

## PROGRAMA DE ASIGNATURA

<b>ASIGNATURA:</b> Ingeniería del Software II	<b>AÑO:</b> 2012
<b>CARÁCTER:</b> Obligatoria	
<b>CARRERA:</b> Licenciatura en Ciencias de la Computación	
<b>RÉGIMEN:</b> cuatrimestral	<b>CARGA HORARIA:</b> 120 hs.
<b>UBICACIÓN en la CARRERA:</b> Quinto año – Primer cuatrimestre	

### FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno

- se familiarice con las técnicas de análisis de software y de garantía de calidad avanzadas.
- Comprenda los fundamentos detrás del análisis estático, el model-checking, el análisis basado en constraint solving y el testing.

### CONTENIDO

I. El problema de la corrección del software: (1) Definición de sistemas críticos, (2) Limitaciones del testing y la simulación, (3) Discusiones sobre verificación, (4) Métodos formales para Ingeniería de Software.

II. Programación concurrente: (1) Definición de sistemas reactivos, (2) Interacción entre procesos, (3) Los problemas de la concurrencia, (4) Semántica de los programas concurrentes, (5) Interleaving y no determinismo, (6) Razonamiento sobre programas concurrentes, (7) La necesidad de abstraer para modelar, (8) El lenguaje de modelado FSP: sintaxis y semántica, (9) La herramienta LTSA.

III. Sincronización de procesos concurrentes: (1) Recursos compartidos: interferencia y exclusión mutua, (2) Detección de errores, (3) Monitores, sincronización condicional e invariantes del monitor, (4) Semáforos y su invariante, (5) Buffers acotados, (6) Bisimulación como equivalencia de procesos, (7) Comunicación mediante pasaje de mensajes, (8) Pasaje sincrónico de mensajes, (9) Recepción selectiva, (10) Pasaje asincrónico de mensajes, (11) Rendezvous. (12) Transacciones distribuidas.

IV. Propiedades de los sistemas concurrentes: (1) Categorías de propiedades: *alcanzabilidad*, *safety*, *liveness*, y *fairness*, (2) Necesidad de la categorización de propiedades, (3) Propiedades como conjuntos de trazas, (4) Lenguajes  $\omega$ -regulares, (5) Formalización de las propiedades de *safety* y *liveness*, (6) Otras propiedades, (7) Análisis automatizado de propiedades usando FSP: deadlock, *safety* y *liveness*.

V. Lógicas temporales: (1) Limitaciones de los métodos previos y de las lógicas usuales, (2) Lógicas modales, (3) Introducción a las lógicas temporales, (4) La lógica temporal lineal LTL, (5) Sintaxis y semántica, (6) Operadores derivados y leyes, (7) Especificación de propiedades con LT: *Safety* y *Liveness*, (8) *Fairness*: incondicional, débil y fuerte, (9) Otros tipos propiedades en LTL.

VI. Model checking: (1) El modelo de un sistema, (2) Autómatas de Büchi: definición y uso para presentar programas y propiedades, (3) Model Checking de propiedades LTL con enfoque en la teoría de autómatas, (4) Herramientas de model checking, (5) El model checking de propiedades descritas en LTL Spin, (6) Promela: modelado y análisis, (7) El model checker de propiedades descritas en CTL (computational tree logic) SMV, (8) El model checker de propiedades de tiempo Uppaal, (9) Otros model checkers: Kronos y Zeus, Prism y Liquor, etc.

VII. Especificaciones de sistemas: (1) Características de los lenguajes de especificación, (2) Las lógicas como lenguajes de especificación, (3) Lógica proposicional: Sintaxis, semántica y poder expresivo, (4) SAT solving en la lógica proposicional: ventajas y desventajas, (5) Lógica de primer orden: Sintaxis, semántica y poder expresivo, (6) SAT solving en la lógica de primer orden, (7) El álgebra relacional. Sintaxis, Semántica y Axiomas

VIII. El lenguaje de especificación Alloy: (1) Sintaxis del lenguaje Alloy, (2) Características de Alloy, (3) Uso de Alloy para la resolución de problemas con restricciones (constraint solving), (4) Modelos de ejecuciones, (5) Uso de Alloy para verificar refinamientos, (6) Análisis de especificaciones en Alloy: Cotas, cuantificadores no acotados, axiomas de generación.

IX. Algoritmos para verificar satisfactibilidad en lógica proposicional: (1) Algoritmos simples: Tablas de verdad y argumentos semánticos, (2) Algoritmos avanzados, (3) Tablas de verdad revisadas, (4) Conversión a forma normal conjuntiva, (5) Regla de resolución clausal, (6) Propagación de restricciones booleanas, (7) El algoritmo de Davis, Putnam, Logemann & Loveland, (8) Cláusulas de Horn, (9) Linealidad de la resolución en la lógica de Horn, (10) La lógica de Horn como base de la programación lógica y los demostradores automáticos de teoremas.

X. Testing: (1) Definición del testing basado en modelos, (2) Testing con modelos formales, (3) El proceso de testing formal, (4) Conformidad correccion y exhaustividad, (5) La teoría de conformidad de testing basada en entradas y salidas (ioco: Input/Output Conformance Testing), (6) Extensión con tiempo y canales de la teoría ioco, (7) Definición de cubrimiento semántico.

## BIBLIOGRAFÍA

- [1] J. Magee y J. Kramer. Concurrency: State Models & Java Programs, 2<sup>nd</sup> edition. Wiley 2006.
- [2] B. Alpern y F. Schneider. Defining Liveness. Information Processing Letter 21:181-185. 1985
- [3] B. Alpern y F. Schneider. Recognizing Safety and Liveness. Distributed Computing 2 (3): 117-126. 1987.
- [4] C. Baier and J.-P. Katoen. Principles of Model Checking. MIT Press, 2008.
- [5] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen. Systems and Software Verification Model-Checking Techniques and Tools. Springer, 2001.
- [6] M. Müller-Olm, D. Schmidt, B. Steffen. Model-Checking: A Tutorial Introduction. En A. Cortesi, G. Filé (Eds.), Procs. Of SAS'99, LNCS 1694, pp. 330-354. Springer 1999.
- [7] G. J. Holzmann. The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, 2003.
- [8] D. Jackson. Software Abstractions: Logic, Language, and Analysis. MIT Press, 2006.
- [9] A.R. Bradley y Z. Manna. The Calculus of Computation: Decision Procedures with Applications to Verification. Springer, 2007.
- [10] P. Jalote. *An Integrated Approach to Software Engineering*, Third Edition. Springer. 2005.
- [11] J. Tretmans. A formal Approach to Conformance Testing. PhD Thesis. Univeristeit Twente, The Netherlands, 1992.

## METODOLOGÍA DE TRABAJO

La materia se organiza en clases teóricas, clases prácticas y actividades de laboratorio. En las clases teóricas se brindan los contenidos fundamentales de la asignatura siguiendo la selección de capítulos de la bibliografía..

En las clases prácticas se ejercita sobre los temas cubiertos en la teoría, con especial énfasis en actividades de modelado y análisis.

La asignatura incluye el desarrollo de dos trabajos prácticos con entrega, que son resueltos en grupo en un período de una (1) semana cada uno. En estos trabajos prácticos se experimenta con problemas de tamaño más significativo, en comparación con los ejercicios de las clases prácticas, y se integran contenidos de diferentes tópicos de la materia.

## **EVALUACIÓN**

### Condiciones para la regularidad

- Aprobación del proyecto, pasando satisfactoriamente cuatro instancias de seguimiento/evaluación del mismo.

### Condiciones para la aprobación

- Aprobación de un examen final escrito.
- Defensa del proyecto.

### Promoción

- Aprobación del proyecto
- Aprobación de dos evaluaciones parciales con nota no inferior a 7 (siete).