ANX-PR/CL/001-01

LEARNING GUIDE

SUBJECT

103000583 - Abstract interpretation

DEGREE PROGRAMME

10AK - Master Universitario en Software y Sistemas

ACADEMIC YEAR & SEMESTER

2017/18 - Semester 2

Index

Learning guide

1. Description

1.1. Subjet details

Name of the subject	103000583 - Abstract interpretation
No of credits	4 ECTS
Type	Optional
Academic year ot the programme	First year
Semester of tuition	Semester 2
Tuition period	February-June
Tuition languages	English
Degree programme	10AK - Master Universitario en Software y Sistemas
Centre	Escuela Tecnica Superior de Ingenieros Informaticos
Academic year	2017-18

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Manuel Carro Li?ares (Subject coordinator)	2304	manuel.carro@upm.es	F - 15:00 - 19:00 Please send an e-mail to set up an appointment before going to the instructor's office.

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

2.3. External faculty

Name and surname	Email	Institution
Pierre Ganty	pierre.ganty@imdea.org	IMDEA Software Institute

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

- Foundations for programming languages

3.2. Other recommended learning outcomes

- Students who wish to take this course should write its coordinator before enrolling to ensure that there are free slots available.

- Understanding of basic mathematical notations and set theory. Functional language and yacc parser. Compiler technologies. General acquaintance with programming and programming languages.

4. Skills and learning outcomes *

4.1. Skills to be learned

CEM1 - Identificar, a partir del estado de la cuestión, la presencia de problemas de investigación relacionados con la concepción, la construcción, el uso y la evaluación de sistemas sociotécnicos complejos que hagan un uso intensivo de software

CEM5 - Aportar soluciones a aquellos problemas abiertos relacionados con el ámbito de aplicación y los métodos, técnicas y herramientas de Verificación y Validación de Software

CG12 - Comprensión amplia de las técnicas y métodos aplicables en una especialización concreta, así como de sus límites

CG13 - Apreciación de los límites del conocimiento actual y de la aplicación práctica de la tecnología más reciente.

CG14 - Conocimiento y comprensión de la informática necesaria para la creación de modelos de información, y de los sistemas y procesos complejos

CG4 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

CG7 - Especificación y realización de tareas informáticas complejas, poco definidas o no familiares

CG8 - Planteamiento y resolución de problemas también en áreas nuevas y emergentes de su disciplina

CG9 - Aplicación de los métodos de resolución de problemas más recientes o innovadores y que puedan implicar el uso de otras disciplinas

CGI20 - Adquirir conocimientos científicos avanzados del campo de la informática que le permitan generar nuevas ideas dentro de una línea de investigación.

CGI23 - Capacidad de leer y comprender publicaciones dentro de su ámbito de estudio/investigación, así como su catalogación y valor científico

4.2. Learning outcomes

RA43 - : Ser capaz de utilizar las herramientas existentes para el Análisis estático de programas, la Verificación formal de programas y la Transformación automática de programas

RA44 - Conocer los fundamentos de la interpretación abstracta como método de análisis estático de programas

* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

Abstract interpretation is a formal mathematical framework which makes it possible to define and implement program analyses which are correct by construction -- provided that some basic properties are met. Software is an essential part of more and more complex systems. In order to master that complexity, it is necessary to be able to compute automatically reliable information about software. This is the goal of static analysis. To reason rigorously about programs, mathematical models of the behavior or programs are developed. Such models are called semantics. Depending on the properties to compute, we can use different semantics - some more detailed for more precision, others coarser for more efficiency. Abstract interpretation is a theory of approximation that allows the comparison of different semantics, or building new semantics through refinement or abstraction. Because all interesting properties of programs are undecidable, all static analysis methods introduce approximations in one way

or another. Therefore, abstract interpretation is a good framework to give a unified vision of such methods and to compare them.

The objective of the course will be to give the basics of abstract interpretation, and to describe some applications.

5.2. Syllabus

1. An informal introduction
2. Program properties
3. Property approximations
4. Morphisms and connections
5. Abstraction of fixpoints
6. Conception of a reachability analysis
7. Applications
8. Approximated fixpoints and widening
9. Refinement of analyses

6. Schedule

6.1. Subject schedule*

Week	Face-to-face classroom activities	Face-to-face laboratory activities	Other face-to-face activities	Assessment activities
1	**Theory and practical work** Duration: 03:45 Problem-solving class			
2	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
3	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
4	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
5	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
6	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
7	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
8	**Theory and practical work** Duration: 03:45 Problem-solving class			**Question and answer session** Other assessment Continuous assessment Duration: 01:00
9				
10				
11				
12				
13				
14				
15				
16				

17				**Presentation of practical work** Individual work Final examination Duration: 00:30

The independent study hours are training activities during which students should spend time on individual study or individual assignments.

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

* The subject schedule is based on a previous theorical planning of the subject plan and might go to through experience some unexpected changes along throughout the academic year.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Continuous assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
2	Question and answer session	Other assessment	Face-to-face	01:00	13.5%	0 / 10	
3	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14
4	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14
5	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14
6	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14
7	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14

8	Question and answer session	Other assessment	Face-to-face	01:00	14.42%	0 / 10	CG8 CEM1 CEM5 CG7 CG4 CG12 CG14

7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Presentation of practical work	Individual work	Face-to-face	00:30	100%	0 / 10	CG4 CG8 CEM1 CEM5 CG7 CG12 CG14

7.1.3. Referred (re-sit) examination

No se ha definido la evaluación extraordinaria.

7.2. Assessment criteria

Students will be evaluated based on the correctness of their practical work and a presentation thereof and on their participation in the classes, represented by the Q&A component. The Q&A sessions have all the same weight; the percentage which appears in the learning guide is a decimal approximation.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Various	Bibliography	Will be given based on the level of the students
O'Caml compiler	Others	Compiler for the programming language to be used in the course

9. Other information

9.1. Other information about the subject

All students wishing to take this course are required to get in touch with the coordinator of the course prior to enrollment in order to verify whether the requirements for the course are met and to ensure that there are available slots for this course. **Please consult** http://software.imdea.org/graduateschool .