DEGREE CURRICULUM

HARDWARE AND SOFTWARE VERIFICATION SYSTEMS

Coordination: BEJAR TORRES, RAMON

Academic year 2017-18

HARDWARE AND SOFTWARE VERIFICATION SYSTEMS 2017-18

Subject's general information

Subject name	HARDWARE AND SOFTWARE VERIFICATION SYSTEMS
Code	102044
Semester	2nd Q(SEMESTER) CONTINUED EVALUATION
Typology	<table><tr><th>Degree</th><th>Course</th><th>Typology</th><th>Modality</th></tr><tr><td>Bachelor's Degree in Computer Engineering</td><td>4</td><td>COMPULSORY</td><td>Attendance-based</td></tr></table>
ECTS credits	6
Groups	1GG
Theoretical credits	3
Practical credits	3
Coordination	BEJAR TORRES, RAMON
Department	INFORMATICA I ENGINYERIA INDUSTRIAL
Teaching load distribution between lectures and independent student work	6 ECTS = 25x6 = 150 - 60 hours of on-class activities - 90 hours of autonomous activities
Important information on data processing	Consult this link for more information.
Language	Spanish / English All the learning material, exercises and homework will be presented in English Classes can be given in English if students need so. Personal attention can also be given in English.
Distribution of credits	3 crèdits teoria 3 crèdits pràctica
Office and hour of attention	The meetings will be arranged with the professor on demand.

HARDWARE AND SOFTWARE VERIFICATION SYSTEMS 2017-18

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
BEJAR TORRES, RAMON	ramon@diei.udl.cat	6	by appoinment via email

Subject's extra information

To successfully address the subject, it is advisable to have previously taken courses with contents about:

- Computational Logic (that is present in many computer engineering university degrees).
- Artificial intelligence (that is present in many computer engineering university degrees).

So, students wanting to study this subject should have finished at least those computer engineering subjects.

Learning objectives

Expected learning outcomes linked to UdL strategic and cross-disciplinary competences :

- To know how to prepare technical documents with different presentation tools for digital documents (CT3).
- To know how to work with technical and scientific documents written in English (CT2).
- To understand the main problems encountered in the design of intelligent systems capable of reasoning and learning and knows how to analize the requirements in the design of these systems (EPS6)**.**

Expected learning outcomes linked to specific competences :

- To know and understand the fundamental problems of the formal specification of the behavior of programs (GII-C5).
- To know the formal specification of programs based on formal languages (GII-C5).
- To know how to use basic tools for semi-automatic verification of software, that need interaction with people to complete tasks that can not be automated 100%, and understand the limitations that the theoretical results on computability imposed on these tools (GII-C5 , GII-C6).

Competences

Strategic:

CT2. Mastering a foreign language, especially English.

CT3. Training Experience in the use of the new technologies and the information and communication technologies.

Cross-disciplinary:

EPS6. Capacity of analysis and synthesis.

Specific:

GII-C5 Capacity to acquire, obtain, formalise and represent the human knowledge in a computable form to solve problems by means of a computer system in any field of application, particularly in the ones related

with computation, perception and performance in environments or intelligent surroundings.

GII-C6. Capacity to develop and evaluate interactive systems and of presentation of complex information and its application to solve problems of design of computer-person interaction.

Subject contents

1. Introduction to the verification of software and hardware

2. Formal verification of algorithms

3. Full Verification

    - Hoare Logic Calculus

    - Verification of While-Do programs

    - Partial versus full verification.

    - Forward verification with status updates

    - Verification of programs with the Key-Hoare tool

4. Partial Verification and Bug Finding

    - Verification systems through Bounded Model Checking (BMC)

    - Verification ANSI-C programs by CBMC tool

    - Hardware Verification by BMC

Methodology

There will be three types of activities:

1) Lectures (50% of the classes).

2) Classes for solving exercises, lab problems and mandatory assignments (50% of the classes).

3) Independent work outside class for finishing exercises and doing the mandatory assignments.

Development plan

Week	Description	Face-to-Face Activity	Autonomous Activity	Hours (F and A)
1	Presentation and introduction to formal software verification	Lectures		**4 -**
2	Formal specification and Symbolic Execution	Lectures and problem solving laboratory	**Solve Exercises**	4 6
3	Formal specification and Symbolic Execution	Lectures and problem solving laboratory	**Study Solve Exercises**	4 7
4	Hoare Logic with states – assignments and conditionals	Lectures and problem solving laboratory	**Solve Exercises**	4 6
5	Hoare Logic with states - loops	Lectures and problem solving laboratory	**Study Solve Exercises**	**4 7**

6	Hoare Logic with states - loops	Lectures and problem solving laboratory	**Solve Exercises**	4 6
7	Hoare Logic with states – loop termination	Lectures and problem solving laboratory	**Work on 1st problem set assignment** **Solve Exercises**	4 8
8	Hoare Logic with states – loop termination	Lectures and problem solving laboratory	**Study** **Work on 1st problem set assignment**	4 8
9	**Evaluation**	**Written Exam on complete verification with Hoare Logic with states**	**Study** **Work on 1st problem set assignment**	4 8
10	Introduction to BMC for digital software and hardware systems	Lectures and programming laboratory	**Solve Exercises**	4 6
11	Static Analysis in CBMC and formula building	Lectures and programming laboratory	**Solve Exercises**	4 6
12	The CBMC verification tool	Lectures and programming laboratory	**Study** **Solve Exercises**	4 6
13	CBMC user assertions	Lectures and programming laboratory	**Study** **Solve Exercises**	4 10
14	CBMC automatic assertions and code coverage	Lectures and programming laboratory	Study **Work on 2nd problem set assignment**	4 8
15	Verification of Verilog Hardware Designs	Lectures and programming laboratory	Study **Work on 2nd problem set assignment**	4 8
16			**Study** **Work on 2nd problem set assignment**	- 6
17	**Evaluation**	**Written Exam on BMC and CBMC**	Study **Work on 2nd problem set assignment**	2 6
18				
19				

Evaluation

Table with evaluation activities

Acr.	Evaluation activity	Weight	Minimum grade	In group	Mandatory	
P1	*Formal Verification Problems (1)*	30%	NO	YES	YES	
P2	*Formal Verification Problems (2)*	30%	NO	YES	YES	
PR	*Optional Exercises*	15%	NO	NO	NO	
E1	*Written Exam (1)*	15%	NO	NO	YES	
E2	*Written Exam (2)*	15%	NO	NO	YES	

FinalGrade = 0,3*P1 + 0,3*P2 + 0,15*PR + 0.15*E1 + 0.15*E2
Observe that the maximum grade is 10.5

Bibliography

All learning material will be provided during the course in the form of slides, lecture notes and manuals of the different programs to be used. There is no convenient text book that can be used to follow the contents of this subject.