CoTi - Exercise Session

Atien
9.10.'23

My Polybox for notes and recordings:

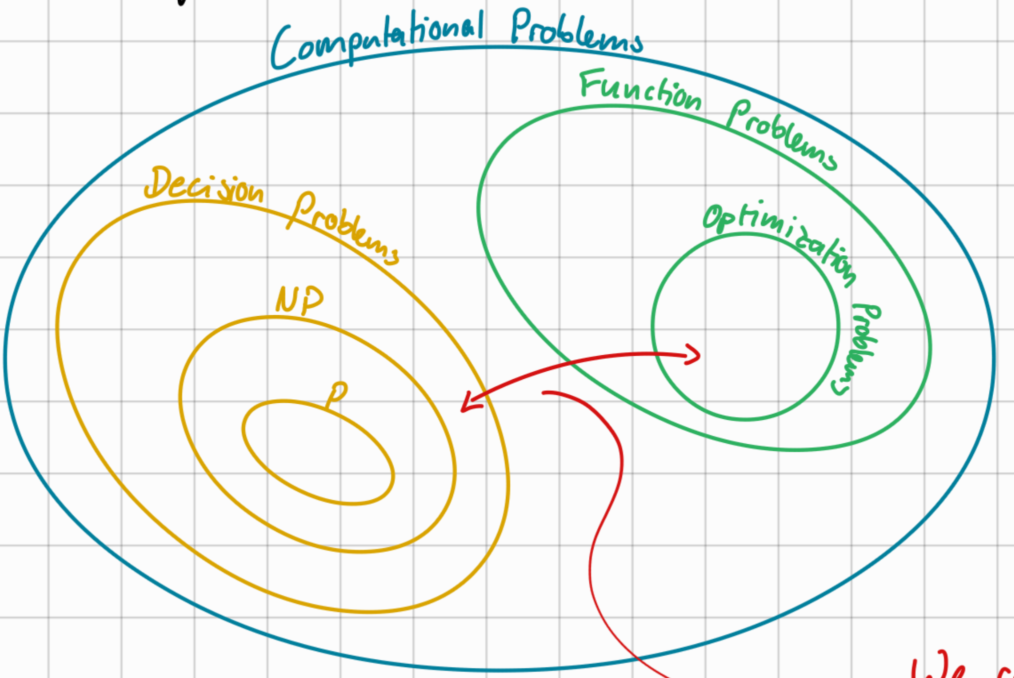
Now also on moodle!

Complexity 1

- P & NP
- Min. Vertex Cover, Max. Independant Set, Max. Clique
- NP-hard
- Boolean Formulas & SAT
- Circuit Complexity

P & NP

Problem Types:



We can find corresponding decision problems to all optimization problems

e.g.: What is the shortest path between ETH HG and ETH ETZ?

⇕

Is there a path between HG and ETZ that is shorter than k meters?

P: Decision problems that can be solved in polynomial time

NP: Decision problems that's solutions are verifiable in polynomial time

How to show a problem is in P?

Way 1) Provide an algorithm running in polynomial time and solving the problem

Way 2) Polynomial Reduction:

If we can show that we can solve our problem A by solving another problem B polynomial times:

$$B \text{ is in } P \Rightarrow A \text{ is in } P$$

We the write:

$$A \leq B$$

↑ B is at least as complex as A

Example: Given multiple ranking lists ranking different qualities of ETH-Departments

Papers/Person	Beer Consumption/Person	Superiority to other departments
1. D-MATH	1. D-MAVT	1. D-ITET
2. D-PHYS	2. D-ITET	⋮
3. ⋮	⋮	D-GESS

And B tells us if D-ITET is in the top 10 of a list in polynomial time.

Problem A: Is D-ITET in the top 10 of at least k of the rankings?

Solution:

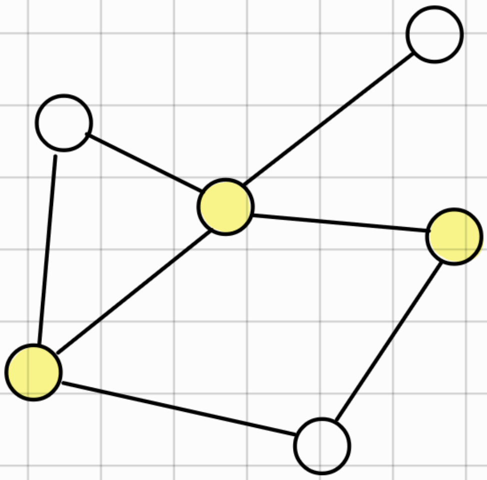
```
A(rankings, k):
    counter = 0
    for r in rankings:
        if B(r):
            counter += 1
    return counter ≥ k
```

→ B will be invoked linearly often

$\Rightarrow \quad A \leq B$

$\Rightarrow \quad A \text{ in } P$

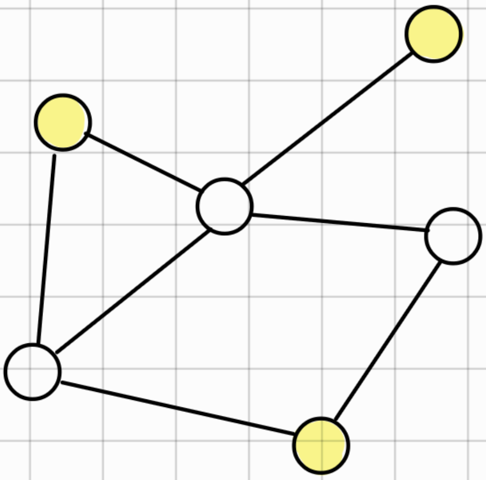
(Minimum) Vertex Cover



The (minimal) set of vertices/nodes so that every edge of the graph is attached to the set.

We can verify in polynomial time if a set of vertices is a vertex cover (e.g. iterate over all edges)
=> in NP

(Maximum) Independant Set

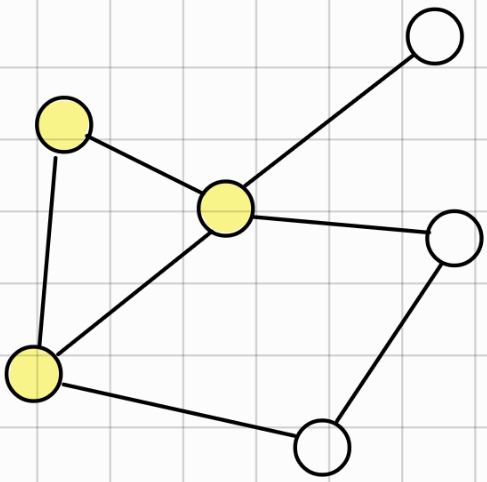


The (maximum) set of vertices/nodes so that no edge of the graph links two vertices of the set

The IS-Problem is also in NP

Note: The VC-Problem is the inverse problem to the IS-Problem!

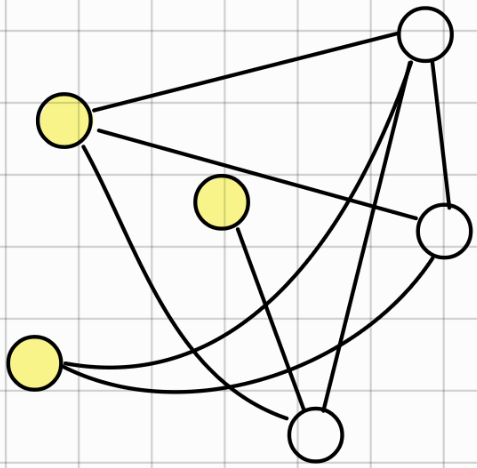
(Maximum) Clique



The (maximum) set of vertices/nodes so that every vertex is linked pair wise to every other vertex of the set by an edge of the graph

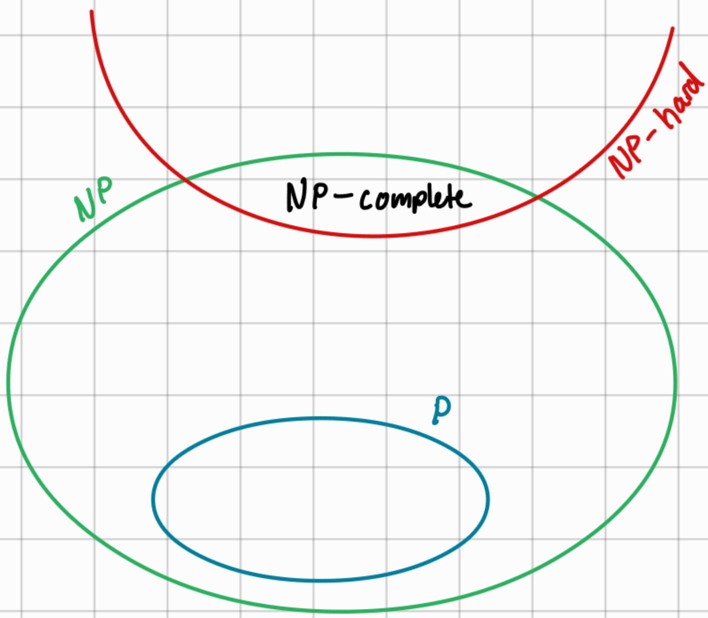
Also in NP

And we can use the complement graph to transform this problem to the IS-Problem:



Note: For those three problems there is no known algorithm to put them in P :(

NP-hard



Any NP problem $\leq$ Any NP-hard problem

NP-hard problem solve all NP problems

It could be that P = NP (We do not know)

Boolean Formulas & SAT

We have variables $x_1, x_2, \dots, x_n \in \{True, False\}$
or $x_1, x_2, \dots, x_n \in \{0, 1\}$

A boolean formula joins them using boolean operators:

or ⇐> $\vee$
and ⇐> $\wedge$
not ⇐> $\neg$

e.g.:

$$(\neg x \vee y) \wedge (\neg x \vee \neg y) \wedge (x \vee y)$$

Clause

Clauses joined with and's
↳ Conjunctive Normal Form (CNF)

$$(\dots) \vee (\dots) \vee (\dots)$$

Clauses joined with or's
↳ Disjunctive Normal Form (DNF)

SAT: Finding a satisfying assignment of variable values for a boolean formula

NP-complete

The number of variables in the clauses corresponds to the order of the SAT-Problem

2-SAT: $(\neg x \lor y) \land (\neg x \lor \neg z)$

3-SAT: $(x \lor y \lor z) \land (\neg x \lor y \lor z)$

SAT $\leq$ 3-SAT

so: 4-SAT $(x \lor y \lor z \lor w)$ => 3-SAT $(x \lor y \lor h) \land (z \lor w \lor \neg h)$

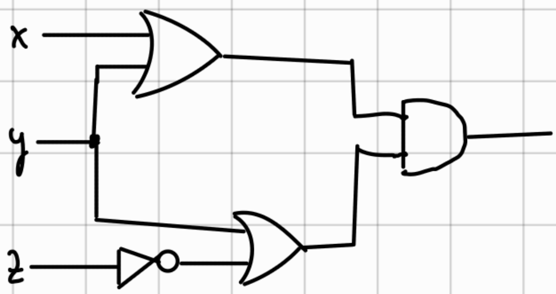
helper variable (no real use)

2-SAT $(x \lor y)$ => 3-SAT $(x \lor y \lor h) \land (x \lor y \lor \neg h)$

Circuit-SAT:

We can use logic gates to model a boolean formula.

$(x \lor y) \land (y \lor \neg z)$



SAT ← equivalent → Circuit-SAT

Minimum Circuit Size Problem:

↳ in NP

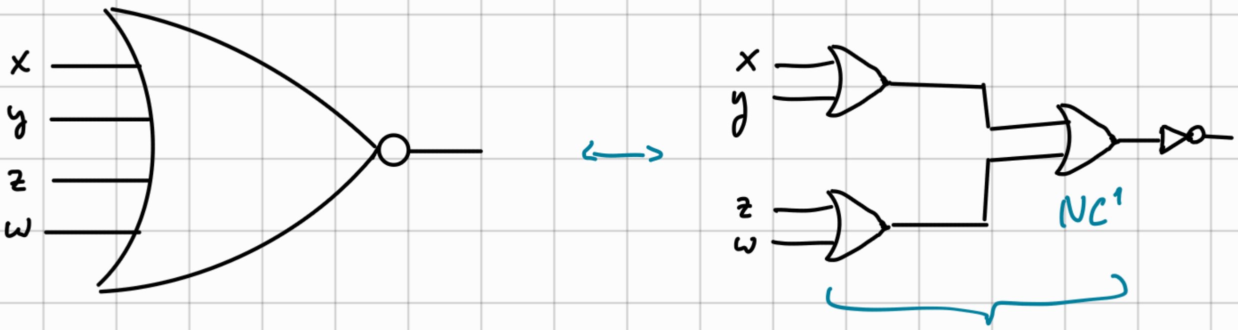
Given a boolean formula what is the minimum number of logic gates needed to implement it?

But we do not know if it is NP-hard :(

Circuit Complexity

We want to have low depth for our circuits

↑ more depth more delay



Depth of the circuit is logarithmic with respect to the inputs

Classes:

$\dots^1$ means the depth is Logarithmic

$NC^0 \subseteq NC^1 \subseteq NC^2 \subseteq \dots$

NC: logic gates have max. 2 inputs

$AC^0 \subseteq AC^0 \subseteq AC^2 \subseteq \dots$

AC: logic gates can have more than 2 inputs

$\dots^0$ means the depth is constant

$\dots^i$ means the depth is in $O(\log(n)^i)$, $n :=$ #inputs