

# COMPLEXITY THEORY

## Lecture 27: Summary and Consultation

Markus Krötzsch, Stephan Mennicke, Lukas Gerlach  
Knowledge-Based Systems

TU Dresden, 30th Jan 2024

More recent versions of this slide deck might be available.  
For the most current version of this course, see  
[https://iccl.inf.tu-dresden.de/web/Complexity\\_Theory/en](https://iccl.inf.tu-dresden.de/web/Complexity_Theory/en)

# Summaries and Outlooks

# Summary and Outlook (2)

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)

Busy Beaver is uncomputable

Halting is undecidable (for many reasons)

Oracles 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)

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?**

- Defining complexity classes
- Time complexity
- Non-deterministic time

## Summary and Outlook (5)

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)

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-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)

**3-SAT** and **HAMILTONIAN PATH** are also NP-complete

So are **SUBSET SUM** and **KNAPSACK**, but only if numbers are encoded efficiently (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: Relationships of Space and Time (9)

Summing up, we get the following relations:

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

We also noted  $P \subseteq coNP \subseteq PSpace$ .

## 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  $coNL$ ?

↪ the first: **nobody knows** (YCTBF); the others: see upcoming lectures

# Summary and Outlook (10)

**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)

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:

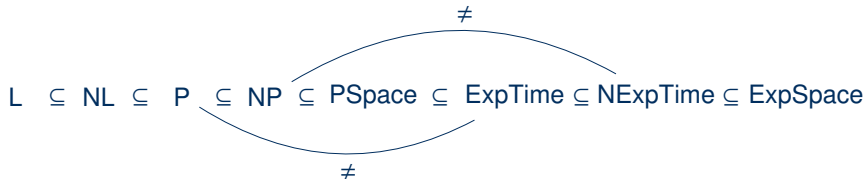
$$\begin{array}{cccccccccccc} L & \subseteq & NL & \subseteq & PTime & \subseteq & NP & \subseteq & PSpace & = & NPSPACE \\ \parallel & & \parallel & & \parallel & & ? & & \parallel & & \parallel \\ coL & \subseteq & coNL & \subseteq & coP & \subseteq & coNP & \subseteq & coPSpace & = & coNPSPACE \end{array}$$

## 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)

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)

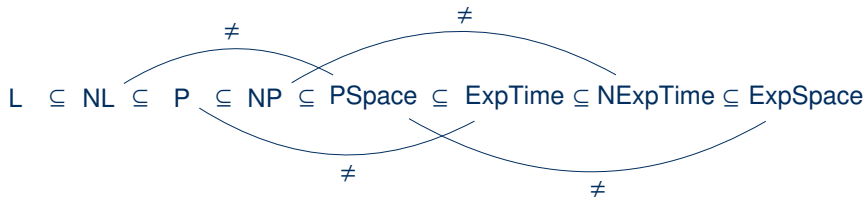
The diagram shows that in sequences such as  $P \subseteq NP \subseteq PSpace \subseteq ExpTime$ , one of the inclusions must be proper – but we don't know which (expectation: all!)

### 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)

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?

- The inner structure of NP revisited
- Computing with oracles (reprise)
- The limits of diagonalisation, proved by diagonalisation

# Summary and Outlook (14)

Ladner's theorem tells us that, in the intuitive 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)

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)

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

**Corollary 27.1 (“Alternating Space = Exponential Deterministic Time”):**  $\text{AL} = \text{P}$  and  $\text{APSpace} = \text{ExpTime}$ .

We can sum up our findings as follows:

$$\begin{array}{ccccccc} \text{L} & \subseteq & \text{PTime} & \subseteq & \text{PSpace} & \subseteq & \text{ExpTime} & \subseteq & \text{ExpSpace} \\ & & \parallel & & \parallel & & \parallel & & \parallel \\ & & \text{ALogSpace} & \subseteq & \text{APTime} & \subseteq & \text{APSpace} & \subseteq & \text{AExpTime} \end{array}$$

### 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** 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

The typical complete problems for the classes in the polynomial hierarchy are QBF with bounded forms of quantifier alternation

## What's next?

- Some more about the polynomial hierarchy
- End-of-year consultation
- Computing with circuits

# Summary and Outlook (18)

Circuits provide an alternative model of computation

$$P \subseteq P_{/poly}$$

**CIRCUIT-SAT** is NP-complete.

$P_{/poly}$  is very powerful – uniform circuit families help to restrict it

## What's next?

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

# Summary and Outlook (19)

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

NC defines a hierarchy of problems below P:

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

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

## What's next?

- Randomness
- Quantum Computing
- Examinations

# Summary and Outlook (21)

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
- Quantum Computing
- Examinations

# Summary and Outlook (22)

BPP provides a **robust** notion of practical probabilistic algorithm

**Polynomial identity testing** is in BPP (and not known 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
- Quantum Computing
- Summary

# Summary and Outlook (23)

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

$$P = ZPP = RP = \text{coRP} = \text{BPP} \subseteq PP$$

## What's next?

- Quantum computing
- Summary
- Examinations

# Summary and Outlook (24)

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
- Interactive Proof Systems
- Summary & consultation



# Summary and Outlook (25)

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?

- Interactive Proof Systems
- Summary & consultation
- Examinations

# Summary and Outlook (26)

Interactive proofs enable probabilistic machines to solve problems beyond NP

**GRAPH NON-ISOMORPHISM** has an interesting interactive zero-knowledge proof protocol

$IP = PSpace$

## What's next?

- Summary & consultation
- Examinations