



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Teoría de la Computación y Verificación de Programas

Carrera/ Plan

Licenciatura en Informática Planes 90, 003/07-2012

Año: 4º y 5º

Régimen de Cursada: Semestral

Carácter: Obligatoria

Correlativas: Matemáticas 3, y Conceptos y Paradigmas de Lenguajes de Programación.

Año 2012

Profesores:., Ricardo Rosenfeld.

Hs. Semanales: 6 hs.

FUNDAMENTACIÓN

Teoría de la Computación y Verificación de Programas es una materia introductoria de fundamentos de la teoría de la computación (computabilidad y complejidad computacional) y de la teoría de correctitud de programas (semántica de lenguajes de programación, verificación formal de programas, desarrollo sistemático de programas). De este modo trata dos importantes pilares de las ciencias de la computación, necesarios en la formación de un profesional de la informática, habiendo éste ya recibido y madurado entre otros conocimientos de algorítmica y estructuras de datos, matemáticas discretas y lógica matemática. Al mismo tiempo, como distintos contenidos de la complejidad computacional y de la verificación de programas hoy día están abiertos a distintos caminos de investigación, se pretende con la materia estimular este estudio brindando herramientas básicas y esenciales.

OBJETIVOS GENERALES:

<p>Parte 1. Modelización de una computadora por medio de una máquina de Turing (MT). Estudio de la capacidad de una MT para resolver problemas (computabilidad, decidibilidad y complejidad computacional temporal), y de la relación entre las MT que reconocen lenguajes, generan lenguajes y calculan funciones.</p> <p>Parte 2. Estudio introductorio de elementos de la teoría de correctitud de programas (métodos de verificación de programas, propiedades de los métodos). Instanciación inicial en la programación de entrada/salida secuencial determinística.</p>

CONTENIDOS MINIMOS:



- Máquinas de Turing. Modelos equivalentes.
- Computabilidad, decidibilidad y complejidad computacional temporal.
- Técnicas de inducción, diagonalización y reducción de problemas.
- Lenguajes formales y autómatas. Jerarquía de Chomsky. Reconocimiento de lenguajes.
- Especificación de programas. Semántica operacional de los lenguajes de programación.

Aplicación de la lógica de primer orden.

- Métodos de verificación de programas. Propiedades de sensatez y completitud.
- Verificación de programas de entrada/salida secuenciales determinísticos.

PROGRAMA ANALÍTICO

Parte 1. Sección 1

Máquinas de Turing. Distintos modelos de máquinas de Turing. Equivalencia de modelos de máquinas de Turing. Computabilidad y decidibilidad. Lenguajes no recursivamente numerables, recursivamente numerables y recursivos.

Máquina de Turing Universal. El Problema de la Detención (Halting Problem). Diagonalización. Reducción de problemas. Teorema de Rice.

Lenguajes formales, autómatas y gramáticas. Generación y reconocimiento de lenguajes.

Clasificación de los lenguajes formales y autómatas. Jerarquía de Chomsky.

Parte 1. Sección 2.

Generalidades de la complejidad computacional temporal y espacial de problemas.

Representación de problemas. Jerarquía temporal. Tiempo polinomial determinístico y no determinístico (P y NP).

Reducción polinomial de problemas. Completitud. Problemas NP-completos. El Problema de la Satisfabilidad (SAT). Teorema de Cook. Conjuntos de problemas CO-NP, NPI y EXP.

Misceláneos: problemas de búsqueda, optimización y enumeración; máquinas de Turing probabilísticas.

Parte 2

Elementos de correctitud de programas: estado, programa, especificación, semántica de lenguajes de programación, correctitud parcial y total de programas, métodos de verificación de programas, sensatez y completitud de los métodos de verificación de programas. Aplicación de la lógica de primer orden, inducción matemática y estructural, relaciones de orden bien fundadas.

Verificación de programas de entrada/salida secuenciales determinísticos. Sistemas deductivos H y H* para correctitud parcial y total de programas de entrada/salida secuenciales determinísticos.

Sensatez y completitud de H y H*.

METODOLOGÍA DE ENSEÑANZA



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

La asignatura consiste en el dictado de clases teóricas y prácticas; ambas están estrechamente vinculadas y articuladas.

En las clases teóricas se brindan explicaciones conceptuales, con participación e intercambio de los alumnos, que aportan al abordaje de los trabajos prácticos.

En las prácticas se trabaja a partir de la entrega de enunciados que involucran la resolución de problemas teórico-prácticos.

Se utiliza la plataforma virtual de gestión de cursos para la publicación de las clases, prácticas y artículos de interés; consultas; y foro de discusión.

EVALUACIÓN

La aprobación de la cursada consiste en la entrega de trabajos prácticos quincenales, los cuales se publican en la plataforma virtual de gestión de cursos y son de carácter obligatorio. Además hay un examen al final de la dictada que requiere de un mínimo de 4 pts.

La aprobación de la materia consiste en un examen final que requiere un mínimo de 4 pts.

Nota: en caso de que el alumno haya obtenido en el examen asociado a la aprobación de la cursada 7 puntos, queda eximido del segundo examen.

Comentario: La diversidad y complejidad de algunos temas y su encadenamiento lógico, ameritan que se haga un seguimiento bastante personalizado sobre los alumnos. El mecanismo de trabajos prácticos quincenales obligatorios, previos al primer examen, ha demostrado ser un buen esquema de dictado y aprobación de la asignatura.

BIBLIOGRAFÍA OBLIGATORIA

- Teoría de la Computación y Verificación de Programas. Rosenfeld & Irazábal. Edulp y McGraw Hill. 2010.
- Apuntes publicados en la plataforma virtual de gestión de cursos, que varían año a año.

BIBLIOGRAFÍA COMPLEMENTARIA

- Introduction to Automata Theory, Language & Computation. Hopcroft y Ullman. Prentice-Hall. 1979.
- Computers and Intractability: A Guide to the Theory of NP-Completeness. M. Garey y D. Johnson. W. H. Freeman and Company, New York. 1979.
- Introduction to the Theory of Complexity. Bovet y Crescenzi. Prentice-Hall. 1994.
- Computational Complexity. Christos Papadimitriou. Addison-Wesley. 1995.
- Computational Complexity: A Conceptual Perspective. O. Goldreich. Cambridge University Press. 2008.
- Computational Complexity: A Modern Approach. S. Arora y B. Barak. Princeton Univ. 2007.
- Program Verification. Nissim Francez. Addison-Wesley. 1992.
- Verification of Sequential and Concurrent Programs. Apt y Olderog. Springer. 1997.
- Logic in Computer Science. M. Huth y M. Ryan. Cambridge University Press. 2004.



CRONOGRAMA DE CLASES Y EVALUACIONES

Contenidos/Actividades	Evaluaciones
<ul style="list-style-type: none">- Clase 1. Máquinas de Turing. Distintos modelos de máquinas de Turing y equivalencia entre ellos.- Clase 2. Lenguajes recursivos, recursivamente numerables y no recursivamente numerables. Propiedades.- Clase 3. Lenguajes y problemas de decisión. Máquina de Turing universal. El Problema de la Detención. Diagonalización. Problemas decidibles, computables y no computables.- Clases 4 y 5. Reducción de problemas. Teorema de Rice.- Clase 6. Máquinas de Turing como generadoras de lenguajes. Gramáticas. Jerarquía de lenguajes de Chomsky. Máquinas de Turing restringidas: autómatas finitos, autómatas con pila, autómatas acotados linealmente. Propiedades de las distintas clases de lenguajes de la Jerarquía de Chomsky.	Trabajo Práctico 1 de clases 1 y 2. Trabajo Práctico 2 de clases 3 y 4. Trabajo Práctico 3 de clases 5 y 6.
<ul style="list-style-type: none">- Clase 7. Complejidad computacional temporal. Jerarquía temporal. Representación de problemas.- Clase 8. Tiempo polinomial determinístico y no determinístico. Clases de problemas P y NP.	Trabajo Práctico 4 de clases 7 y 8.
<ul style="list-style-type: none">- Clase 9. Reducciones polinomiales de problemas. Problemas NP-completos. El Problema de la Satisfactibilidad. Teorema de Cook.- Clase 10. Clases de problemas NPI, CO-NP y EXP. Problemas de búsqueda, optimización y enumeración. Máquinas de Turing probabilísticas.	Trabajo Práctico 5 de clases 9 y 10.
<ul style="list-style-type: none">- Clase 11. Definiciones iniciales de la teoría de correctitud de programas: estado, programa, especificación, semántica operacional de los lenguajes de programación, correctitud parcial y total de programas, métodos de verificación de programas, sensatez y completitud de los métodos de verificación de programas.	Trabajo Práctico 6 de clases 11 y 12.
<ul style="list-style-type: none">- Clase 12. Lenguaje de programación secuencial determinístico PLW. Sintaxis y semántica operacional de PLW.- Clases 13 y 14. Método de verificación H para la correctitud parcial de programas PLW. Prueba de sensatez y completitud de H.- Clase 15. Método de verificación H* para la correctitud total de programas PLW. Prueba de sensatez y completitud de H*.	Trabajo Práctico 7 de clases 13 y 14. Trabajo Práctico 8 de clase 15. Primer examen . Segundo examen.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

Mails: rosenfeld@pragmaconsultores.com y jeronimo.irazabal@gmail.com

Plataforma: <http://webunlp.unlp.edu.ar>

Firma del profesor responsable:

Ricardo F. Rosenfeld

Firmas del/los profesores responsables: