

# Exercises Complexity Theory

Lecture 4

May 6, 2024

**Only the exercises where points are given can be handed in.**  
(The maximum number of points per exercise is written in the margin.)

To be handed in on **May 21, 2024**, in Brightspace under Assignment 4, **deadline: 13:30**.

**Exercise 1.** Do the following statements hold? In each case, give a proof or a counterexample.

- (10) (a) If  $A \in \mathbf{P}$  and  $B \in \mathbf{P}$ , then  $A \cdot B \in \mathbf{P}$ .
- (10) (b) If  $A \in \mathbf{P}$ , then  $A^* \in \mathbf{NP}$  (where  $A^* = \{w_1 \cdots w_n \mid n \in \mathbb{N}, \forall i. w_i \in A\}$ ).
- (10) (c) If  $A \in \mathbf{NP}$  and  $B \in \mathbf{NP}$ , then  $A \cup B \in \mathbf{NP}$ .
- (d) If  $A \in \mathbf{P}$ , then  $A^* \in \mathbf{P}$ .

**Exercise 2.** Prove the following statements.

- (10) (a) If  $A \in \mathbf{P}$  and  $B \subseteq \{0, 1\}^*$  with  $B \neq \emptyset$  and  $B \neq \{0, 1\}^*$ , then  $A \leq_P B$ .
- (10) (b) If  $A \in \mathbf{P}$  and  $A \in \mathbf{NPH}$ , then  $\mathbf{P} = \mathbf{NP}$ .

**Exercise 3.** Do the following hold? Give a proof or a counterexample.

- (a) If  $A \subseteq B \leq_P C$ , then  $A \leq_P C$ .
- (b) If  $A \leq_P B \subseteq C$ , then  $A \leq_P C$ .

**Exercise 4.** Put the following propositional formulas in CNF and determine if they are satisfiable.

- (5) (a)  $\varphi_1 := (\neg p \vee \neg(q \vee r)) \wedge (p \rightarrow q) \wedge (p \vee \neg q) \wedge (q \vee (r \wedge p))$ .
- (5) (b)  $\varphi_2 := ((p \rightarrow q) \rightarrow p) \rightarrow p$ .
- (5) (c)  $\varphi_3 := \neg(p \rightarrow q) \vee (q \wedge r)$ .

**Exercise 5.** We consider boolean logic with three connectives and one constant  $\vee, \wedge, \rightarrow, \perp$ . We know that **SAT** is in **NPH**. For each of the following problem, determine whether it is in **P** or in **NPH**. Prove your answer.

- (10) (a) The **SAT** problem for formulas restricted to  $\wedge, \rightarrow$
- (10) (b) The **SAT** problem for formulas restricted to  $\wedge, \neg$ .

(15) **Exercise 6.** Suppose we have a polynomial algorithm for **SAT**, that is: an  $f$  that computes for a boolean formula  $\varphi$  whether  $\varphi$  is satisfiable or not, in polynomial time in the size of  $\varphi$ .

Give an algorithm  $g$  that computes, in polynomial time, a satisfying assignment for a formula  $\varphi$  (in case  $\varphi$  is satisfiable, and return 0 otherwise).