

# Programa de Ingeniería del Software II

- I. **El problema de la corrección del software:** (1) Definición de sistemas críticos, (2) Limitaciones del testing y la simulación, (3) Discusiones sobre verificación.
- 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 LTL: 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 representar programas y propiedades, (3) Model Checking de propiedades LTL con enfoque en la teoría de autómatas. (4) Herramientas de model checking, (5) El model checker de propiedades descriptas en LTL Spin, (6) Promela: modelado y análisis, (7) El model checker de propiedades descriptas en CTL (computational tree logic) SMV, (8) El model checker de propiedades de tiempo Uppaal (9) Otros model checkers.
- 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 y axiomas de generación.
- IX. **Algoritmos para verificar satisfactibilidad en lógica proposicional:** (1) Algoritmos simples: Tablas de verdad y argumentos semántico, (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 corrección 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.