CMSC 28100

Introduction to Complexity Theory

Autumn 2025
Instructor: William Hoza

Coping with intractability

Approximation algorithm for Knapsack

- For every $w_1, \dots, w_k, v_1, \dots, v_k, W$, define

$$\text{OPT} = \max\left\{\sum_{i \in S} v_i : S \subseteq \{1, \dots, k\} \text{ and } \sum_{i \in S} w_i \leq W\right\}$$

Theorem: For every $\epsilon > 0$, there exists a poly-time algorithm such that given $w_1, \dots, w_k, v_1, \dots, v_k, W$, the algorithm outputs $S \subseteq \{1, \dots, k\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq (1 - \epsilon) \cdot \text{OPT}$

Approximation algorithm for Knapsack

- **Algorithm:** Let $v_i' = \lfloor \alpha v_i \rfloor$, where $\alpha = \frac{k}{\epsilon \cdot \max(v_1, \ldots, v_k)}$, so $v_i' \leq k/\epsilon$
- Output $S \subseteq \{1, \ldots, k\}$ that maximizes $\sum_{i \in S} v_i'$ subject to $\sum_{i \in S} w_i \leq W$
  - Polynomial time, because we can encode v_i' in unary
- **Correctness proof:** Let $S' \subseteq \{1, \ldots, k\}$ be optimal. Then

$$\sum_{i \in S} v_i \geq \frac{1}{\alpha} \sum_{i \in S} v_i' \geq \frac{1}{\alpha} \sum_{i \in S'} v_i' > \frac{1}{\alpha} \sum_{i \in S'} (\alpha v_i - 1) \geq \left(\sum_{i \in S'} v_i \right) - \frac{k}{\alpha} = \mathrm{OPT} - \epsilon \cdot \max(v_1, \ldots, v_k)$$

$$\geq (1 - \epsilon) \cdot \mathrm{OPT}$$

Approximation algorithms are not a panacea

- In some cases, approximation algorithms take some of the sting out of NP-completeness
- However:
  - Approximation is not always applicable
    - E.g., 3-COLORABLE is simply not an optimization problem
  - Even if it's applicable, approximation is not always feasible!

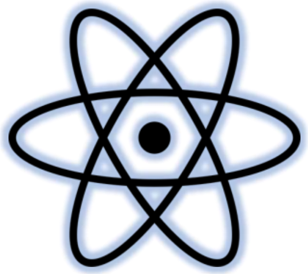
Inapproximability of the clique problem

- For a graph G, let $\omega(G)$ denote the size of the largest clique in G

Theorem: Let $\epsilon > 0$. Suppose there exists a poly-time algorithm such that given a graph $G = (V, E)$, the algorithm outputs a clique $S \subseteq V$ satisfying $|S| \geq \epsilon \cdot \omega(G)$. Then $\mathrm{P} = \mathrm{NP}$.

- (Proof omitted. Not on exercises / exams)

Quantum computing



- Another approach for coping with intractability: Quantum Computing
- A quantum computer is a computational device that uses special features of quantum physics
- A detailed discussion of quantum computing is outside the scope of this course
- We will discuss only some key facts about quantum computing

Quantum computing

- So far, researchers have constructed rudimentary quantum computers
- There are huge ongoing efforts to build fully-functional quantum computers

Quantum complexity theory

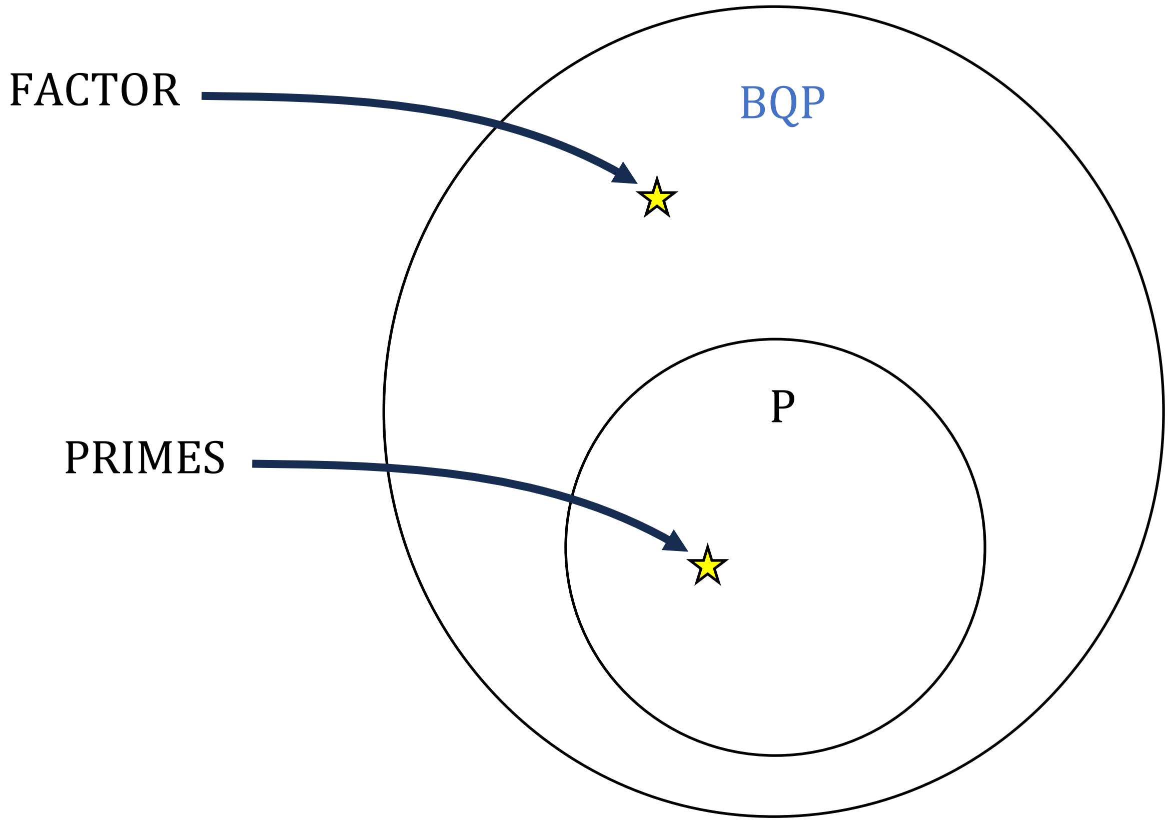
- One can define a complexity class, BQP, consisting of all languages that could be decided in polynomial time by a fully-functional quantum computer
- The mathematical definition of BQP is beyond the scope of this course
- One can prove that $\mathrm{BPP} \subseteq \mathrm{BQP} \subseteq \mathrm{PSPACE}$

Shor's algorithm

- $\mathrm{FACTOR} = \{\langle N, K \rangle : N \text{ has a prime factor } p \leq K\}$
- **Conjecture:** $\mathrm{FACTOR} \notin \mathrm{P}$

Theorem (Shor's algorithm): $\mathrm{FACTOR} \in \mathrm{BQP}$

- FACTOR is a likely counterexample to the extended Church-Turing thesis!

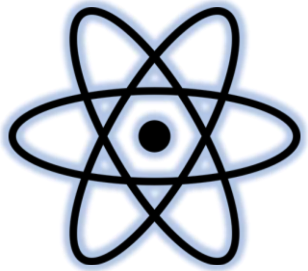


- FACTOR $= \{\langle N, K\rangle : N$ has a prime factor $p \leq K\}$

- PRIMES $= \{\langle K\rangle : K$ is a prime number$\}$

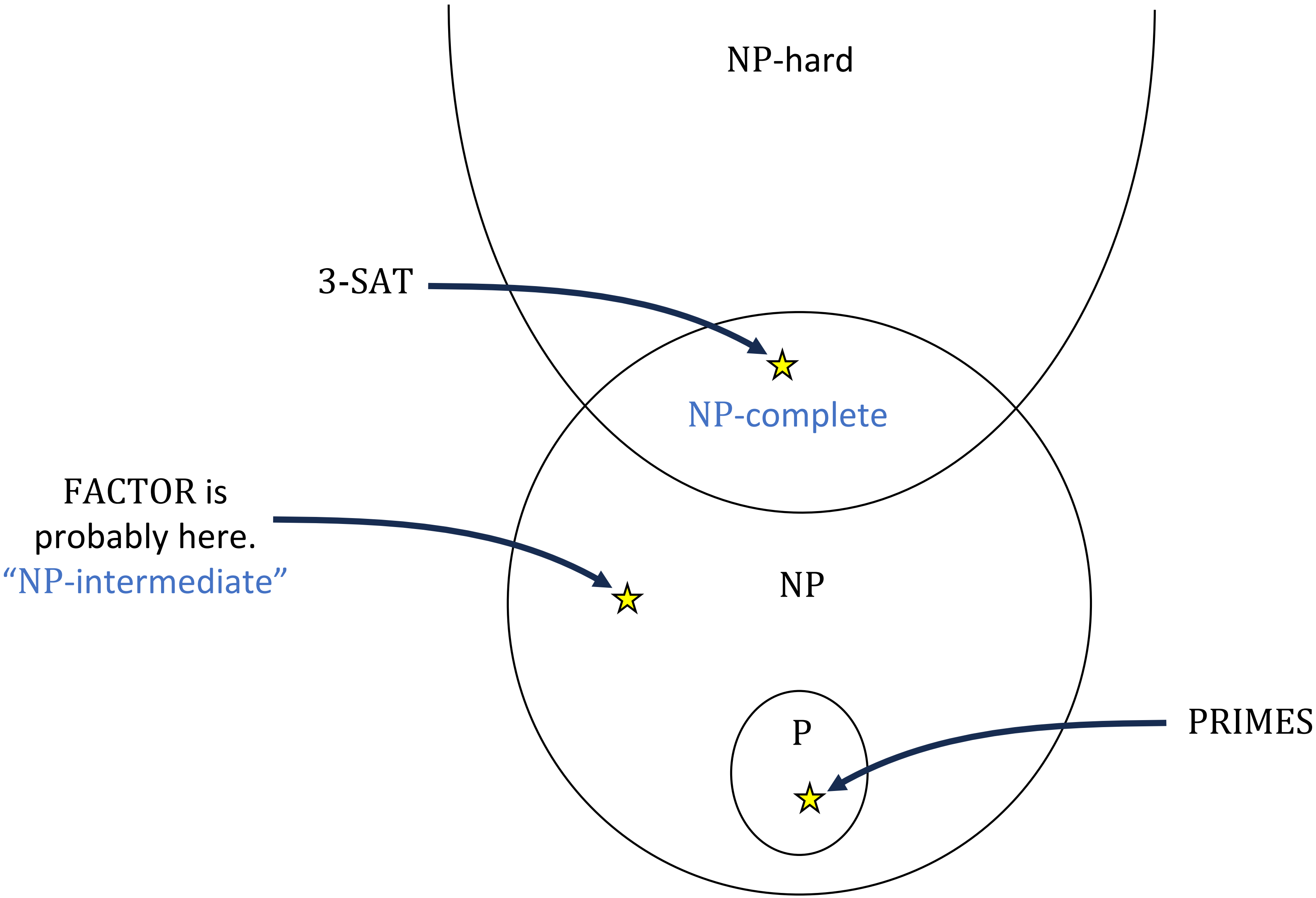
Quantum computing and $\mathbf{NP}$-completeness

- $\text{FACTOR} = \{\langle N, K\rangle : N \text{ has a prime factor } p \leq K\}$
- $\text{FACTOR} \in \text{NP}$ (guess the factor)
- Is FACTOR NP-complete?
- If yes, then $\text{NP} \subseteq \text{BQP}$, meaning that all NP-complete problems could be solved in polynomial time on a fully-functional quantum computer! 😲



Complexity of factoring integers

- Typically, when we encounter some $Y \in \mathrm{NP}$, either
  - we can prove $Y \in \mathrm{P}$, or
  - we can prove that Y is NP-complete
- FACTOR is one of the rare exceptions to this rule
- **Conjecture:** FACTOR is neither in P nor NP-complete!


NP-hard
3-SAT
NP-complete
FACTOR is
probably here.
“NP-intermediate”
NP
P
PRIMES

Complexity of factoring integers

- What evidence suggests that FACTOR is not NP-complete?
- Key: The complexity class coNP
- Informal definition: coNP is like NP, except that we swap the roles of “yes” and “no”

The complexity class $\mathbf{coNP}$

- Let $Y \subseteq \{0,1\}^*$
- **Definition:** $Y \in \mathrm{coNP}$ if there exists a randomized polynomial-time Turing machine M such that for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ rejects } w] = 0$
  - If $w \notin Y$, then $\Pr[M \text{ rejects } w] \neq 0$

The complexity class $\mathbf{coNP}$

- Let $Y \subseteq \{0,1\}^*$ and let $\bar{Y} = \{0,1\}^* \setminus Y$
- **Fact:** $Y \in \mathrm{NP}$ if and only if $\bar{Y} \in \mathrm{coNP}$
- coNP is the set of complements of languages in NP

What is coP?

A: The set of languages that are not in P

B: $\mathrm{coP} = \mathrm{P}$

C: The set of algorithms that do not run in polynomial time

D: The notion of "coP" doesn't make any sense

Respond at PollEv.com/whoza or text "whoza" to 22333

The complexity class $\mathbf{coNP}$

- Example: A Boolean formula is unsatisfiable if it is not satisfiable
- Let $\text{3-UNSAT} = \{\langle \phi \rangle : \phi \text{ is an unsatisfiable 3-CNF formula}\}$
- Then $\text{3-UNSAT} \in \text{coNP}$, because a satisfying assignment is a certificate showing that $\langle \phi \rangle \notin \text{3-UNSAT}$

FACTOR $\in$ coNP

- $\text{FACTOR} = \{\langle K, R\rangle : K \text{ has a prime factor } p \text{ such that } p \leq R\}$
- **Claim:** $\text{FACTOR} \in \text{coNP}$
- **Proof:** Given $\langle K, R\rangle$:
  - Nondeterministically guess numbers $d \leq \log K$ and $p_1, p_2, \ldots, p_d \leq K$
  - If $p_1, \ldots, p_d$ are prime, $p_1 \cdot p_2 \cdot p_3 \cdots p_d = K$, and $\min(p_1, \ldots, p_d) > R$, reject
  - Otherwise, accept

PRIMES $\in$ P

The complexity class $\mathrm{NP} \cap \mathrm{coNP}$

- We have shown that $\mathrm{FACTOR} \in \mathrm{NP}$ and $\mathrm{FACTOR} \in \mathrm{coNP}$
- $\mathrm{FACTOR} \in \mathrm{NP} \cap \mathrm{coNP}$
- $Y \in \mathrm{NP} \cap \mathrm{coNP}$ means that for every instance, there is a certificate
  - A certificate of membership for YES instances
  - A certificate of non-membership for NO instances