Ministerio de Cultura y Educación **(Programa del año 2016)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area V: Automatas y Lenguajes

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DE COMPUTACION	ING. INFORM.	026/12-08/15	2016	1° cuatrimestre
FUNDAMENTOS DE COMPUTACION	ING. EN COMPUT.	28/12	2016	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ROGGERO, PATRICIA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
APOLLONI, JAVIER MARIANO	Prof. Colaborador	P.Adj Exc	40 Hs
LEGUIZAMON, MARIO GUILLERMO	Prof. Co-Responsable	SEC F EX	6 Hs
LOOR, FABRICIO	Auxiliar de Práctico	A.1ra Simp	10 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	1º Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
14/03/2016	24/06/2016	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 autómata y gramática. Jerarquía de Chomsky.

Bolilla 2.
Lenguajes Regulares (Tipo 3). Autómata Finito Determinístico (AFD) y No-Determinístico (AFND), AFND con transiciones épsilon, Aplicaciones. Expresiones regulares (ER). Gramáticas regulares (GR). Equivalencia entre: AFD - AFND, AFD - AFND epsilon, ER - AFND epsilon, AFD - GR. Minimización de AFD. Propiedades de clausura de los Lenguajes Regulares.

Bolilla 3.
Gramáticas y lenguajes libres del contexto (Tipo 2). Motivación e introducción. Árbol de derivación. Frontera. Forma sentencial izquierda y derecha. BNF. 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-Completitud. 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 JFLAP o similar.
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 JFLAP o similar.

Práctico 4: Gramáticas Libres del Contexto - Autómatas 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 o similar.
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-Completitud. 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 primer 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.

Práctico de Laboratorio:
Desarrollar una herramienta de Búsqueda de Patrones (Pattern Matching) sobre archivos de texto valiéndose del lenguaje de programación JAVA y las clases para el manejo de expresiones regulares y de interfaces gráficas. La herramienta tiene por objeto que el alumno se familiarice con el uso y la sintaxis de las expresiones regulares en el ámbito de los lenguajes de programación. Al finalizar el desarrollo, el alumno debe entregar la solución correctamente funcionando, junto con los detalles de implementación y códigos fuentes correspondientes.

VIII - Regimen de Aprobación

El alumno puede regularizar (luego rendir el examen final) o promocionar, las condiciones son:

A. Régimen para Alumnos Regulares

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. Aprobación del práctico de laboratorio propuesto.

4. Aprobar 2 exámenes parciales prácticos o alguna de sus respectivas recuperaciones. Tal como lo establece la reglamentación vigente, Ord 32/14 CS, que modifica el Régimen de Regularización de Materias establecido según Ord. 13/03 CS, se otorgan dos (2) recuperaciones por cada parcial.

Nota:
1. Setenta por ciento (70%) es el porcentaje mínimo, de los ejercicios a resolver, necesario para aprobar el parcial. Además al menos el 50% de cada uno de los ejercicios involucrados en el parcial deberán ser completados para considerar su aprobación.
2. Si cualquier punto no fuera cumplimentado implicará que el alumno pase a condición de libre.

B. Régimen para Alumnos Promocionales

1. Idem a lo requerido para alumnos regulares, salvo que:
a. El alumno deberá asistir a un 80% de las clases teóricas y prácticas.

2. Aprobar, con un mínimo de 7 (siete), un examen integrador oral y/o escrito al final del cuatrimestre.

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

C. El curso no admite rendir el examen final en condición de Libre.
D. 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 Autómatas" (1997).
[8] - Apunte realizado por la Cátedra de "Análisis Lexicográfico - JFlex".
[9] - Notas de Clase de la Cátedra.

X - Bibliografia 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-Completitud. Clases de Complejidad Temporal y Espacial.

XIII - Imprevistos

Eventualmente si por motivos de fuerza mayor, no se cuenta con la totalidad de las semanas de clase previstas, el profesor responsable determinará que temas serán propuestos para que los alumnos estudien por su cuenta.

XIV - Otros