Seminar Theoretical Computer science. April 2nd, 2018.

If L is a language over an alphabet Σ, then L^{c} represents the complement $\Sigma^* \setminus L$. Let $\text{co}\mathsf{NP} = \{L^{\mathsf{c}} \colon L \in \mathsf{NP}\}$.

The class coNP

Exercise 1. Prove that if $\mathsf{NP} \neq \text{co}\mathsf{NP}$ then $\mathsf{P} \neq \mathsf{NP}$.

Exercise 2. Show that if $\mathsf{NP} \subseteq \text{co}\mathsf{NP}$ then $\mathsf{NP} = \text{co}\mathsf{NP}$.

Exercise 3. Suppose $L_1, L_2 \in \mathsf{NP} \cap \text{co}\mathsf{NP}$. Show that $L_1 \oplus L_2 = \{x : x \text{ is in exactly one of } L_1, L_2\}$ is in $\mathsf{NP} \cap \text{co}\mathsf{NP}$.

Exercise 4. We say that a language L is coNP-complete if (1) $L \in \text{co}\mathsf{NP}$ and (2) $A \leq_p L$ for all $A \in \text{co}\mathsf{NP}$. Show that 3SAT^{c} is coNP-complete.

The class PSPACE

Exercise 5. Show that if $A \in \mathsf{PSPACE}$, then $A^* \in \mathsf{PSPACE}$.

Exercise 6. Two Boolean formulas are *equivalent* if they describe the same Boolean function. For example, the formulas $x_1 \wedge \overline{x}_2$ and $x_1 \wedge x_1 \wedge \overline{x}_2$ are equivalent. A Boolean formula is *minimal* if there exists no equivalent formula for which the number of $\wedge$ and $\vee$ symbols is strictly smaller. For example, the formula $x_1 \wedge x_1 \wedge \overline{x}_2$ is not minimal, but $x_1 \wedge \overline{x}_2$ is. Let MinFormula be the set of minimal Boolean formulas.

- Show that MinFormula $\in \mathsf{PSPACE}$.
- Explain why this argument fails to show that MinFormula $\in \text{co}\mathsf{NP}$: If $\phi \notin$ MinFormula, then ϕ has a smaller equivalent formula. A non-deterministic Turing machine can verify that $\phi \in \overline{\text{MinFormula}}$.

Exercise 7. Show that if $\text{TQBF} \in \mathsf{NP}$, then $\mathsf{NP} = \text{co}\mathsf{NP}$.

Exercise 8. The cat-and-mouse game is played by two players "Cat" and "Mouse" on an arbitrary graph. At a given moment of time, each player occupies a vertex of a connected graph. Players move in turns. At his turn, a player must move to a vertex that is adjacent to the current position.

A special vertex of the graph is called Hole. Cat wins if the two players ever occupy the same node. Mouse wins if it reaches the Hole before the preceding happens. The game is draw if a situation repeats (a situation is determined by the player's positions and the player's turn to move).

The problem Happy Cat:

Input:	A graph (V, E), vertices $c, m, h \in V$ (initial positions of Cat, Mouse and the Hole).
Question:	Does Cat have a winning strategy if Cat moves first?

Prove that this problem is in P.

PSPACE-completeness

Exercise 9. The problem in-space acceptance:

Input:	A Turing machine M and an input x.
Question:	Does M accept x without ever leaving the first $|x| + 1$ cells on the tape?

Prove that the problem in-space acceptance is in PSPACE. Prove that the problem is PSPACE-complete.

Exercise 10. Consider TQBF problem restricted to monotone formulas (no negations). Show (at least) one of the two: monotone-TQBF is in P; monotone-TQBF is PSPACE-complete.

Exercise 11. In this problem a graph (V, E) is given by a Boolean circuit ϕ. This means that V equals the set of m-bit strings for some m, and that two vertices x and y are connected if and only if $\phi(x, y) = 1$. Prove that the following problem is PSPACE-complete.

The problem succinct connectivity:

Input:	A graph G represented by a circuit and two nodes s and t.
Question:	Is there a path from s to t in G?