Universitat Autònoma de Barcelona

2023/2024

Computational Logic

Code: 106569
ECTS Credits: 6

Degree	Type	Year	Semester
2504392 Artificial Intelligence	FB	1	1

Contact

Name: Maria Pilar Dellunde Clave

Email: pilar.dellunde@uab.cat

Teaching groups languages

You can check it through this link. To consult the language you will need to enter the CODE of the subject. Please note that this information is provisional until 30 November 2023.

Prerequisites

There are no prerequisites.

Objectives and Contextualisation

As a knowledge representation formalism, a reasoning system, an analytical tool, or even a programming language, the function of logic in artificial intelligence (AI) has been prominent since the inception of the discipline. The objective of this course is, therefore, to delve into the role of logic within AI, by providing students with an understanding of its fundamental concepts, techniques, and methods. This will enable them to proficiently apply logic across these varying facets of AI.

Competences

- Analyse and solve problems effectively, generating innovative and creative proposals to achieve objectives.
- Develop critical thinking to analyse alternatives and proposals, both one's own and those of others, in a well-founded and argued manner.
- Identify, understand and apply the fundamental concepts and techniques of knowledge representation, reasoning and computational learning for the solution of artificial intelligence problems.
- Know, understand, use and apply appropriately the mathematical foundations necessary to develop systems for reasoning, learning and data manipulation.
- Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.

- Work independently, with responsibility and initiative, planning and managing time and available resources, and adapting to unforeseen situations.

Learning Outcomes

1. Analyse and solve problems effectively, generating innovative and creative proposals to achieve objectives.
2. Develop critical thinking to analyse alternatives and proposals, both one's own and those of others, in a well-founded and argued manner.
3. Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
4. Understand problem modelling for logic programming languages and how to resolve them using satisfiability-based algorithm.
5. Understand the basic notions and mathematical foundations of classical logical formalisms, automatic reasoning techniques and argumentation in AI.
6. Understand the usefulness of theorem provers for solving problems represented in a logical language.
7. Work independently, with responsibility and initiative, planning and managing time and available resources, and adapting to unforeseen situations.

Content

Introduction and Motivation

The Importance of Logic in AI

Defining Logic

Part I. Propositional Logic (Truth-functional Logic, TFL)

I.1 Syntax of TFL (alphabet, connectives, sentences...).

I.2 Semantics of TFL (truth-functional connectives, characteristic truth tables, complete truth tables, partial truth tables...).

I.3 Natural language formalization in TFL (and its limitations).

I.4 Reasoning in TFL (rules and methods).

I.5 Resolution for TFL (transform formulas into normal forms- as DNF or CNF).

I.6 Introduction to Propositional Logic Programming

Part II. First-Order Logic (FOL)

II.1 Syntax of FOL (quantifiers, formulas, sentences...).

II.2 Semantics of TFL (extensionality, interpretations...).

II.3 Natural language formalization in FOL (and its limitations).

II.4 Resolution for TFL (transform formulas into normal forms).

II.5 Introduction to First-Order Logic Programming.

Methodology

The course methodology is based on short lectures by the professor, problem-solving during class time (specifically, students will engage in individual or group practices to reinforce their learning of the lesson and do evaluative exercises), and flipped learning (that is, students will complete the lectures with readings and work at home). In some classes, time will be kept for reviewing and correcting the evaluative practices.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Exercise in class	30	1.2	2, 6, 3
Introduction and discussion of the main theoretical concepts	12	0.48	5
Type: Supervised			
Assimilation of theoretical concepts	10	0.4	1, 6
Reinforcement and follow-up in the resolution of exercises	12	0.48	2
Type: Autonomous			
Autonomous work and readings	38	1.52	7
Preparing and solving exercises	30	1.2	2, 6, 3, 7

Assessment

The assessment can be carried out in two ways:

Continuous assessment. It is divided into two same-weight parts. On the one hand, students are required to complete, in the classroom, ten practices, including both individual and group assignments. On the other hand, there will be an individual final exam consisting of the content of Parts I and II. Therefore, the course's final grade will be determined based on the performance in the ten practical exercises (50%) and the final exam (50%). To be evaluated with the continuous assessment, the student must have taken at least 7 practicals. Otherwise, the student will not have passed the continuous assessment and if they meet the conditions, they will have the option to present for recovery (see the Recovery section).

Single assessment. It will consist of the written delivery of exercises (with a value of 50% of the final mark), and an exam (with a value in the final mark of 50%) of all the material given in the course.

Recovery: The recovery test is a final exam. To participate in recovery, students must have previously been evaluated in a set of activities whose weight is equivalent to a minimum of 2/3 parts of the total qualification (continuous evaluation) or deliver all the exercises and have done the exam (single assessment).

On carrying out each evaluation activity, lecturers will inform students (on Moodle) of the procedures to be followed for reviewing all grades awarded, and the date on which such a review will take place.

Students will obtain a "No avaluable" course grade unless they have submitted more than 1/3 of the assessment items.

In the event of a student committing any irregularity that may lead to a significant variation in the grade awardedto an assessment activity, the student will be given a zero for this activity, regardless of any disciplinary process that may take place. In the event of several irregularities in assessment activities of the same subject, the student will be given a zero as the final grade for this subject.

In the event that tests or exams cannot be taken onsite, they will be adapted to an online format made available through the UAB's virtual tools (original weighting will be maintained). Homework, activities, and class participation will be carried out through forums, wikis, and/or discussions on Teams, etc. Lecturers will ensure that students are able to access these virtual tools, or will offer them feasible alternatives.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Evaluative exercises	50%	16	0.64	1, 4, 5, 7
Exam	50%	2	0.08	1, 2, 6, 3, 7

Bibliography

P. D. Magnus, *Forallx*, University at Albany. With additions under a Creative Commons License by T. Button, J. R. Loftis, and R.Trueman, 2021, http://forallx.openlogicproject.org/. (Also with additions by P. Dellunde and V. Costa.)

M. Ben-Ari: *Mathematical Logic for Computer Science.* Springer, 2012.

J. van Benthem, H. van Ditmarsch, J. van Eijck, J. Jaspars. *Logic in Action*. Open Course Project, 2016, https://www.logicinaction.org/.

D. Barker-Plummer, J. Barwise, J. Etchemendy. *Language, Proof and Logic*. CSLI Publications, 2011, second edition.

Software

SWI-Prolog