



FID3006 Compilers and Execution Environments 7.5 credits

Kompilatorer och exekveringsmiljöer

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

Establishment

Grading scale

G

Education cycle

Third cycle

Specific prerequisites

Language of instruction

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

Intended learning outcomes

The overall aim of the course is to provide an understanding of how a programming language is implemented including common theories and recent research results and how these theories and research results are applied. The course will

cover techniques for reading, understanding, translating, improving, and executing programs.

This understanding means that after the course a student should be able to:

- explain the overall structure of a compiler.
- describe regular expressions and finite automata and use them unambiguously for lexical analysis; combine and apply techniques to construct a deterministic finite automaton from regular expressions.
- describe context free grammars and apply them to capture common programming language constructs; describe the basic approach of top-down and bottom-up parsing; construct recursive descent, LL(1), LR(0), and SLR parsers; identify whether grammars can be used with a given parsing technique.
- list and describe tasks carried out during semantic analysis.
- describe the design principles behind intermediate program representations; explain and apply methods for selecting instructions; describe the organization of activation records and identify the impact of different design decisions for activation records on program execution.
- define the liveness of variables and compute liveness information from control flow graphs; define the principles of conservative approximations for analyzing programs; construct interference graphs and perform complete register allocation on them reflecting the design of activation records.
- name key components in an execution environment; describe simple techniques for heap management and garbage collection; identify the amortized costs of different approaches to garbage collection; list common techniques and explain their properties used in virtual execution environments.
- describe common techniques for program optimization; identify loops in programs using dominators; describe and apply techniques for optimizing loops and memory access.
- describe how characteristics of hardware architectures influence compilation of programs; give examples of important characteristics.
- describe and apply current research trends in compilers and execution environments in all areas mentioned above

Course contents

Reading programs: lexical and syntactic analysis. Finite automata, regular expressions, LL- and LR-parsing.

Understanding programs: Semantic analysis, type checking. Scope control, declarations and expressions.

Translating programs: machines and instructions. Intermediate code, instruction selection, stacks and procedure calls.

Executing programs: virtual execution environments and run-time systems. Memory management, garbage collection, loading and linking, just-in-time compilation. Dependencies on hardware architecture.

Improving programs: optimization. Machine independent optimizations (dataflow analysis, strength reduction, ...)

Machine level optimization (register allocation, scheduling, prefetching, power consumption...)

Research area overview: major conferences and journals. Current hot topics and connection to other research areas.

Examination

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.

Approved written examination, approved home assignments, and approved application of current research (in the form of using it for a research paper, report, or project, etc).

Other requirements for final grade

Godkänd skriftlig tentamina, godkända inlämningsuppgifter och godkänd tillämpning av nuvarande forskning (till exempel: användning i en forskningsartikel, forskningsreport, eller forskningsprojekt).

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.