

MATH-683

Fine-grained and parameterized complexity

Eisenbrand Friedrich, Polak Adam Teodor

Cursus	Sem.	Type
Mathematics		Opt.

Language of teaching	English
Credits	4
Session	
Exam	During the semester
Workload	120h
Hours	84
Courses	28
Exercises	28
TP	28
Number of positions	

Frequency

Only this year

Remark

Spring semester - Tuesdays and Thursdays

Summary

The classical distinction between polynomial time solvable and NP-hard problems is often too coarse. This course covers techniques for proving more fine-grained lower and upper bounds on complexity of computational problems.

Content

What makes a computational problem hard to solve efficiently? According to the classical paradigm, problems in P are easy, while NP-complete problems are hard. In practice though this classification is often too coarse. Indeed, even if a problem is NP-complete and a polynomial time algorithm should not be expected, we might still look for the fastest possible exponential time algorithm, or an algorithm which runs fast when some secondary parameter, other than the input size, is small. Besides, for problems in P, it happens that even a quadratic time algorithm is too slow in practice, and we look for a faster one, preferably a linear time one. Without tight complexity lower bounds we can never know if further speedups are possible or if the current algorithms are optimal. In this course we explore a currently vivid research area - fine-grained complexity - which brings tools to prove such lower bounds.

We still have no clue how to prove unconditional lower bounds for running time in reasonably strong computational models, e.g. in the word RAM model. Hence, in fine-grained complexity, we settle for conditional hardness results, proved by reductions - most often from well-studied problems, believed to be hard. Depending on the choice of the starting problem, such reductions usually should not be considered impossibility proofs - certainly, any of the assumed hardness hypotheses may turn out false. However, fine-grained reductions let us pinpoint reasons why solving the target problem faster is difficult, and on the other hand they give us a new perspective on how the source problem can be solved. Besides, studying reductions often leads to better understanding of combinatorial structures involved in computational problems.

In this course we will cover key hypotheses of fine-grained complexity, numerous reductions - often between seemingly distant problems - and algorithmic techniques that enable surprising improvements. The course will be research-oriented, and we will focus on open problems.

A tentative list of topics:

- Parameterized complexity of Vertex Cover, branching, kernelization
- Color coding technique, algorithms for Longest Path and Subset Sum
- Iterative compression technique, Feedback Vertex Set
- Treewidth: definition, dynamic programming, bidimensionality, lower bounds

- (Strong) Exponential Time Hypothesis and Sparsification Lemma
- Orthogonal Vectors, lower bounds for diameter approximation and edit distance
- All Pairs Shortest Paths and problems equivalent under subcubic reductions
- Unweighted APSP, Seidel's and Zwick's algorithms
- Boolean Matrix Multiplication and Approximate Pattern Matching
- 3SUM and Convolution 3SUM equivalence, Triangle Listing lower bounds
- Zero Triangle, reductions from 3SUM and APSP, hardness of Set Disjointness
- (min,+)-convolution, reductions to 3SUM and APSP, equivalence with Knapsack
- (min,max)-product and APSP approximation
- OMv-Hypothesis and lower bounds for dynamic problems

Keywords

Computational complexity, fine-grained reductions, FPT

Resources

Bibliography

Cygan et al., Parameterized Algorithms, Springer, 2015 (selected chapters); various research papers suggested during the course; lecture notes

Ressources en bibliothèque

- [Parameterized Algorithms, / Cygan](#)