



UNIVERSIDAD NACIONAL DE RÍO CUARTO

FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

CARRERA/S: Licenciatura en Ciencias de la Computación (Cód. 14)

PLAN DE ESTUDIOS: Año 1999 – Versión 1

ASIGNATURA: Autómatas y Lenguajes

CÓDIGO: 1961

DOCENTE RESPONSABLE: Francisco Bavera

EQUIPO DOCENTE: Dr. Francisco Bavera
Dr. Renzo Degiovanni

AÑO ACADÉMICO: 2016

REGIMEN DE LA ASIGNATURA: cuatrimestral

RÉGIMEN DE CORRELATIVIDADES: (para cursado)

<i>Aprobada</i>	<i>Regular</i>
	1956

CARGA HORARIA TOTAL: 140

TEÓRICAS: 56 hs **PRÁCTICAS:** 56 hs **LABORATORIO:** 28 hs

CARÁCTER DE LA ASIGNATURA: Obligatoria

A. CONTEXTUALIZACIÓN DE LA ASIGNATURA

Primer cuatrimestre de cuarto año.

B. OBJETIVOS PROPUESTOS

- Conocimientos básicos sobre la teoría de lenguajes y autómatas.
- Capacidad de trabajo con formalismos, de demostración de propiedades y obtención de conclusiones.
- Capacidad de usar modelos formales para resolver problemas.
- Conocimientos de los algoritmos y técnicas usadas en el análisis léxico y sintáctico.
- Habilidad en la transformación de gramáticas y en la construcción de definiciones guiadas por sintaxis.
- Conocimiento de las herramientas y técnicas usadas en la construcción de compiladores y otros procesadores de lenguajes.

C. CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR

Lenguajes. Gramáticas, la jerarquía de Chomsky. Inclusión propia de las clases. Vinculación de cada clase con la correspondiente familia de Autómatas. Vinculación de los lenguajes independientes del contexto con el problema de la compilación. Análisis sintáctico descendente y ascendente, métodos más importantes. Generación de analizadores lexicales y sintácticos, herramientas. Gramáticas de atributos y esquemas de traducción. Generación de código intermedio y objeto.

D. FUNDAMENTACIÓN DE LOS CONTENIDOS

El proceso de desarrollo de software ha ido resolviendo cada vez problemas más abstractos, planteados cada vez en un nivel más elevado respecto del ambiente de la aplicación, acompañando al crecimiento de la complejidad de los problemas que son directamente resueltos por herramientas de software. Lejos de apartarse de estas características generales, la problemática ligada a los lenguajes de programación es un ejemplo arquetípico. Este programa centra sus contenidos en algunas de las teorías que han sustentado este proceso de generalización abarcativa del proceso de desarrollo de software antes mencionado. Durante su desarrollo el alumno estudia los modelos conceptuales que sirven para la resolución de problemas del desarrollo procesadores de lenguajes y las implementaciones prácticas de las soluciones obtenidas mediante ellos, usando herramientas cada vez más generales. Se busca que el alumno manipule los formalismos estudiados demostrando propiedades y aplicándolos a la construcción de soluciones. En cuanto a la demostración de la corrección de los algoritmos y construcciones estudiadas se buscará utilizar métodos de derivación de algoritmos y construcción de autómatas que garanticen su corrección, aplicando estrategias de diseño a partir de las propiedades que debe cumplir el objeto a construir, en lugar de usar la técnica corriente en la literatura de Lenguajes y Autómatas, que consiste en la presentación arbitraria de la solución y en la demostración posterior, generalmente por inducción, de su corrección.

En todo momento se hará referencia a la vinculación de los temas de estudio tanto con su ubicación en el desarrollo histórico de la disciplina, como con la práctica profesional y con los demás contenidos de la carrera.

Las actividades se seleccionaron con el fin de cumplir con los objetivos propuestos, los cuales, todos contribuyen a fortalecer el perfil del egresado y su práctica profesional. Se hace especial hincapié en desarrollar la autonomía del alumno para aprender y utilizar los conceptos y herramientas.

La forma de evaluación consta de dos partes: (1) Dos exámenes parciales escritos. Los parciales evalúan fundamentalmente la capacidad de resolución de problemas y de implementación de soluciones; (2) la entrega de dos trabajos prácticos obligatorios que incluyen la resolución de problemas de mayor complejidad, la implementación de algoritmos y el estudio de distintos métodos.

Conocimientos esperados de los alumnos de las materias correlativas: conocimientos básicos de álgebra y lógica; conocimientos de programación y del proceso de compilación.

E. ACTIVIDADES A DESARROLLAR

CLASES TEÓRICAS: Introducción de los temas, conceptos teóricos y ejemplos de casos prácticos de su uso. Carga horaria: 4 hs. semanales.

CLASES PRÁCTICAS: Se trabaja con los conceptos introducidos usándolos tanto para construir soluciones como para demostrar propiedades. Carga horaria: 4 hs. Semanales.

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO: Ejercicios obligatorios que deben ser entregados, los mismos consisten en: 1) la implementación de algunos algoritmos usados en la resolución de problemas; 2) la resolución de los ejercicios usando distintas herramientas. Carga horaria: 28 hs. totales.

F. NÓMINA DE TRABAJOS PRÁCTICOS

Trabajo Práctico N°1: Repaso de Conceptos Previos

Trabajo Práctico N°2: Gramáticas, Autómatas Finitos y Expresiones Regulares

Trabajo Práctico N°3: Lenguajes Regulares

Trabajo Práctico N°4: Gramáticas Independientes de Contexto y Autómatas Pila

Trabajo Práctico N°5: Parsing y Gramáticas LL y LR

Trabajo Práctico N°6: Esquemas de Traducción y Gramáticas de Atributos

G. HORARIOS DE CLASES: Lunes y Miercoles de 10 hs a 14 hs.

HORARIO DE CLASES DE CONSULTAS: A determinar con los alumnos.

H. MODALIDAD DE EVALUACIÓN:

- **Evaluaciones Parciales:** Dos exámenes parciales con una instancia de recuperación cada uno. Los parciales evalúan fundamentalmente la capacidad de resolución de problemas y de implementación de soluciones.
- **Evaluación Final:** Un examen final escrito y oral. Evalúa fundamentalmente la adquisición de los conceptos fundamentales, su vinculación con el resto de la carrera y la capacidad de manipularlos y demostrar propiedades.
- **CONDICIONES DE REGULARIDAD:** Aprobación de dos parciales o de sus instancias de recuperación y entrega de los dos trabajos prácticos obligatorios.
- **CONDICIONES DE PROMOCIÓN:** No posee régimen de promoción.

PROGRAMA ANALÍTICO

A. CONTENIDOS

Unidad 0. Revisión de conceptos a ser usados. Relaciones, operaciones entre relaciones, representación mediante matrices booleanas, clausura transitiva, relaciones de equivalencia y conjunto cociente. Inducción estructural. cadenas sobre un alfabeto.

Unidad I Lenguajes, operaciones sobre lenguajes. Gramáticas, relación de derivación, lenguaje generado. Clasificación de Chomsky, ejemplos. Lenguajes regulares, Autómatas Finitos Determinísticos y No determinísticos, Autómatas No determinísticos con transiciones lambda, expresiones, conjuntos regulares, demostración de la equivalencia de las distintas definiciones. Invariantes de estado, construcción de Autómatas Finitos guiada por invariantes de estado.

Unidad II. Lema de pumping para lenguajes regulares, estados accesibles, algoritmo de eliminación de estados inaccesibles. estados indistinguibles, relación de indistinguibilidad y de k-indistinguibilidad, propiedades. Autómata reducido, Construcción de un autómata reducido equivalente a uno dado, algoritmo y demostración de la equivalencia. Propiedades de clausura de los lenguajes regulares, decisión de los problemas básicos sobre lenguajes regulares. El análisis lexicográfico símbolos o tokens, patrones y lexemas. Algoritmo de generación de analizadores lexicográficos. Herramientas existentes lex, grep y awk.

Unidad III. Lenguajes independientes del contexto. Árbol de derivación derivación más hacia la derecha y más hacia la izquierda. Ambigüedad, gramáticas ambiguas y lenguajes intrínsecamente ambiguos. Lema de pumping para lenguajes independientes del contexto. Símbolos activos y alcanzables de una gramática, gramáticas reducidas. Símbolos anulables, obtención de una gramática propia equivalente a una dada. Autómatas Pila y Autómatas Pila Determinísticos. Otros modelos de autómatas, aceptación por pila vacía, equivalencia de los dos modelos. Equivalencia de los Autómatas Pila y la Gramáticas Independientes del Contexto. Propiedades de clausura de los lenguajes independientes del contexto. Análisis sintáctico ascendente y descendente. Vinculación de los lenguajes independientes de contexto con el problema de la compilación. Parsing descendente recursivo. Computo de símbolos directrices, Gramáticas LL(1) y LL(k). Eliminación de recursividad a izquierda, factorización. Método de análisis descendente no recursivo invariante del algoritmo.

Unidad IV. Métodos ascendentes. Algoritmo de corrimiento y reducción, invariante del algoritmo. El problema de la determinación del pivote (handle). Somera idea de los métodos de precedencia simple y de operadores, construcción de las relaciones, algoritmos de reconocimiento, funciones de precedencia, computo y determinación de su existencia.

Gramáticas LR(k). Regularidad del conjunto de prefijos viables de un gramática independiente del contexto, ítems válidos LR(0), construcción del autómata el lenguajes de prefijos viables. Construcción del conjunto canónico de ítems LR(0), Gramáticas LR(0), algoritmo de reconocimiento. Gramáticas SLR(1). Gramáticas LR(1), ítem válido LR(1), Construcción del conjunto canónico de ítems LR(1), Gramáticas LR(1), construcción del conjunto canónico de ítems LALR(1) a partir del LR(1). Herramientas de generación de analizadores sintácticos (utilización de yacc)

Unidad V. Gramáticas de atributos. Atributos sintetizados y heredados. Árbol decorado de una sentencia, evaluación, grafo de dependencias, gramáticas evaluables. Esquemas de traducción. Aplicaciones: Generación de Código Intermedio y Objeto.

B. CRONOGRAMA DE CLASES Y PARCIALES

Semana	Día/ Fecha	Teóricos	D/ F	Prácticos	D/ F	Laboratorios	Parcial / Recup.
1	14/03	Introducción – Conceptos Previos – Lenguajes – Gramáticas – Lenguajes Regulares: GR, AF, ER		Repaso			
2	21/03	Equivalencia entre GR, AF, ER		Repaso			
3	28/03	Equivalencia entre GR, AF, ER		Gramáticas			
3	30/03	Propiedades de clausura de Leng. Reg. Construcción de AF.		Gramáticas			
4	4/04	Lema de pumping para LR.		Lenguajes Regulares: AF, ER y GR			
4	6/04	Minimización		Lenguajes Regulares			
5	11/04	Decidibilidad de problemas básicos. Construcción de analizadores léxico. Lex		Lenguajes Regulares			
5	13/04	Lenguajes indep. del contexto. Arbol de derivación. Ambigüedad, Gramáticas reducidas y propias.		Leng. Libres de Contexto			
6	18/04	Lema de pumping para CFL.		Leng. Libres de Contexto		Entrega ej. oblig. 1	
6	20/04	Consulta		Consulta			
7	25/04						1° Parcial
7	27/04	Autómatas Pila, Equivalencia de AP (PDA) y GIC (CFG).		Leng. Libres de Contexto			
8	2/05	Parsers ascendentes y descendentes. LL(1), Transformación de gramáticas. Parser descendente recursivo y por tablas.		Leng. Libres de Contexto y LL(1)			
8	4/05	Precedencia Simple (PS)		LL(1) y PS			
9	9/05						Recup.
9	11/05	Métodos LR. LR(0), SLR(1), LR(1), LALR.		PS			
10	16/05	LR cont.		LR		Entrega ej. oblig. 2	
10	18/05	Definiciones guiadas por sintaxis (DGS)		LR			
11	23/05	Definiciones guiadas por sintaxis cont.		LR y DGS			
12	30/05						2° Parcial
12	1/06			DGS: ET y GA			
13	6/06	Consulta		Consulta			
13	8/06	Consulta		Consulta		Entrega ej. oblig. 3	
14	13/06						Recup.
14	15/06						

C. BIBLIOGRAFÍA

De lectura obligatoria:

- Introduction to Automata Theory, Languages and Computation. J. E. Hopcroft, R. Motwani y J. Ullman. Addison-Wesley. 2007.
- Compilers: Principles, Techniques, and Tools. Second Edition, by Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman and Monica S. Lam. 2008.

De consulta:

- Teoría de la Computación. Gonzalo Navarro. Apuntes y Ejercicios. DCC-UChile. 2011 (online).
- Teoría de autómatas y Lenguajes Formales. Alfonseca, Alfonseca y Salomon. Mc Graw-Hill. 2007.
- Diseño de Compiladores Modernos. D Grune, et. al. Editorial MCGRAW-HILL. 2001.
- Modern Compiler implementation in Java. Andrew Appel, Cambridge University Press, 1998.