



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area V: Automatas y Lenguajes

(Programa del año 2014)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DE COMPUTACION	ING. INFORM.	026/1 2	2014	1° cuatrimestre
FUNDAMENTOS DE COMPUTACION	ING. EN COMPUT.	28/12	2014	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ROGGERO, PATRICIA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
LEGUIZAMON, MARIO GUILLERMO	Prof. Co-Responsable	SEC F EX	4 Hs
APOLLONI, JAVIER MARIANO	Responsable de Práctico	JTP Exc	40 Hs
FUNEZ, DARIO GUSTAVO	Auxiliar de Práctico	A.1ra Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	3 Hs	1 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
12/03/2014	19/06/2014	15	90

IV - Fundamentación

El estudio de la teoría de lenguajes es de gran importancia para entender de manera formal las características de los lenguajes de programación desde el punto de vista de su generación (a través de dispositivos generadores) y su reconocimiento (a través de dispositivos reconocedores o autómatas). Asimismo, el estudio de los modelos formales de computación es necesario para el entendimiento de la idea intuitiva de algoritmo. A través de dichos modelos formales es posible el estudio de la teoría de computabilidad y complejidad, la cual nos permite dividir en clases de problemas no-decidibles y decidibles y dentro de estos últimos, establecer jerarquías de clases de complejidad, tanto temporal y espacial.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de comprender la forma en que funciona cada autómata y la correspondencia entre autómata, gramática y lenguajes, particularmente lenguajes de programación. Además, comprender los modelos avanzados de computación, como Máquina de Turing, a los efectos de aprender, con cierta profundidad, la teoría y práctica de computabilidad (lenguajes decidibles y no-decidibles) y complejidad computacional (problemas NP-completos y otros relacionados). Se espera también poner énfasis en la vinculación de ciertos aspectos de la teoría de lenguajes con aplicaciones prácticas.

VI - Contenidos

Bolilla 1.

Alfabeto. Sentencia. Lenguaje. Representación de los lenguajes. Dispositivos generadores y reconocedores. Concepto general de Gramáticas y Autómatas, como dispositivos generadores de lenguajes y reconocedores de lenguajes, respectivamente. Relación general entre autómatas y gramática. Jerarquía de Chomsky.

Bolilla 2.

Lenguajes Regulares (Tipo 3). Autómata Finito Determinístico y No-Determinístico. Equivalencia de aceptación de un AFD y un AFND. AFND con transiciones épsilon. Equivalencia de aceptación entre AFND y AFND-épsilon. Expresiones regulares y Gramáticas regulares (Tipo 3). Equivalencia entre: lenguajes regulares, lenguajes aceptados por autómatas finitos determinísticos y no-determinísticos y gramáticas regulares. Minimización de autómatas finitos determinísticos. Analizador lexicográfico o Scanner.

Bolilla 3.

Gramáticas y lenguajes libres del contexto (Tipo 2). Motivación e introducción. Árbol de derivación. Frontera. Forma sentencial izquierda y derecha. Ambigüedad. Autómatas a Pila (Push Down). Autómata Push-Down determinístico. Configuraciones. Lenguaje aceptado por Pila Vacía y Estado Final. Equivalencia de aceptación por Pila Vacía y Estado Final.

Bolilla 4.

Concepto de Algoritmo y Computabilidad. Conjuntos recursivos y recursivamente enumerables. Máquinas de Turing (MT). MTs como reconocedor. MT como computadora de funciones numéricas. Extensiones de MTs: No-Determinísticas, con k-cintas, otros modelos. Restricciones de MTs: Autómata Linealmente Acotado (ALA). Recursividad y lenguajes reconocidos por un ALA. Otros modelos de computación equivalentes a MTs: Funciones Recursivas y Sistemas de Reescritura (p.e, semi-Thue). Equivalencia de los Modelos Formales. Tesis de Church-Turing.

Bolilla 5.

Problemas decidibles y no-decidibles. Concepto de reducción de un problema a otro. Propiedades de los lenguajes recursivos y recursivamente enumerables. MT Universal. Problema de Detención (Halting). Problema de Correspondencia de Post (PCP). Problemas decidibles y no-decidibles para Lenguajes Tipo 0, 1, 2 y 3. Aplicación de PCP para problemas de lenguajes Tipo 2.

Bolilla 6.

Complejidad Computacional. Complejidad Temporal y Espacial. Notación O-grande. Jerarquías de complejidades temporales y espaciales. Relación entre complejidades temporales y espaciales. Otras notaciones de complejidad. Problemas tratables e intratables. Clases P y NP. NP-Complejidad. Problema de Satisfactibilidad (SAT). Teorema de Cook. Problemas NP-Completos (aplicación del concepto de reducción): 3-SAT, Clique, Cobertura de vértices, etc. Clases P-Space y P-Space Completo. Máquinas de Turing y Modelos Finitos. Definición de lógica de primer orden y lógica de segundo orden. Teorema de Fagin y la clase NP.

VII - Plan de Trabajos Prácticos

Las clases prácticas consistirán de la resolución de ejercicios en lápiz y papel, pero además el alumno deberá utilizar herramientas informáticas que implementan los conceptos teóricos desarrollados en la asignatura, de manera que pueda poner en práctica los conocimientos adquiridos.

Prácticos de Aula:

Práctico 1: Alfabetos - Cadenas - Lenguajes.

Hacer un análisis de la representación finita de lenguajes, en particular de lenguajes de programación. La metodología a utilizar es la resolución de ejercicios en lápiz y papel.

Práctico 2: Lenguajes Regulares, Autómatas Finitos y Gramáticas Regulares. Construir autómatas, expresiones regulares y gramáticas regulares. Proponer utilizando autómatas finitos modelos de problemas de la vida real.

La metodología a utilizar es la resolución de ejercicios en lápiz y papel y la corroboración de lo realizado manualmente utilizando la herramienta Chalchalo y/o JFLAP.

Práctico 3: Equivalencias entre Autómatas Finitos, Expresiones Regulares y Gramáticas Regulares. Minimización.

Aplicación de algoritmos que muestran las equivalencias y uso de algoritmos de minimización. Resolución de ejercicios en lápiz y papel. Corroboración de lo realizado manualmente utilizando herramientas de software. Chalchalero y/o JFLAP.

Práctico 4: Gramáticas Libres del Contexto - Automatas Push-Down.

Resolución de ejercicios en lápiz y papel, básicamente relacionados a lenguajes de programación, utilizando GLC, BNF y APD. Corroboración de los ejercicios realizados a mano utilizando la herramienta JFLAP.

Práctico 5: MTs, Extensiones de MTs, ALA. Modelos equivalentes a MT. Se pretende plantear ejercicios que muestren las diferentes formas de resolverlos mediante los distintos modelos de computación.

Práctico 6: Problemas decidibles y no-decidibles. Aplicación del concepto de reducción de un lenguaje a otro para determinar la decidibilidad de problemas relevantes en la teoría de la computación. También se consideran aquí problemas conocidos de la teoría de lenguajes. MTs es el modelo computacional sobre el cual estarán basadas las reducciones a aplicar.

Práctico 7: Complejidad Computacional y NP-Complejidad. Similar al práctico 6, pero en este caso, se incluirán varios problemas de la vida real para determinar la complejidad inherente de los mismos a través del uso del concepto de reducción. La primera parte del práctico incluirá problemas de índole general y teórico, mientras que la segunda parte estará orientada a promover la investigación del alumno acerca de la complejidad de ciertos problemas de la vida real y para los cuales se pueda establecer similitudes en cuanto su complejidad inherente respecto a problemas vistos en la primera parte del práctico o en teoría.

Prácticos de Laboratorio:

1- Implementación de un Analizador Lexicográfico a través del uso de herramientas automáticas (JFlex). El alumno deberá entregar el código correspondiente y un informe escrito que describa la solución planteada, los detalles de implementación, etc.

2. Uso de un simulador ad hoc de Máquinas de Turing para validar el diseño de las MTs realizados en prácticos de aula. Esta actividad permitirá realizar una auto-evaluación por parte del alumno y comprender más profundamente el comportamiento de este tipo de modelo computacional. El simulador puede ser cualquiera de los muchos disponibles.

VIII - Regimen de Aprobación

El alumno podrá cursar la materia bajo régimen regular.

A. Condiciones para regularizar:

1. Asistencia al 60% de las clases prácticas.
2. Entrega del 70% de los ejercicios de prácticos de aula solicitados.
3. Presentación y aprobación de los prácticos de máquina propuestos (fuente) y todo el material que sea solicitado.
4. Aprobar 2 exámenes parciales (prácticos) o sus respectivas recuperaciones, o, para quienes corresponda, la recuperación por trabajo,

Setenta por ciento (70%) es el porcentaje mínimo, de los ejercicios ha resolver, necesario para aprobar los parciales o recuperaciones.

B. El curso no admiten rendir examen final en condición de Libre.

C. El examen final puede ser oral y/o escrito.

IX - Bibliografía Básica

[1] - Hopcroft J.; Ullman J.; Motwani R. – “Introduction to automata theory, languages and computation”. Addison Wesley. 3° Edición. (2007).

[2] - Sudkamp, T.A. - “Languages and Machines (An Introduction to the Theory of Computer Science)”. Addison Wesley. 3° Edición. (2005).

[3] - Kozen, D. – “Theory of computation (Texts in Computer Science)”. Springer. 1° Edición (2006).

[4] - Sipser, M. – “Introduction to the Theory of Computation”. PWS Publishing Company. 2° Edición. (2005).

[5] - Martfn-Vide C., Paun, G. y Mitrana, V. (Editores), Formal Languages and Applications (Studies in Fuzziness and Soft Computing). Springer (2004).

[6] - Libkin, L. – “Elements of Finite Model Theory.” Springer (2004).

[7] - Ramón Brena Pinero "Lenguajes Formales y Automatas" (1997).

[8] - Apunte realizado por la Cátedra de "Análisis Lexicográfico - JFlex".

[9] - Notas de Clase de la Cátedra.

X - Bibliografía Complementaria

[1] - Wood, D. – “Theory of computation”. John Wiley & Sons, Inc. (1988).

[2] - Hopcroft J. y Ullman J.- “Introduction to automata theory, languages and computation”. Addison Wesley (1979).

XI - Resumen de Objetivos

Al finalizar el curso se espera que el alumno sea capaz de comprender la forma en que funcionan los diferentes tipos de autómatas y la correspondencia entre cada tipo, gramática y lenguajes, particularmente en lo referido a lenguajes de programación. Además, comprender los modelos avanzados de computación, como Máquina de Turing, a los efectos de aprender, con cierta profundidad, la teoría y práctica de computabilidad (lenguajes decidibles y no-decidibles) y complejidad computacional (problemas NP-completos y otros relacionados).

XII - Resumen del Programa

- Sentencias. Lenguajes. Representación de Lenguajes.

- Lenguajes Regulares. Autómatas Finitos. Expresiones Regulares. Gramáticas Regulares. Minimización. Analizador Lexicográfico.

- Gramáticas Libres del Contexto. Autómatas Push-Down.

- Máquinas de Turing. Extensiones de Máquinas de Turing. Autómatas Linealmente Acotados. Modelos equivalentes a Máquinas de Turing.

- Problemas decidibles y no-decidibles.

- Complejidad Computacional y NP-Compleitud. Clases de Complejidad Temporal y Espacial.

XIII - Imprevistos

XIV - Otros