TECHNISCHE UNIVERSITÄT DRESDEN

COMPLEXITY THEORY

Lecture 26: Summary and Consultation

Markus Krötzsch
Knowledge-Based Systems

TU Dresden, 29th Jan 2019

Summaries and Outlooks

Summary and Outlook (1)

Things we covered in this course:

- Introduction and Organisation
- Turing Machines and Languages
- Undecidability and Recursion (2)
- Time Complexity and Polynomial Time
- NP and NP Completeness (3)
- Space Complexity: PSpace, L, NL (2)
- Hierarchy Theorems and Gaps (2)
- P vs. NP: Ladner's Theorem
- P vs. NP and Diagonalisation
- Alternation
- The Polynomial Hierarchy
- Questions and Answers
- Circuit Complexity and Parallel Computation (2)
- Probabilistic TMs and Complexity Classes (3)
- Quantum Computing (2)

Things we did not cover here:

- Interactive Proofs
- Approximation
- Cryptography
- More quantum computation
- Derandomisation and Pseudo-Random Numbers
- Counting Complexity / Function Problems
- Average Case Complexity
- Descriptive Complexity
- Parametrised Complexity
- ...

Summary and Outlook (2): Turing Machines and Languages

Turing Machines are a simple model of computation

Recognisable (semi-decidable) = recursively enumerable

Decidable = computable = recursive

Many variants of TMs exist – they normally recognise/decide the same languages

What's next?

- A short look into undecidability
- Recursion and self-referentiality
- Actual complexity classes

Summary and Outlook (3): Undecidability

Busy Beaver is uncomputable

Halting is undecidable (for many reasons)

Orcales and Turing reductions formalise the notion of a “subroutine” and help us to transfer our insights from one problem to another

What’s next?

- Some more undecidability
- Recursion and self-referentiality
- Actual complexity classes

Summary and Outlook (4): Undecidability and Recursion

Most properties related to the computation of TMs are undecidable

Many-one reductions establish a closer relationship between two problems than Turing reductions

There are non-semi-decidable problems

Turing machines can work with their own description

What’s next?

- No lectures next week
- Defining complexity classes
- Time complexity

Summary and Outlook (5): Time Complexity and Polynomial Time

Complexity classes are based on asymptotic resource estimates, further generalised by considering general classes of bounds (e.g., all polynomial functions)

Ignoring constant factors is justified due to Linear Speedup

P is the most common approximation of “efficient”

Polynomial many-one reductions are used to show membership in P

What’s next?

- NP
- Hardness and completeness
- More examples of problems

Summary and Outlook (6): Nondeterministic Polynomial Time

NP can be defined using polynomial-time verifiers or polynomial-time nondeterministic Turing machines

Many problems are easily seen to be in NP

NTM acceptance is not symmetric: coNP as complement class, which is assumed to be unequal to NP

What’s next?

- NP hardness and completeness
- More examples of problems
- Space complexities

Summary and Outlook (7): NP Completeness

NP-complete problems are the hardest in NP

Polynomial runs of NTMs can be described in propositional logic (Cook-Levin)

Clique and **Independent Set** are also NP-complete

What's next?

- More examples of problems
- The limits of NP
- Space complexities

Summary and Outlook (8): NP-Complete Problems

3-Sat and **Hamiltonian Path** are also NP-complete

So are **SubSet Sum** and **Knapsack**, but only if numbers are encoded effiently (pseudo-polynomial time)

There do not seem to be polynomial certificates for coNP instances; and for some problems there seem to be certificates neither for instances nor for non-instances

What's next?

- Space
- Games
- Relating complexity classes

Summary and Outlook (9): Space Complexity

Summing up, we get the following relations:

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSpace} = \mathsf{NPSpace} \subseteq \mathsf{ExpTime} \subseteq \mathsf{NExpTime}$$

We also noted $\mathsf{P} \subseteq \mathsf{coNP} \subseteq \mathsf{PSpace}$.

Open questions:

- Is Savitch's Theorem tight?
- Are there any interesting problems in these space classes?
- We have PSpace = NPSpace = coNPSpace.
  But what about L, NL, and coNL?

$\rightsquigarrow$ the first: nobody knows (YCTBF); the others: see upcoming lectures

Summary and Outlook (10): Polynomial Space

True QBF is PSpace-complete

FOL Model Checking and the related problem of SQL query answering are PSpace-complete

Some games are PSpace-complete

What's next?

- Some more remarks on games
- Logarithmic space
- Complements of space classes

Summary and Outlook (11): Games/Logarithmic Space

Winning board games that don't allow moves to be undone is often PSpace-complete

L is the class of problems solvable using only a fixed number of linearly bound counters and pointers to the input

NL is the corresponding non-deterministic class, but we do not know if L = NL

Summary:

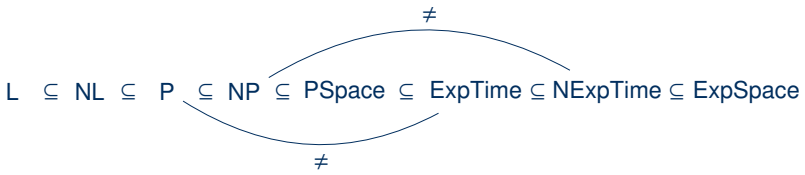
$$\begin{array}{ccccccccccc} \mathsf{L} & \subseteq & \mathsf{NL} & \subseteq & \mathsf{PTime} & \subseteq & \mathsf{NP} & \subseteq & \mathsf{PSpace} & = & \mathsf{NPSpace} \\ \| & & \| & & \| & & ? & & \| & & \| \\ \mathsf{coL} & \subseteq & \mathsf{coNL} & \subseteq & \mathsf{coP} & \subseteq & \mathsf{coNP} & \subseteq & \mathsf{coPSpace} & = & \mathsf{coNPSpace} \end{array}$$

What's next?

- So many $\subseteq$! Will we ever get a strict $\subset$?
- More generally: can more resources solve more problems?

Summary and Outlook (12): Hierarchy Theorems

The time hierarchy theorems tell us that more time leads to more power:



However, they don't help us in comparing different resources and machine types (P vs. NP, or PSpace vs. ExpTime)

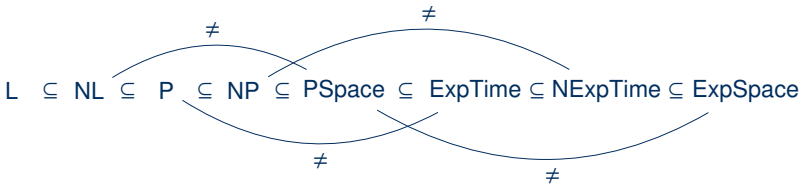
The diagram shows that in sequences such as P $\subseteq$ NP $\subseteq$ PSpace $\subseteq$ ExpTime, one of the inclusions must be proper – but we don't know which (expectation: all!)

What's next?

- The space hierarchy theorem
- Do we need time and space constructibility? What could possibly go wrong ...?
- The limits of diagonalisation, proved by diagonalisation

Summary and Outlook (13): Space Hierarchy and Gaps

Hierarchy theorems tell us that more time/space leads to more power:



However, they don't help us in comparing different resources and machine types (P vs. NP, or PSpace vs. ExpTime)

With non-constructible functions as time/space bounds, arbitrary (constructible or not) boosts in resources do not lead to more power

What's next?

- Computing with oracles (reprise)
- The limits of diagonalisation, proved by diagonalisation
- P vs. NP again

Summary and Outlook (14): P vs. NP: Ladner's Theorem

Ladner's theorem tells us that, in the inuitive case that P $\neq$ NP, there must be (counterintuitively?) many problems in NP that are neither polynomially solvable nor NP-complete

The proof is based on a technique of lazy diagonalisation

What's next?

- Generalising Ladner's Theorem
- Computing with oracles (reprise)
- The limits of diagonalisation, proved by diagonalisation

Summary and Outlook (15): P vs. NP and Diagonalisation

Ladner's theorem can be generalised to find intermediate problems elsewhere

Many results in complexity theory relativise to oracle TMs for some oracle (the same for all TMs considered)

The P vs. NP question does not relativise, as a famous result of Baker, Gill, and Solovay tells us

What's next?

- Generalising NTMs with alternation
- A hierarchy between NP and PSpace

Summary and Outlook (16): Alternation

For $f(n) \geq \log n$, we have shown ASpace(f) = DTime($2^{O(f)}$).

Corollary 16.11 ("Alternating Space = Exponential Deterministic Time"):
AL = P and APSpace = ExpTime.

We can sum up our findings as follows:

$$\begin{array}{ccccccccc} \mathsf{L} & \subseteq & \mathsf{PTime} & \subseteq & \mathsf{PSpace} & \subseteq & \mathsf{ExpTime} & \subseteq & \mathsf{ExpSpace} \\ & & = & & = & & = & & = \\ & & \mathsf{ALogSpace} & \subseteq & \mathsf{APTime} & \subseteq & \mathsf{APSpace} & \subseteq & \mathsf{AExpTime} \end{array}$$

What's next?

- Alternation as a resource that can be bounded
- A hierarchy between NP and PSpace
- End-of-year consultation

Summary and Outlook (17): The Polynomial Hierarchy

The Polynomial Hierarchy is a hierarchy of complexity classes between P and PSpace

It can be defined by stacking NP-oracles on top of P/NP/coNP, or, equivalently, by bounding alternation in polytime ATMs

"Most experts" think that

- The polynomial hierarchy does not collapse completely (same as P ≠ NP)
- The polynomial hierarchy does not collapse on any level
  (in particular PH ≠ PSpace and there is no PH-complete problem)

But there can always be surprises . . .

What's next?

- Some more about the polynomial hierarchy
- End-of-year consultation
- Holidays

Summary and Outlook (18): Questions and Answers

We do not know if the Polynomial Hierarchy is real or collapses

Answer 1: The Logarithmic Hierarchy collapses.

Answer 2: We don't know that NP-hard imples P-hard.

Answer 3: Being outside of P does not make a problem P-hard.

What's next?

- Holidays
- Circuits as an alternative model of computation
- Randomness

Summary and Outlook (19): Circuit Complexity

Circuits provide an alternative model of computation

Nonuniform circuit families are very powerful, and even polynomial circuits can solve undecidable problems

Log-space-uniform polynomial circuits capture P.

Most boolean functions cannot be expressed by polynomial circuits, yet we don't know of any such function that is even in NExp

What's next?

- Circuits for parallelism
- Complexity classes (strictly!) below P
- Randomness

Summary and Outlook (20): Circuits and Parallel Computation

Small-depth circuits can be used to model efficient parallel computation

NC defines a hierarchy of problems below P:

$$AC^0 \subset NC^1 \subseteq L \subseteq NL \subseteq AC^1 \subseteq NC^2 \subseteq \cdots NC \subseteq P$$

P-complete problems, such as Horn logic entailment, are believed not to be efficiently parallelisable.

What's next?

- Randomness
- Summary
- Examinations

Summary and Outlook (21): Probabilistic Turing Machines

Probabilistic TMs can be used to randomness in computation

PP defines a simple "probabilistic" class, but is too powerful in practice.

BPP provides a better definition of practical probabilistic algorithm

What's next?

- More probabilistic classes
- Summary
- Examinations

Summary and Outlook (22): Probabilistic Complexity Classes

BPP provides a robust notion of practical probabilistic algorithm

Polynomial identity testing is in BPP (and not know to be in P)

BPP is different from many other classes in that it has a "semantic" definition based on the behaviour rather than merely the syntax of TMs

What's next?

- More relationships to more (probabilistic) classes
- Summary
- Examinations

Summary and Outlook (23): Probabilistic Complexity Classes

Complexity relationships: see board (or make your own drawing)

Probabilistic classes with ones-sided error – RP and coRP – are common.

ZPP defines random computations with zero-sided error, but probabilistic runtime.

Many experts believe that

$$P = ZPP = RP = coRP = BPP \subsetneq PP$$

What's next?

- Quantum computing
- Summary
- Examinations

Summary and Outlook (24): Quantum Computing (1)

Quantum Mechanics is a highly successful theory of physical reality

At its heart, it is based on probability distributions represented by unit vectors in the Euclidian norm – called (pure) states.

Probabilities can be modified by performing linear, norm-preserving transformations, captured conveniently in unitary matrices.

What's next?

- Quantum Computation proper
- Summary & consultation
- Examinations

Summary and Outlook (25): Quantum Computing (2)

Quantum computing is an exciting alternative theory of computation that might become practice in some future

We know that $P \subseteq BPP \subseteq BQP \subseteq PP \subseteq PSpace$, but little more

There are many further topics on quantum computing not discussed here – algorithms, encryption, error-correction, etc.

What's next?

- Summary & consultation
- Examinations