

C.96'
Ⓟ

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

- 1. DEPARTAMENTO: Computación.
- 2. CUATRIMESTRE: Primero de 1996
- 3. ASIGNATURA: **DIFERENTES ENFOQUES SEMANTICOS**
- 4. CARRERA: DOCTORADO EN CIENCIAS DE LA COMPUTACION
- 5. CARÁCTER DE LA MATERIA: Optativa
- 6. NUMERO DE CÓDIGO DE CARRERA: 18
- 7. NUMERO DE CÓDIGO DE MATERIA: C022
- 8. PUNTAJE: 1 punto
- 9. PLAN DE ESTUDIOS AÑO: -----
- 10. DURACIÓN DE LA MATERIA: Cuatrimestral
- 11. HORAS DE CLASE SEMANAL:
 - a) TEÓRICAS 4HS.
 - b) LABORATORIO HS.
 - c) PROBLEMAS 4 HS.
 - d) SEMINARIOS
- 12. CARGA HORARIA TOTAL: 8HS.
- 13. ASIGNATURAS CORRELATIVAS: -----
- 14. FORMA DE EVALUACIÓN: Prácticos y Final
- 15. PROGRAMA Y BIBLIOGRAFÍA: Adjuntas a esta hoja

FECHA: 15/04/96

Firma del Profesor

Lic. MONICA BOBROWSKI

Firma del Director

Lic. ROBERTO BEVILACQUA
DIRECTOR ADJUNTO INTERINO
DEPARTAMENTO DE COMPUTACION

Sello Aclaratorio

C 96
4

**PROPUESTA DE MATERIA OPTATIVA
PRIMER CUATRIMESTRE 1996**

Título: Diferentes enfoques semánticos

Docente: Mónica Bobrowski

Carga horaria: 6 horas semanales (entre teoría y práctica)

Puntaje sugerido: 2 puntos plan 87. Se podría analizar la posibilidad de dar puntos para la carrera de doctorado.

Cantidad máxima de alumnos: 20 alumnos (extensible a 25 si se incorporan docentes). En caso de ser mayor el número de inscriptos, la cátedra elegirá un método de selección que será informado a los alumnos a más tardar en la primera clase.

Prerrequisitos:

- Lógica
- Nociones de Computabilidad
- Matemática Discreta

Introducción:

Al dar semántica a un lenguaje de programación se construyen modelos matemáticos. Su propósito es servir como base para comprender y razonar sobre el comportamiento de los programas. Un modelo matemático no sólo es útil para realizar varios tipos de análisis y verificación. El solo hecho de tratar de definir el significado de las estructuras de un programa de manera precisa puede revelar todas las sutilezas de las cuales hay que cuidarse.

Históricamente, la semántica de los lenguajes de programación a menudo es vista desde tres enfoques:

La *Semántica Operacional* describe el significado de un lenguaje de programación especificando cómo se ejecuta sobre una máquina abstracta.

La *Semántica Denotacional* utiliza conceptos matemáticos más abstractos para definir el significado de un lenguaje de programación. La denotación de un comando es una función parcial sobre estados.

La *Semántica Axiomática* trata de fijar el significado de una construcción sintáctica dando reglas de prueba en el marco de una lógica.

Sin embargo, resulta incorrecto ver estos tres estilos como opuestos. Cada uno tiene su utilidad.

En este curso, estudiaremos los fundamentos de los tres enfoques descritos, analizando la utilidad de aplicación de cada uno mediante casos de estudio.

Objetivos:

- Que el alumno conozca tres formas de dar semántica a lenguajes de programación
- Que el alumno compare los métodos estudiados, analizando la conveniencia de su aplicación
- Que el alumno aplique los métodos a casos de estudio
- Que el alumno investigue otras formas de dar semántica y otros tipos de lenguajes de programación

Contenidos:

- Repaso de teoría básica de conjuntos (conjuntos, relaciones y funciones, notación Lambda, etc.).
- Definición de un lenguaje imperativo simple IMP
- Introducción a la semántica operacional - Semántica operacional de IMP
- Algunos principios de inducción (matemática, estructural, bien fundada)
- Definiciones inductivas
- Semántica denotacional de IMP
- Semántica axiomática de IMP
- Conclusiones

Metodología:

La materia consistirá en clases teóricas complementadas con trabajos prácticos con ejercicios a ser resueltos por los alumnos. En la última parte de la materia, los alumnos deberán elegir un tema para profundizar, que podrá ser escogido de los presentados por la cátedra o acordado con la misma a propuesta del alumno. Entre los temas sugeridos por la cátedra figuran: Tres enfoques sobre un caso complejo, No determinismo y paralelismo, Semántica de lenguajes funcionales, Semántica de lenguajes lógicos, etc. La cátedra pondrá a disposición de los alumnos la bibliografía necesaria para estudiar los temas.

Evaluación:

La materia se evaluará mediante dos parciales que abarcarán los conceptos estudiados y por medio de la exposición del trabajo de profundización sobre el tema elegido por el alumno. Estos trabajos podrán ser a lo sumo en parejas, dependiendo de la complejidad de los temas.

Bibliografía básica:

- Winskel, G.: "The Formal Semantics of Programming Languages - An Introduction", MIT Press, 1993.
- Plotkin, G.D.: "Structural Operational Semantics", Lecture Notes, DAIMI FN-19, Aarhus University, Denmark, 1981
- Halmos, P.R.: "Naive Set Theory", Litton Ed Publ. Inc., 1960
- Gries, D.: "The Science of Programming", Springer Texts and Monographs in Computer Science, 1981
- Stoy, J.: "Denotational Semantics: the Scott-Strachey approach to programming language theory", MIT Press, 1977