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 2010

Grading scale

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

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

• ID1018 Programming I 7.5 credits or equivalent course.
• ID1020 Algorithms and Data Structures 7.5 credits or equivalent course.
• SF1624 or IX1303 Algebra and Geometry 7.5 credits or equivalent course.
• SF1610 or IX1500 Discrete Mathematics 7.5 credits or equivalent course.

Language of instruction

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

Intended learning outcomes

In this course the student develops knowledge and skills in constructing and understanding the meaning of programs in the paradigm of logic programming. Logic programming is programming with a special class of logical expressions. In systems that execute logic programs the user poses questions to a database of statements (the program) and the system answers these questions.

The course contributes to the available repertoïre of programming systems for the civilingenjör. In particular this can be of great value to construct prototypes of software, but also full applications can be constructed.

The course is eligible in several competence profiles of the civ-ing (master of science) programmes, e.g. theoretical computer science on the D-program. It is for instance useful in conjunction with the courses in compiler construction or for verification of formal systems (both hardware and software).

After an approved course, the student is expected to be able to:

- explain the basic concepts in logic programming (program syntax for LP including DCG-notation for grammars, the equality theory for terms, the unification procedure
- implement algorithms over trees and lists in logic programming
- explain and use the logical and the most common non-logical control structures in Prolog
- discuss the stronger and weaker sides of logic programming and Prolog
- explain briefly whether and why a certain task is suitable (or not suitable) to be implemented using logic programming techniques
- implement smaller program prototypes as logic programs
- succesfully perform a smaller logic programming project in a group or as an individual within given time frames
- orally and in writing explain the meaning (the operational and the declarative semantics) of logic programs.

It is also desirable that the student can:

- explain the operational and declarative semantics for logic programs
- construct adequate representations of abstract data types, such as sets, sparse matrices, hash tables in logic programs
- use and explain difference structures (d-lists)
- use metaprogramming techniques in logic programming
- judge suggested improvements of the language design including its operational semantics.

Course contents

This is a course on the theory and methods of logic programming. The most widespread logic-based language Prolog is introduced and is used in the course. An overview of application areas is presented as well as an introduction to constraint programming and other extensions.

Horn clause logic and its usage in knowledge representation and reasoning. The pure Prolog programming language and its informal semantics. Theory of pure Prolog. Real Prolog: arithmetics structure inspection, metalogic, cut and negation. Programming techniques in Prolog: database programming, recursive programming, non-deterministic programming, incomplete data structures, parsing with DCGs.

Application examples: puzzles and games, parsing, compilation, formula manipulation, expert systems. Modifications and extensions of Prolog:

parallel Prolog, equation and constraint solving, richer clause forms.

In the project course the students:

- choose a problem oriented project that is suitable to illustrate the use of logic programming techniques
- formulate the project using logic programming terminology (implementation of a program or analysis of a program/system)
- perform the project, in a group or individually, within the time frame
- report their work through a short written report and an oral presentation in a "mini workshop".

Course literature

The Art of Prolog, Leon Sterling och Ehud Shapiro

Upplaga: 2:a upplagan Förlag: MIT Press År: 1990

ISBN: 0-262-19338-8

Kurskompendium och OH-bilder(Sjöland m.fl.)

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.

Other requirements for final grade

Approved written examination (TEN1; 4,5 credits) and one approved project assignment (LAB1; 3,0 credits).

For an approved grade on the exam for the course part (4,5 credits) the student is expected to be able to:

- explain the basic concepts in logic programming (program syntax for LP including DCG-notation for grammars, the equality theory for terms, the unification procedure
- implement algorithms over trees and lists in logic programming
- explain and use the logical and the most common non-logical control structures in Prolog.

For higher grades the student is also expected to be able to:

- explain the operational and declarative semantics for logic programs
- construct adequate representations of abstract data types, such as sets, sparse matrices, hash tables in logic programs
- use and explain difference structures (d-lists)
- use metaprogramming techniques in logic programming
- discuss the stronger and weaker sides of logic programming and Prolog (there are no clear "truths" here, it is the ability to reason clearly that is valued).

In the project course the student is expected to:

- choose a problem oriented project that is suitable to illustrate the use of logic programming techniques
- formulate the project using logic programming terminology (implementation of a program or analysis of a program/system)
- perform the project in a group or individually, within the time frame
- report through a short written report and an oral presentation in a "mini workshop".

The project course (3,0 credits) is evaluated as Pass/Fail. If you submit a project report in time, this may give bonus points on the exam. You can get at most 10 bonus points added to those given for the exam. Note: the bonus points are valid only for the current academic year. The tasks of the exam are worth 50 points.

The grades for the entire course are defined by total points being the sum of the exam points and the bonus points given on
the assignments. At least 25 total points are required to pass the exam. The written exam is evaluated with the grades A-F.

The grades for the number of total points n are as follows:

n >= 45: A
45 > n >= 40: B
40 > n >= 35: C
35 > n >= 30: D
30 > n >= 25: E

25 > n >= 20: Fx
20 > n : F

In case of the grade Fx, completing examination is possible within three month after the original exam. In that case, the course responsible will on demand offer an extra home assignment to be solved by the student within one week. Please note that completing is only allowed if the student did not receive bonus points. If the student including bonus points achieves a total sum in the interval 25 >= n >=20 and without bonus points 20>n, the bonus points are not counted, but the degree F is given.

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.