Study programme(s): Informatics (IM)

Level: master

Course title: Formal methods in engineering (IB333)

Lecturer: Zoran D. Budimac, Danijela D. Tešendić

Status: obligatory for the module of *Software Engineering;* elective for other modules.

ECTS: 7,5

Requirements: None

Learning objectives

To provide students with deep understanding and critical evaluation of formal methods and to give fundamental details of certain techniques based on automata theory and software tools based on industry-strength tools like the "Statemate".

Learning outcomes

Minimal: Students should be able to critically evaluate the need to establish reliability in large-scale computer systems and to appreciate fundamentals of formal methods. It is also expected that the students will accept the basic conclusions on using formal techniques in the life-time cycle of the system, especially in requirements and architecture design phases.

Optimal: Students should be able to critically evaluate different kinds of large-scale systems and different kinds (transforming to hybrid) of systems. In addition, students are expected to appreciate the role of tools and methods for the formal methods engineering.

Syllabus

Theoretical instruction

Theoretical foundations of large-scale systems, classification of formal methods, transforming, reactive and hybrid systems, automata theory, state-oriented development methods, state diagrams, activity diagrams, real-time aspects.

Practical instruction

Introduction to semantics and the "*Statemate"* tool. Development of real-time system/Analysis and development of several case studies.

Literature

1. Nissim Francez, 'Program Verification', Addison-Wesley, 1992
2 S. Hassoun and T Sasao, 'Logic Synthesis and Verification', 2002

Weekly teaching load				Other:
Lectures: 2	Exercises: 3	Other forms of teaching:	Student research:	

Teaching methodology

Classical methodology is applied in lectures with the use of a video-beam. Exercises are conducted in traditional teaching methods, with the use of a beam-projector, and serve to analyze study examples and practical exercises on computers to practice the principles introduced and get familiar with the recommended tools. Students build their knowledge by researching the topics, whereby knowledge is checked through producing the papers that are presented during and at the end of the course.

Grading (maximum number of points 100)

Pre-exam obligations	**points**	**Final exam**	**points**
Active participation in lectures		Written exam	
Practical instruction	**12**	Oral exam	**40**
Colloquia		.....	
Seminar(s)	**48**		