

## Summary: Lecture 6

Summary for the chapter 8.2. [1, 7]

### Completeness

Let  $C$  be a complexity class and let  $L$  be a language in  $C$ .  $L$  is called  $C$ -complete if any language  $L' \in C$  can be reduced to  $L$ .

(Every language of a complexity class can be reduced to  $L$ .)

- reducibility is transitive  $\rightarrow$  problems are ordered by difficulty
- complete problems can capture the difficulty of a class
- problem is seen as completely understood if the problem is complete

### Question:

Which problems can be reduced to a formal language?

SAT can be expressed as formal language. [6]

$\Rightarrow$  SAT can be reduced to a formal language. (?)

SAT is in NP. [7]

Because CIRCUIT SAT can be reduced to SAT: CIRCUIT SAT can be reduced to a formal language. (?) CIRCUIT SAT is NP-complete. [3]

### Formal language

Formal languages are abstract languages, which define the syntax of the words that get accepted by that language. It is a set of words that get accepted by the language and has a set of symbols that is called alphabet, which contains all the possible characters of the words. Those characters are called nonterminal symbols. [2, 8]

#### Kleene star

The Kleene star  $\Sigma^*$  of an alphabet  $\Sigma$  is the set of all words that can be created through concatenation of the symbols of the alphabet  $\Sigma$ . The empty word  $\epsilon$  is included.

#### Formal language

A formal language  $L$  over an alphabet  $\Sigma$  is a subset of the Kleene star of the alphabet:  $L \subseteq \Sigma^*$

Where to set the line between language decisions and other problems? Can every problem be constructed as a formal language?

Is everything that is reducible to SAT reducible to a formal language because of the transitivity?

I assume it does not have an influence on the complexity of a problem if it can be expressed as a formal language? Are formal languages part of specific complexity classes?

### Closed under reduction

The following complexity classes are all closed under reductions:

P   NP   coNP   L   NL   PSPACE   EXP

A class  $C$  is closed under reductions if whenever  $L$  is reducible to  $L'$  and  $L' \in C'$ , then  $L \in C'$ .

If a complete problem in  $C$  belongs in a class  $C' \subseteq C$ ,  $C = C'$ .

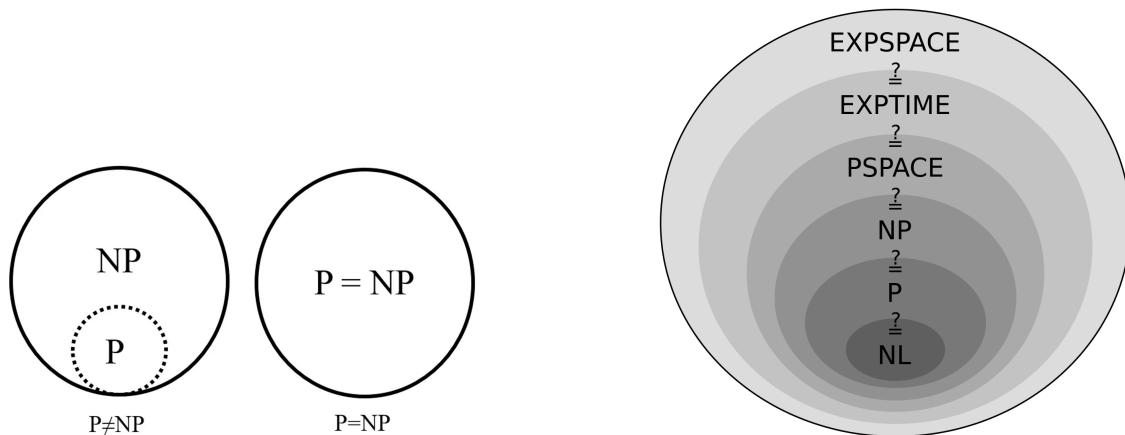


Figure 1: P and NP sets [5] and complexity classes [4]

- examples:
  - if an NP-complete language is in P, then  $NP = P$
  - if a P-complete language is in L, then  $P = L$
  - if a P-complete language is in NL, then  $P = NL$
  - no EXP-complete language can be in P

### P-completeness of CIRCUIT VALUE

#### Problem: Circuit Value

The CIRCUIT VALUE Problem is the problem of computing the output of a given Boolean circuit on a given input.

In terms of time complexity, it can be solved in linear time (topological sort).

- P-complete

#### Proof idea:

- CIRCUIT VALUE is in P (prerequisite for being P-complete)
- show: any language  $L \in P$  can be reduced to CIRCUIT VALUE

TODO

## The reduction (?)

### Problem: Circuit Sat

The circuit satisfiability problem (CIRCUIT SAT) is the decision problem of determining whether a given Boolean circuit has an assignment of its inputs that makes the output true.

Input: a Boolean circuit  $C$

Question: Is there a truth assignment which makes  $C$  output the value true?

## CIRCUIT SAT is NP-complete

- circuit decides nondeterministically (?)
- a variable is added in the nondeterministic Turing Machine
- check if one of the variables is true: use this choice (?)
- problem: can we set these variables such that the Turing Machine accepts?
- answer corresponds directly to *is there a choice of decisions such that the Turing machine accepts?*
- extremely direct reduction
- Cook's theorem :)
- SAT is NP-complete

TODO

Questions:

## References

- [1] Martin Berglund. *Lecture notes in Computational Complexity*.
- [2] *Chomsky's Normal Form (CNF)*. Website. <https://www.javatpoint.com/automata-chomskys-normal-form>, opened on 26.09.2022.
- [3] *Circuit satisfiability problem – Proof of NP-Completeness*. Website. [https://en.wikipedia.org/wiki/Circuit\\_satisfiability\\_problem](https://en.wikipedia.org/wiki/Circuit_satisfiability_problem), opened on 25.11.2022.
- [4] *Complexity classes diagram image source*. [https://en.wikipedia.org/wiki/Complexity\\_class](https://en.wikipedia.org/wiki/Complexity_class).
- [5] *Image source: P-NP sets*. <https://www.techno-science.net/actualite/np-conjecture-000-000-partie-denouee-N21607.html>.
- [6] klaus-joern Lange. “The Boolean Formula Value Problem as Formal Language”. In: (Jan. 2012). DOI: 10.1007/978-3-642-31644-9\_9.
- [7] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.
- [8] A.J. Kfoury Robert N. Moll Michael A. Arbib. *An Introduction to Formal Language Theory*. Springer-Verlag, 1988.