UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

Tools and Techniques for Program Analysis, Verification and Transformation

Learning Guide – Information for Students

1. Description

Grade	Máster Universitario en Ingeniería de Software – European Master on Software Engineering
Module	Advanced Software Engineering Aspects
Area	
Subject	Tools and Techniques for Program Analysis, Verification and Transformation
Type	Elective
ECTS credits	4
Responsible department	Computer Languages and Systems and Software Engineering
Major/Section/	

Academic year	2012/2013
Term	2nd term
Language	English
Web site	http://lml.ls.fi.upm.es/av

2. Faculty

NAME and SURNAME	OFFICE	email
Germán Puebla (Coord.)	2305	german@fi.upm.es

3. Prior knowledge required to take the subject

Passed subjects	•
Other required learning outcomes	•

4. Learning goals

SUBJECT-SPECIFIC COMPETENCES AND PROFICIENCY LEVEL		
Code	**Competence**	**Level**
SC13	To have a vision of the different specific and emergent aspects of Software Engineering, and to go further in some of them.	S
SC14	To understand what nowadays software engineering procedures can and cannot reach, their limitations and their possible future evolution.	S

Proficiency level: knowledge (K), comprehension (C), application (A), and analysis and synthesis (S)

SUBJECT LEARNING OUTCOMES

Code	Learning outcome	Related competences	Proficiency level
LR1	Within an application field of Software Engineering, uses and designs the appropriate solution to solve some of its problems, describing the technical difficulties and the application limits	SC13, SC14	A
LR2	Facing a real problem, chooses an appropriate Software Engineering solution, analyzing its viability, what can and cannot be achieved from the current state of development of the selected solution, and what is expected to advance in the future	SC13, SC14	A
LR3	Explains which are the Software Engineering limits and frontiers, and the base of new tendencies and developments and advanced topics and their possible application	SC13, SC14	A

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I1	Appropriate use of static analyzers	LR1
I2	Use of the COSTA system	LR1, LR3
I3	Generation of JML specifications	LR2
I4	Use of the KeY system	LR1, LR3
I5	Use of partial evaluators	LR2

(Optionally, use rubric table instead)

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
Exercises on static analysis	1st week	Laboratory	10.00%
Advanced exercises on static analysis	2nd week	Laboratory	15.00%
Exercises on the use of COSTA	3rd week	Laboratory	10.00%
Advanced exercises on the use of COSTA	4th week	Laboratory	15.00%
Exercises on the use of KeY	5th week	Laboratory	10.00%
Advanced exercises on the use of Key	6th week	Laboratory	15.00%
Exercises on partial evaluation	7th week	Laboratory	10.00%
Advanced exercises of partial evaluation	8th week	Laboratory	15.00%
			Total: 100%

UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

GRADING CRITERIA

The course will be graded using a continuous assessment system. For this, the final grade will be the average grade of the exercises which will be performed at the weekly laboratory sessions. In such sessions, the students will try to solve a number of exercises of different size. The execises will be turned in at the end of the laboratory session, except for larger exercises which may be finalized independently by the students over a longer period.

Solving the exercises will require both some expertise with the use of the tools studied in the course and some understanding on the underlying techniques for program analysis, verification and transformation.

6. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Static Analysis	1.1 Static Analysis vs Dynamic Analysis	I1
	1.2 Tools for Static Analysis	I1
	1.3 Foundations of Abstract Interpretation	I1
	1.4 Numeric Abstract Domains. Widening and Narrowing	I1
Chapter 2: Resource Consumption	2.1 Cost Models. Upper and Lower Bounds. Asymptotic Cost	I2
	2.2 The COSTA system	I2
	2.3 Brief Introduction to Java and Java bytecode	I2
	2.4 Generation of Recurrence Relations	I2
Chapter 3: Program Verification	3.1 First Order Logic and interactive proofs in the KeY system	I4
	3.2 The JML specification language	I3
	3.3 Program Logics for Java	I4
	3.4 Translating JML to Program Logics	I3, I4
	3.5 Interactive Verification of Proof Obligations in Key	I4
Chapter 4: Program Transformation	4.1 Program Folding and Unfolding	I5
	4.2 Partial Evaluation	I5
	4.3 Applications of Partial Evaluation	I5

7. Brief description of organizational modalities and teaching methods

TEACHING ORGANIZATION		
Scenario	**Organizational Modality**	**Purpose**
	Theory Classes	***Talk to students***
	Seminars/Workshops	***Construct knowledge through student interaction and activity***
	Practical Classes	***Show students what to do***
	Placements	***Round out student training in a professional setting***
	Personal Tutoring	***Give students personalized attention***
	Group Work	***Get students to learn from each other***
	Independent Work	***Develop self-learning ability***

TEACHING METHODS			
	Method	**Purpose**	
	Explanation/Lecture	***Transfer information and activate student cognitive processes***	Known as explanation, this teaching method involves the "*p* *aim of providing information organized according to criteria* known as *lecture*, mainly focuses on the verbal exposition b study. The term *master class* is often used to refer to a spe special occasions
	Case Studies	***Learning by analyzing real or simulated case studies***	Intensive and exhaustive analysis of a real fact, problem or interpreting or solving the problem, generating hypotheses, and, sometimes, training in possible alternative problem-so

	Exercises and Problem Solving	***Exercise, test and practice prior knowledge***
	Problem-Based Learning (PBL)	***Develop active learning through problem solving***
	Project-Oriented Learning (POL)	***Complete a problem-solving project applying acquired skills and knowledge***
	Cooperative Learning	***Develop active and meaningful learning through cooperation***
	Learning Contract	***Develop independent learning***

Situations where students are asked to develop the suitabl
applying formulae or running algorithms, applying informati
results. It is often used to supplement lectures.

Teaching and learning method whose starting point is a pro
has to solve to develop a number of previously defined con

Teaching and learning method where have a set time to de
task by planning, designing and completing a series of acti
applying what they have learned and making effective use

Interactive approach to the organization of classroom work
their peers' learning as part of a co-responsibility strategy f

This is both one of a number of methods for use and an ov

An agreement between the teacher and student on the ach
independent work proposal, supervised by the teacher, and
essential points of a learning contract are that it is a written
requiring personal involvement and having a time frame for

BRIEF DESCRIPTION OF THE ORGANIZATIONAL MODALITIES AND TEACHING METHODS	
THEORY CLASSES	Will be used for presenting the techniques underlying the tools to be used during the course
PROBLEM-SOLVING CLASSES	Not planned in this course
PRACTICAL WORK	Half of the lecture time will be at a laboratory, where the students will practice with the use of the tools presented in the course
INDIVIDUAL WORK	The students will have to reinforce the material presented during the theory classes.
GROUP WORK	Not planned in this course
PERSONAL TUTORING	Students will be able to attend personal tutoring, following the procedure established at the School

8. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	Nielson, Nielson, Hankin. *Principles of Program Analysis*. 2nd Ed, 2005. Springer.
	E. Albert, P. Arenas, S. Genaim, G. Puebla, and D. Zanardini. *Resource usage analysis and its application to resource certification*. In *Foundations of Security Analysis and Design V, FOSAD Tutorial Lectures*, vol. 5705 of *LNCS*. Springer, 2009
	Bernhard Beckert, Reiner Hähnle, Peter H. Schmitt (Eds.). *Verification of Object-Oriented Software: The KeY Approach*. 2006, Springer.
	N.D. Jones, C.K. Gomard, and P. Sestoft, *Partial Evaluation and Automatic Program Generation.* Prentice Hall International, June 1993
WEB RESOURCES	Subject web site (http://lml.ls.fi.upm.es/av)
EQUIPMENT	Laboratory
	Room
	Access to the COSTA and KeY systems

9. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 1 (13 hours)	• Introduction to Static Analysis (2 hours)	• Getting acquainted with static analyzers (2 hours)	• Personal study (4 hours) • Use of static analyzers (5 hours)			
Week 2 (13 hours)	• Foundations of Abstract Interpretation (2 hours)	• Numeric Abstract Domains (2 hours)	• Personal study (4 hours) • Advanced use of static analyzers (5 hours)			
Week 3 (13 hours)	• Cost Models. Upper and Lower Bounds. Asymptotic Cost (2 hours)	• Introduction to the COSTA system (2 hours)	• Personal study (4 hours) • Use of COSTA (5 hours)			
Week 4 (13 hours)	• Generation of Recurrence Relations (2 hours)	• Java and Java bytecode. Advanced use of COSTA (2 hours)	• Personal study (4 hours) • Advanced use of COSTA (5 hours)			
Week 5 (14 hours)	• First Order Logic and the JML specification language (2 hours)	• Interactive Proofs with the KeY system (2 hours)	• Personal study (5 hours) • Use of KeY (5 hours)			
Week 6 (14 hours)	• Program Logics and its translation to JML (2 hours)	• Interactive verification of Proof Obligations in KeY (2 hours)	• Personal study (5 hours) • Advanced use of KeY (5 hours)			
Week 7 (14 hours)	• Program Transformation. Fold, Unfold techniques (2	• Partial Evaluation tools (2 hours)	• Personal study (5 hours)			

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
	hours)		• Use of partial evaluators (5 hours)			
Week 8 (14 hours)	• Foundations of Partial Evaluation (2 hours)	• Advanced applications of partial evaluation (2 hours)	• Personal study (5 hours) • Advanced use of partial evaluators (5 hours)			

Note: Student workload specified for each activity in hours