

# Introduction to the Theory of Computation

Programme course

6 credits

Introduction to the Theory of Computation

TDDD65

Valid from: 2017 Spring semester

**Determined by**

Board of Studies for Computer Science  
and Media Technology

**Date determined**

2017-01-25

## Main field of study

Computer Science and Engineering, Computer Science

## Course level

First cycle

## Advancement level

G2X

## Course offered for

- Computer Science, Master's Programme

## Entry requirements

Note: Admission requirements for non-programme students usually also include admission requirements for the programme and threshold requirements for progression within the programme, or corresponding.

## Prerequisites

Basic course in discrete mathematics. Basic course in algorithms and data structures.

## Intended learning outcomes

During this course the students will learn concepts and techniques of formal languages, automata, computability and complexity, relevant for the engineering practice. They are applied, among others, in systems modeling and verification and in compiler construction. They also allow to reason if a computational problem at hand can be solved at all and if it has an efficient solution. After this course, a student will be able to use techniques of:

- formal language definition by accepting automata and by formal grammars,
- transformation of such definitions,
- analysis of decidability of algorithmic problems
- complexity analysis.

and demonstrate knowledge concerning

- the basic rules and regulations for advanced studies and how to apply the ethical code of academia in their own studies,
- the requirements of reports and exams in their education.

## Course content

- Finite automata and regular languages.
- Push-down automata and Context-free languages.
- Turing Machines and Church-Turing Thesis.
- The notion of decidability; important undecidable problems.
- Techniques for checking undecidability.
- Big-Oh-notation, techniques for analysis of algorithms.
- Time complexity classes P, NP.
- NP-completeness and its practical relevance.
- Basics of space complexity.

The course also includes university rules, ethical rules, academic writing and reporting and how to prepare for examination.

## Teaching and working methods

Lectures present the theory and seminars are devoted to problem solving. The course runs over the entire autumn semester.

## Examination

UPG3	Assignments - computation	1 credits	U, G
UPG2	Assignments - academic studies	2 credits	U, G
TEN1	Written examination	3 credits	U, 3, 4, 5

## Grades

Four-grade scale, LiU, U, 3, 4, 5

## Other information

Supplementary courses: Compiler construction

## Department

Institutionen för datavetenskap

## Director of Studies or equivalent

Ahmed Rezine

## Examiner

Christer Bäckström

## Course website and other links

<http://www.ida.liu.se/~TDDD65/>

## Education components

Preliminary scheduled hours: 54 h

Recommended self-study hours: 106 h

## Course literature

### Additional literature

#### Books

Michael Sipser, (2006) *Introduction to the Theory of Computation* 2nd edition

Thomas A. Sudkamp, (2006) *Languages and Machines* 3d edition

## Common rules

Regulations (apply to LiU in its entirety)

The university is a government agency whose operations are regulated by legislation and ordinances, which include the Higher Education Act and the Higher Education Ordinance. In addition to legislation and ordinances, operations are subject to several policy documents. The Linköping University rule book collects currently valid decisions of a regulatory nature taken by the university board, the vice-chancellor and faculty/department boards.

LiU's rule book for education at first-cycle and second-cycle levels is available at [http://stydokument.liu.se/Regelsamling/Innehall/Utbildning\\_pa\\_grund-\\_och\\_avancerad\\_niva](http://stydokument.liu.se/Regelsamling/Innehall/Utbildning_pa_grund-_och_avancerad_niva).