Computational complexity

In computational complexity we study the computational resources needed to solve problems and understand the relation between different types of computation. This course advances the students knowledge of computational complexity, and develop an understanding of fundamental open questions.

Content

• Complexity classes (time, space, nondeterminism)
• Space complexity (Logspace, L vs NL)
• Boolean circuits and nonuniform computation
• Power of randomness (interactive proofs)
• Lower bounds for concrete models of computation: Decision trees, communication protocols, propositional proofs.

Keywords

theoretical computer science
computational complexity

Learning Prerequisites

Recommended courses
Theory of computation (CS-251)
Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:
• Demonstrate an understanding of computational complexity and the P vs NP problem
• Formalize and analyze abstractions of complex scenarios/problems
• Express a good understanding of different concepts of proofs
• Prove statements that are similar to those taught in the course
• Use and understand the role of randomness in computation
• Illustrate a basic understanding of probabilistically checkable proofs and their characterization of the class NP (the PCP-Theorem)
• Explain recent exciting developments in theoretical computer science
• Compare different models of computation

Transversal skills
• Demonstrate the capacity for critical thinking
• Summarize an article or a technical report.

Teaching methods
Lecturing and exercises

Expected student activities
Actively attending lectures and exercise sessions. Also homeworks and exam.

Assessment methods
Three homeworks and final exam

Resources
Bibliography
Stasys Jukna: Boolean Function Complexity, Springer

Ressources en bibliothèque
• Boolean Function Complexity / Stasys
• Computational Complexity: A Modern Approach / Arora

Moodle Link
• https://go.epfl.ch/CS-524