



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

**TEORIA DE LA COMPUTACION Y  
VERIFICACIÓN DE PROGRAMAS  
AVANZADA**

**Año 2010**

Carrera: *Licenciatura en Informática*  
**Planes 2003 y 2007.**

Año: *Optativa*

Duración: *Semestral*

Profesor: *Ricardo Rosenfeld*

Hs. semanales: **6 hs**

**OBJETIVOS GENERALES:**

Profundización del análisis estructural de la complejidad computacional, con foco en la complejidad computacional espacial.

Profundización de la teoría de correctitud de programas, considerando la programación no determinística y concurrente, e introduciendo la problemática de la verificación de los programas reactivos utilizando la lógica temporal.

**CONTENIDOS MINIMOS:**

- Jerarquía espacial. Relación con la jerarquía temporal.
- Máquinas de Turing con oráculos, Máquina de Turing probabilísticas, problemas de optimización y aproximaciones polinomiales.
- Verificación de programas no determinísticos. Fairness.
- Verificación de programas concurrentes. Propiedades de tipo safety y liveness.
- Verificación de programas reactivos por medio de la lógica temporal.
- Introducción a la semántica denotacional.

**Programa**

**Parte 1.**

Espacio logarítmico determinístico (LOG) y no determinístico (NLOG). Espacio polinomial determinístico (PSPACE) y no determinístico (NPSPACE). Teorema de Savitch. Teorema de Immerman. Reducción log-space de problemas. Problemas complejos de las distintas clases de la jerarquía espacial.

Máquinas de Turing con oráculo. Máquinas de Turing probabilísticas. Problemas de búsqueda y enumeración, de optimización y aproximaciones polinomiales.

**Parte 2.**

Verificación de programas no determinísticos. Métodos D y D\* de verificación de programas no determinísticos. El concepto de fairness.

Verificación de programas concurrentes con memoria compartida. Métodos de verificación de programas concurrentes con memoria compartida, sin y con primitivas de sincronización (O, O\*, R y R\*). Sensatez y completitud de los métodos.



Propiedades safety y liveness (ausencia de deadlock, exclusión mutua, ausencia de starvation, etc).

Verificación de programas distribuidos (concurrentes sin memoria compartida). Métodos de verificación de programas distribuidos. Sensatez y completitud de los métodos. Propiedades safety y liveness (ausencia de deadlock, exclusión mutua, ausencia de starvation, etc).

Verificación de programas reactivos. Métodos de verificación basados en la lógica temporal para la verificación de programas reactivos.

Misceláneos: extensión de los métodos H y H\* para la verificación de programas con procedimientos, e introducción a la semántica denotacional.

### **Metodología de enseñanza**

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

En las teorías se desarrollan y 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 enunciados que implican la resolución de problemas teóricos y prácticos.

Se dispone de un Sitio de Internet compartido para consultas, foro de discusión, y publicación de artículos interesantes sobre los tópicos tratados.

### **Propuesta de evaluación**

La aprobación de la cursada requiere de la entrega de trabajos prácticos quincenales, los cuales son publicados en la plataforma virtual y de carácter obligatorio. Además hay un examen al final de la dictada que requiere un mínimo de 4 pts, de carácter teórico-práctico.

Aprobación de la materia se efectúa una examinación final con un mínimo de puntaje de 4 pts. *Nota: en caso de que el alumno en el examen asociado a la cursada haya obtenido 7 puntos, queda eximido del segundo examen. En ciertas ocasiones en lugar del examen final se desarrolla un coloquio o un trabajo final individual sobre uno de los tópicos de la asignatura.*

Evaluación del seguimiento: la diversidad y profundidad de los temas y su encadenamiento lógico, ameritan que se haga un seguimiento bastante personalizado sobre los alumnos. El mecanismo de trabajos prácticos quincenales obligatorios, previos a la examinación final, ha demostrado ser un buen esquema de dictado y aprobación de la materia.

### **BIBLIOGRAFIA BASICA:**

- Introduction to Automata Theory, Language & Computation. Hopcroft & Ullman. Prentice-Hall. 1979.
- Introduction to the Theory of Complexity. Bovet & Crescenzi. Prentice- Hall. 1994.
- Computational Complexity. Christos Papadimitriou. Addison-Wesley. 1995.



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

---

- Program Verification. Nissim Francez. Addison-Wesley. 1992.
- Verification of Sequential and Concurrent Programs. Apt & Olderog. Springer. 1997.
- Logic in Computer Science. Huth & Ryan. Cambridge University Press. 2004.
- Teoría de la Computación y Verificación de Programas. Rosenfeld & Irazábal. Edulp y McGraw Hill. 2010.