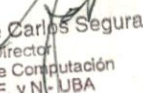


UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES

1. DEPARTAMENTO: Computación
2. CUATRIMESTRE: Segundo de 2004.
3. ASIGNATURA: **Técnicas de análisis estático de programas. Aplicación a la gestión automática de la memoria dinámica en sistemas embebidos**
4. CARRERA: Licenciatura en Ciencias de la Computación
5. CARACTER DE LA MATERIA: Optativa
6. NUMERO DE CODIGO DE CARRERA: 18
7. NUMERO DE CODIGO DE MATERIA:
8. PUNTAJE: 1 punto
9. PLAN DE ESTUDIOS AÑO: 1993
10. DURACION: 1 semana
11. HORAS DE CLASE SEMANAL:
a) TEORICAS/PRACTICAS: 15 horas b) LABORATORIO: c) PRACTICAS:
12. CARGA HORARIA TOTAL: 15 horas
13. ASIGNATURAS CORRELATIVAS: **Conocimientos de Teoría de Lenguajes, Lógica y Computabilidad**
14. FORMA DE EVALUACION: final
15. PROGRAMA Y BIBLIOGRAFIA:

Profesor
Dr. Sergio Yovine


Dr. Enrique Carlos Segura
Director
Depto. de Computación
F. C. E. y N. UBA

15) PROGRAMA:

Introducción: En los últimos años los lenguajes tales como C++, C# y Java se han convertido en los más populares debido a su simplicidad y portabilidad. Además existe una amplia gama de herramientas de desarrollo y librerías que los soportan. Sin embargo, la manera en que se administra la memoria en este tipo de lenguajes, si bien es muy cómoda para los desarrolladores, no es adecuada para entornos embebidos.

Una solución prometedora a este problema consiste en la utilización de técnicas de análisis de programas para la obtención automática de programas compatibles con estos entornos restringidos, sin limitar, a priori, el lenguaje de programación.

Objetivos: El objetivo del curso es, por un lado, presentar a los alumnos un panorama de las técnicas de análisis estático de programas, en particular las basadas en interpretación abstracta. Por otro lado, presentar una aplicación concreta de esas técnicas en el contexto de lenguajes con gestión dinámica de la memoria, con el fin de predecir propiedades sobre la forma y el tamaño del heap.

Programa: Clase 1: Introducción

1. Presentación del curso
 - a. Motivación

- b. Objetivos
- c. Historia

2. Nociones fundamentales
 - a. Ordenes parciales

- b. Reticulados
- c. Puntos fijos
- d. Poliedros
- e. Polinomios de Ehrhart

Clase 2: Interpretación Abstracta

1. Definición
2. Conexiones de Galois
3. Semánticas concretas y abstractas
4. Problemas de terminación del punto fijo
5. Aproximaciones y aceleraciones (Widening).
6. Aplicación a la verificación de programas

Clase 3: Generación automática de invariantes

1. Análisis estático
2. Análisis de un programa simple
3. Cálculo de invariantes poliédricos.
4. Aplicaciones

Clase 4: Análisis estático de la memoria dinámica

1. Pointer Analysis (análisis de la relación entre punteros)
2. Escape Analysis (análisis de interacción y tiempo de vida de los objetos)
3. Shape Analysis (análisis de la forma del heap)
4. Aplicaciones

Clase 5: Aplicaciones y Conclusiones

1. Aplicaciones del análisis estático de programas
2. Un ejemplo detallado: Predicción del consumo de memoria
 - a. Motivación

- b. Problemas
- c. Estrategia
- d. Resultados obtenidos hasta el momento
 3. Presentación del TP
4. Conclusiones del curso

Dr. Enrique Carlos Segura
Director
Depto. de Computación
F. C. E. y N - UBA

16) BIBLIOGRAFIA:

Apunte de cátedra. No fue adjuntada otra bibliografía por parte del docente a cargo.

Dr. Enrique Carlos Segura
Director
Depto. de Computación
F. C. E. y N. UBA