



ID2213 Logic Programming 7.5 credits

Logikprogrammering

This is a translation of the Swedish, legally binding, course syllabus.

Establishment

Course syllabus for ID2213 valid from Autumn 2019

Grading scale

A, B, C, D, E, FX, F

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

- Completed course ID1018 Programming I or DD1393 Software Engineering, or the equivalent.
- Completed course ID1020 or DD1338 Algorithms and Data Structures, or the equivalent.
- Completed course SF1624 or IX1303 Algebra and Geometry, or the equivalent.
- Completed course SF1610 or IX1500 Discrete Mathematics, or the equivalent.

Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

Intended learning outcomes

Having passed the course, the student should be able to:

- explain the basic concepts in logic programming
 1. program syntax for Logic Programming
 2. the equality theory for terms and the unification procedure
 3. the search mechanisms in Prolog
- implement algorithms over trees and lists as logic programs
- explain and use the logical and the most common non-logical control structures in Prolog
- be able to account sketchily for whether and why a certain assignment is suited (or not suited) to be implemented using logic programming techniques
- carry out and present a smaller logic programming project in a group or individually in stated time frames.

For higher grades, the student should also be able to

- use DCG notation for grammars in Prolog
- explain the operational and declarative semantics for logic programs
- design adequate representations of abstract data types such as sets sparse matrices, hash tables in logic programs
- use and explain difference structures (e g d-lists)
- use metaprogramming techniques in logic programming languages.

Course contents

The course presents logic and logic programming for software development. This includes an overview of Horn clauses and their application in knowledge representation and reasoning as well as logic programming in Prolog and its semantics. The course presents algorithms over lists and trees as well as search algorithms in graphs, database programming, recursive programming, non-deterministic programming. Tools for arithmetic in Prolog, structure inspection, metalogic, control mechanisms for search, negation in Prolog, incomplete data structures, parsing by means of DCG. Various more efficient data structures, such as difference structures, are introduced. We show how technologies from functional programming fit in the framework of the logic programming and can be expressed in the programming language Prolog. Finally, the course presents some AI applications, such as simple expert systems, and gives a short overview of current methodological trends.

In this course students will meet the theory and the basic design principles within logic programming while they in for example degree projects, can practise the methods.

Application: puzzles and games, parsing, compilation, formula manipulations, expert systems. Introduction to modifications and extensions of Prolog: parallel Prolog, equation and constraint solving, richer forms of clauses, modules, object orientation and Prolog.

In the project part of the course

- the student identifies and defines a problem-oriented project suited to illustrate the use of logic programming techniques
- the student formulates the project by means of the terminology of logic programming (implementation of a program or analysis of a program/system)
- the student carries out the project in a group or individually within the time frame
- the student presents her/his work with a short report and an oral presentation in a "mini workshop".

Course literature

Sterling and Shapiro, The Art of Prolog, 2nd ed. MIT Press 1994, or other appropriate text.

Examination

- PRO1 - Project, 3.0 credits, grading scale: P, F
- TEN1 - Examination, 4.5 credits, grading scale: A, B, C, D, E, FX, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

If the course is discontinued, students may request to be examined during the following two academic years.

Examination takes place with written or oral examination, oral project presentation as well as assessment of the presentations of fellow students, written project report as well as program code for the project.

In agreement with KTH's coordinator for disabilities, it is the examiner who decides to adapt an examination for students in possess of a valid medical certificate.. The examiner may permit other examination forms at the re-examination of few students

Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.