitos deterministas.
- Autómatas finitos no-deterministas.
- Autómatas finitos con salida.
- Lenguajes regulares.

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura.

Unidad N° 2: TEORÍA DE AUTÓMATAS DE PILA

Contenidos:

Objetivo: Al finalizar la unidad, el alumno analizará, definirá y construirá gramáticas regulares, de contexto libre, autómatas de pila, aplicará la forma normal de Chomsky, demostrará la equivalencia entre CFG y PDA

- Autómata de pilas (PDA/AP).
- Autómata de pilas no deterministas (APND).
- Forma normal de Chomsky (CNF)
- Forma normal de Backus (BNF)
- Lenguajes independientes del contexto.
- Lenguajes de programación.
- Gramáticas de contexto libre (CFG)
- Gramáticas regulares.
- Derivación por derecha y por izquierda.
- Árboles.
- Ambigüedad.
- Equivalencia entre las gramáticas de contexto libre y los autómatas de pila.
- Sea $CFG = PDA$.

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura.

Unidad N° 3: MÁQUINAS DE TURING

Contenidos:

Objetivo: Al finalizar la unidad, el alumno analizará, definirá y construirá máquinas de Turing.

- Definición formal.
- Representación.
- Interpretaciones de las computaciones.
- Configuración de una máquina de Turing.

- Máquina de Turing multicinta.
- Máquina universal de Turing.
- Codificación de una máquina de Turing.
- Problema de parada (detención).

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura

Unidad N° 4: COMPILADORES

Contenidos:

Objetivo: Al finalizar la unidad, el alumno podrá definir, analizar y construir un traductor o un compilador.

- Traducción de lenguajes
- Fases de un traductor y agrupación de las fases.
- Lenguajes y gramáticas. Conceptos básicos.
- Función del analizador lexicográfico.
- Descripción y reconocimiento de los tokens.
- Construcción de un analizador léxico a mano.
- Autómatas finitos como reconocedores de lenguajes Regulares.
- Programa LEX (FLEX) para construcción de un analizador.
- Función del análisis sintáctico.
- Gramáticas libres de contexto.
- Ambigüedad.
- Tipos de análisis sintácticos.
- Construcción de analizadores sintácticos descendentes.
- Procedimientos de análisis LL(1).
- Construcción de analizadores sintácticos ascendentes.
- Procedimientos de análisis LR(1).
- El programa YACC generador de analizadores.
- Especificación natural.
- Tipos de semántica: operacional, axiomática y denotativa.
- Compatibilidad de tipos.
- Rutinas semánticas.
- Especificación formal.

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura

Unidad N° 5: TEORÍA DE LA COMPUTABILIDAD

Contenidos:

Objetivo: Al finalizar la unidad, el alumno podrá determinar que puede o no puede con una computadora.

- Conceptos básicos de Teoría de Computabilidad.
- Formalización, representación y solución de problemas.

- Procedimientos y algoritmos.
- Problemas computables y no computables.
- Enumerabilidad, decidibilidad y generabilidad.
- Función computable.

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura

Unidad N° 6: TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL

Contenidos:

Objetivo: Al finalizar la unidad, el alumno calcular la eficiencia de algoritmos y de esta manera, lograr diseñar algoritmos eficientes

- Análisis de Complejidad de Algoritmos.
- Análisis de Algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso.
- Orden de un algoritmo.
- Principios y consideraciones para la determinación del orden.
- Notación $O()$.
- Balance entre tiempo y espacio en los algoritmos.
- Clasificación de problemas:
- Problema de parada (detención).
- Problemas tratables e intratables.
- Clase P, NP, NP completa y CO-NP.

Bibliografía específica de la unidad:

Se emplea la bibliografía obligatoria en todas las unidades de la asignatura

3. BIBLIOGRAFÍA⁴:

Bibliografía básica:

- Daniel L. A. Cohen: Introduction to Computer Theory, John Wiley & Sons Inc. 1986.
- Dean Kelley: Teoría de Autómatas y Lenguajes Formales, Prentice Hall 1995.
- J. Glenn Brookshear: Teoría de la Computación, lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana 1993.
- Juan M. Cueva, Lenguajes, Gramáticas y Autómatas. Segunda Edición, Oviedo, 2001. (Libro electrónico). <https://dl.dropboxusercontent.com/u/23495713/TC/Cueva.pdf>
- Julio Ariel Hurtado Alegria, Raul Kantor, Carlos Luna, Luis Sierra, Dante Zanarini. Temas de Teoría de la Computación. Primera edición. Proyecto LATIn. 2014. (Libro electrónico). <https://dl.dropboxusercontent.com/u/23495713/TC/Kantor.pdf>

⁴ Se requiere consultar en la Biblioteca de la UNdeC la existencia de textos referidos a la temática de cada asignatura a fin de trabajar con material ya existente, en caso de no existir textos relacionados realizar la solicitud correspondiente.

- Linz P.: Introduction to Formal Languages and Automata, D. C. Heath and Company 1990.
- Serafín Moral, Teoría de Automatas y Lenguajes Formales. Granada, 2001. (Libro electrónico). <https://dl.dropboxusercontent.com/u/23495713/TC/Serafin.pdf>
- T. A. Sudkamp: Introduction to the theory of Computer Science, Addison-Wesley Publishing Company 1997.
- Ullman J. D.; Hopcroft, J. E.: Introduction to Automata Theory, Languages and Computations. Addison-Wesley Publishing Company 1979

Bibliografía complementaria:

- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman. Compiladores: Principios, técnicas y herramientas. Segunda edición. 2008.
- Dijkstra, E. W., editor., Formal Development Of Programs And Proofs, Addison-Wesley Publishing Company, 1990.
- Benjamin Bustos. Estudio y optimización del algoritmo de ordenamiento. Departamento de Ciencias de la Computación, Universidad de Chile. Disponible en: [www.dc-uchile.cl/~bebustos/files/Bus99.pdf](http://www.dc.uchile.cl/~bebustos/files/Bus99.pdf)
- Gurari, Eitan. An Introduction to the Theory of Computation. Ohio State University Computer Science Press, 1989, ISBN 0-7167-8182-4. Disponible en URL: <http://www.cse.ohio-state.edu/~gurari/theory-bk/theory-bk.html>
- Henry Ledgard, Michael Marcotty: The Programming Language Landscape, Science Research Associates Inc. 1981.
- John Martin, Introduction to Languages and the Theory of Computation (Third edition) McGraw Hill, 2003
- Kain, R. Automata Theory. Mc. Graw-Hill. 1972.
- Kenneth C. Louden. Construcción de compiladores: Principios y práctica. Cengage Learning Editores. 2004.
- Kfoury, A.J.; Moll, R. N.; Arbib, M. A., A Programming Approach to Computability, Springer-Verlag, 1982.
- Manna, Z.; Waldinger, R., The Logical Bases For Computing Programming, Addison-Wesley Publishing Company, 1993.
- Manuel Bermúdez: Guías sobre el proceso de compilación. Universidad de Florida. 1999.
- Manuel Bermúdez: Rpal, Un Lenguaje Funcional. Universidad de Florida. 2004.
- Rosa Guerequeta y Antonio Vallecillo. La complejidad de los algoritmos. Servicio de Publicaciones de la Universidad de Málaga. 1998 Disponible en: <http://www.lcc.uma.es/~av/Libro/CAP1.pdf>
- Sergio Gálvez Rojas y Miguel Mora. Java a Tope: Traductores y Compiladores con LexYacc JFlexCup y JavaCC. Universidad de Málaga, 2005.
- Sudkamp, T. Languages and Machines. Reading, Massachusetts. 1997.



4. OBSERVACIONES⁵:

--

CHILECITO, Provincia de La Rioja, 1 de marzo de 2017

Profesor Titular

RECUERDE ACOMPAÑAR ESTA PROPUESTA CON LAS GUÍAS DE LAS ACTIVIDADES PRÁCTICAS

⁵ Este documento será revisado anualmente.