



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 2010)
 (Programa en trámite de aprobación)
 (Presentado el 30/08/2010 10:44:45)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
COMPUTABILIDAD Y COMPLEJIDAD	LIC.EN CS.DE LA COMPUTACION	006/0 5	2010	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
LEGUIZAMON, MARIO GUILLERMO	Prof. Responsable	P.Adj Exc	40 Hs
HERRERA, NORMA EDITH	Prof. Colaborador	P.Adj Exc	40 Hs
APOLLONI, JAVIER MARIANO	Responsable de Práctico	JTP 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	4 Hs	Hs	6 Hs

Tipificación	Periodo
C - Teoria con prácticas de aula	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
09/08/2010	19/11/2010	15	90

IV - Fundamentación

El presente curso está destinado a alumnos avanzados de la Lic. en Ciencias de la Computación. Se profundiza el estudio de los modelos formales relacionados con la teoría de la computación iniciados en la materia Automatas y Lenguajes y que son de gran importancia para el entendimiento de la idea intuitiva de algoritmo.

El alumno debe aprender y entender la forma en la que funcionan cada uno de dichos modelos formales para que a través de ellos sea posible el estudio de la teoría de computabilidad y complejidad. Y a través de dicha teoría, el alumno podrá comprender la división de la clases de problemas en decidibles y no decidibles y dentro de éstos, establecer jerarquías de clases de complejidad.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de entender el funcionamiento de los modelos formales de la Teoría de la Computación, que extienden el poder computacional de aquellos modelos más limitados estudiados en la materia Automatas y Lenguajes. El modelo principal de computación usado aquí es la Máquina de Turing y alguna de sus variantes. Además se espera que el alumno pueda comprender y dominar adecuadamente los fundamentos de la Ciencia de la Computación, en cuanto a lo que puede y no puede ser computado (computabilidad), y también el costo en tiempo y/o espacio (complejidad) de aquellos lenguajes (problemas) que pueden ser decididos (resueltos).

VI - Contenidos

Bolilla 1.

Algoritmo y Computabilidad. Conjuntos recursivos y recursivamente enumerables. Máquina de Turing. Máquina de Turing como reconocedor. Máquina de Turing como computadora de funciones enteras. Máquina de Turing No Determinística. MT con k-Cintas. Restricciones sobre una MT: Autómata Linealmente Acotado (ALA). Recursividad y lenguajes reconocidos por un ALA. Otros modelos de computación. Sistema de Post. Sistema de Post como generalización del concepto de gramática. Funciones recursivas primitivas y mu-recursivas. Capacidad de los lenguajes de programación. Equivalencia de los Modelos Formales. Tesis de Church.

Bolilla 2.

Problemas decidibles y no decidibles. Concepto de Reducción de un problema a otro. Propiedades de los lenguajes recursivos y recursivamente enumerables. Lenguajes no-recursivamente enumerables. Máquina de Turing Universal. Problema de Halting. Problema de correspondencia de Post. La no decidibilidad del problema de correspondencia de Post. Problemas decidibles y no decidibles para lenguajes de tipo 0, 1, 2 y 3. Aplicación del problema de correspondencia de Post para problemas de decisión vinculados a gramáticas libres del contexto.

Bolilla 3.

Introducción al concepto de complejidad computacional. Complejidad temporal y espacial. Medida de complejidad temporal. Notación O-grande. Jerarquía de algunas complejidades temporales. Introducción a la complejidad espacial. Clases de problemas según su complejidad espacial. Relación entre las complejidades temporales y espaciales. Otras notaciones de complejidad. Análisis de Algoritmos.

Bolilla 4.

Problemas de decisión tratables e intratables. Clases P y NP. NP-completitud. Problema de Satisfactibilidad - Teorema de Cook. Problemas NP-completos y aplicación del concepto de reducción: 3-SAT, Clique, Cobertura de vértices, etc.

VII - Plan de Trabajos Prácticos

Práctico de Aula 1:

Máquina de Turing. Extensiones de Máquina de Turing. Autómata Linealmente Acotado.

Práctico de Aula 2:

Funciones Recursivas y Sistemas de Post.

Práctico de Aula 3:

Problemas Decidibles y No Decidibles.

Práctico de Aula 4:

Complejidad Computacional.

Práctico de Aula 5:

Problemas NP-Completo.

VIII - Regimen de Aprobación

Se admiten alumnos que podrán cursar la materia bajo régimen de aprobación con examen final o por promoción sin examen final.

A. Régimen para la Regularidad:

1. Tener un mínimo de 60% de asistencia a las clases prácticas.
2. Entrega del 70% de los ejercicios de prácticos de aula solicitados.
3. Aprobar 1 (uno) examen parcial práctico o su respectiva recuperación, o para quienes corresponda, la recuperación por

trabajo (ver nota).

4. Si cualquier punto no fuese cumplimentado, implicará que el alumno pase a condición de libre.

Nota: El porcentaje mínimo de los ejercicios a resolver necesarios para aprobar el examen parcial es de 70%. Además al menos el 50% de cada uno de los ejercicios involucrados en el parcial deberán ser completados para considerar su aprobación.

B. Régimen para Promoción sin Examen Final:

1. Idem a lo requerido para el régimen de regularidad, salvo que:

- a. El alumno deberá asistir a un 80% de las clases teóricas y prácticas.
- b. El examen parcial deberá ser aprobado con un porcentaje superior al 80% del total.

2. Aprobar con una nota mínima de 7 puntos un examen integrador oral y/o escrito al final del cuatrimestre.

3. La nota final se computará promediando las notas obtenidas en cada uno de los puntos mencionados previamente.

C. Régimen para rendir examen en condición de alumnos Libres:

1. Realizar un examen escrito de la parte práctica.

2. Habiendo aprobado el examen anterior, pasará a un examen oral o escrito, el cual tendrá las mismas características de un examen final para alumnos regulares.

IX - Bibliografía Básica

[1] J.E. Hopcroft, J. D. Ullman and R. Motwani. "Introduction to automata theory, languages and computation". Tercera Edición. Addison Wesley Publishing Company (2006).

[2] J.E. Hopcroft and J. D. Ullman. "Introduction to automata theory, languages and computation". Addison Wesley Publishing Company (1979).

[3] M. Sipser. "Introduction to the Theory of computation". PWS Publishing Company (1997).

[4] P. J. Denning, J. B. Dennis and J. E. Qualitz. "Machine, languages and computation". Prentice-Hall (1978).

[5] J. Hopcroft and J. Ullman. "Formal languages and their relation to automata". Addison Wesley Publishing Company (1969).

[6] D. Wood. "Theory of computation". John Wiley & Sons, Inc. (1986)

[7] T. A. Sudkamp. "Languages and Machines: An Introduction to the Theory of Computer Science". Segunda Edición. Addison Wesley. (1997)

[8] C. H. Papadimitriou. "Computational Complexity". Addison Wesley Publishing Company (1994).

[9] C. H. Papadimitriou and K. Steiglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall (1998)

X - Bibliografía Complementaria

[1] M. D. Davis and E. J. Weyuker. "Computability, Complexity, and Languages: Fundamentals of Computer Science". Academic Press Inc (1993).

[2] H. R. Lewis and C. H. Papadimitriou. "Elements of the theory of computation". Segunda Edición. Prentice Hall (1998).

[3] C. Teuscher. "Alan Turing: Life and Legacy of a Great Thinker". Springer Verlag (2004).

[4] A. Hodges. "Turing". Routledge (1999)

XI - Resumen de Objetivos

El objetivo principal de este curso es profundizar en los modelos formales de la Teoría de la Computación iniciados en la materia Autómatas y Lenguajes, y lograr un dominio de los fundamentos de la Ciencias de la Computación. De manera que al

finalizar el curso, el alumno podrá ser capaz de:

- lograr un entendimiento de la idea intuitiva de algoritmo utilizando los modelos formales, principalmente la Máquina de Turing,
- determinar aquellos lenguajes (problemas) que pueden y no ser decididos (resueltos) dentro de la Teoría de la Computabilidad,
- utilizar el costo en tiempo y/o espacio para establecer dentro de la Teoría de la Complejidad una jerarquía de clases de complejidad de los lenguajes decidibles,
- determinar cuáles problemas de decisión son tratables y cuáles intratables.

XII - Resumen del Programa

Bolilla 1.

Repaso: Algoritmo y Computabilidad. Modelos formales de computación. Tesis de Church. Máquina de Turing (extensiones y restricciones). Sistema de Post. Funciones recursivas.

Bolilla 2.

Problemas decidibles y no decidibles. Concepto de Reducción. Máquina de Turing Universal. Problema de Halting. Problema de correspondencia de Post. Aplicación del problema de correspondencia de Post para gramáticas libres del contexto.

Bolilla 3.

Complejidad temporal y espacial. Medida de complejidad temporal. Notación O-grande. Jerarquía de algunas complejidades temporales. Clases de problemas según su complejidad espacial. Otras notaciones de complejidad. Análisis de Algoritmos.

Bolilla 4.

Problemas de decisión tratables e intratables. Clases P y NP. NP-completitud. Problemas NP-Complejos. Aplicación del concepto de reducción.

XIII - Imprevistos

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	