

# Métodos formales “livianos”

## Programa

### 1 El Problema de construir programas correctos.

- 1.1 ¿Qué es un programa correcto?
- 1.2 Necesidad de las especificaciones de programas.
- 1.3 Anotaciones en programas.
- 1.4 Revisión de enfoques tradicionales a la validación y verificación de programas.
- 1.5 Testing y sus limitaciones.
- 1.6 Lógica de Hoare.
- 1.7 Programas anotados.
- 1.8 Programas correctos por construcción.
- 1.9 Transformación de programas.
- 1.10 Problemas en la automatización de las técnicas tradicionales de validación y verificación de programas.

### 2 La lógica como lenguaje formal y su expresividad.

- 2.1 Lógica proposicional.
- 2.2 Sus ventajas y sus limitaciones en poder expresivo.
- 2.3 Reducción del problema de la deducción a SAT solving proposicional.
- 2.4 Automatización de la deducción.
- 2.5 La lógica de primer orden.
- 2.6 Características y poder expresivo.
- 2.7 Restricciones para la aplicación de SAT solving para deducción en lógica de primer orden.
- 2.8 Representación de diseños de sistemas mediante lógica de primer orden.
- 2.9 Algunas extensiones importantes a la lógica de primer orden.
- 2.10 La lógica relacional.
- 2.11 Descripción de diseños y propiedades estructurales mediante la lógica relacional.
- 2.12 El poder de SAT solving de primer orden como una técnica de validación de sistemas.
- 2.13 Limitaciones.
- 2.14 La herramienta Alloy.
- 2.15 Definición de diseños mediante firmas y funciones de transformación de estados.
- 2.16 Validación de diseños en Alloy.
- 2.17 Utilización en la práctica.
- 2.18 Ejemplos.

### 3 La importancia del análisis de propiedades de ejecuciones de programas.

- 3.1 Safety y liveness.
- 3.2 Limitaciones de Alloy como una herramienta para la caracterización y validación de propiedades de ejecuciones de programas.
- 3.3 La lógica dinámica.
- 3.4 Especificaciones en lógica dinámica.
- 3.5 Algunas propiedades metalógicas de la lógica dinámica.
- 3.6 La herramienta DynAlloy.
- 3.7 Especificaciones en DynAlloy y su validación.
- 3.8 Atomización de programas y validación incremental.

## Bibliografia

- "Logic in Computer Science, modelling and reasoning about systems" Michael Huth and Mark Ryan, Cambridge University Press, 2004.
- Model Checking (Hardcover) by Edmund M. Clarke, Orna Grumberg, Doron A. Peled The MIT Press, 2000.
- Daniel Jackson: Alloy: a lightweight object modelling notation. ACM Trans. Softw. Eng. Methodol. 11(2): 256-290 (2002)
- Daniel Jackson, Ilya Shlyakhter, Manu Sridharan: A micromodularity mechanism. ESEC / SIGSOFT FSE 2001: 62-73
- Marcelo F. Frias, Rodolfo Gamarra, Gabriela Steren, Lorena Bourg: A strategy for efficient verification of relational specifications, based on monotonicity analysis. ASE 2005: 305-308
- Marcelo F. Frias, Juan P. Galeotti, Carlos Lopez Pombo, Nazareno Aguirre: DynAlloy: upgrading alloy with actions. ICSE 2005: 442-451
- Marcelo F. Frias, Carlos Lopez Pombo, Gabriel A. Baum, Nazareno Aguirre, T. S. E. Maibaum: Reasoning about static and dynamic properties in alloy: A purely relational approach. ACM Trans. Softw. Eng. Methodol. 14(4): 478-526 (2005)
- Marcelo F. Frias, Carlos Lopez Pombo, Nazareno Aguirre: An Equational Calculus for Alloy. ICFEM 2004: 162-175
- Marcelo F. Frias, Carlos Lopez Pombo, Gabriel Baum, Nazareno Aguirre, T. S. E. Maibaum: Taking Alloy to the Movies. FME 2003: 678-697