Aalto University
Department of Computer Science
Pekka Orponen

CS-E4530 Computational Complexity Theory (5 cr)
Second Midterm Exam, Tue 7 Apr 2020, 1–5 p.m.

Instructions

1. This online examination is carried out using the Assignment feature of MyCourses. The submission window opens at 1:00 p.m. and closes at 5:00 p.m. Late submissions are not accepted, but you may update an incomplete submission up to the closing time.

2. The examination has four problems. Write your solutions in clearly legible handwriting, using a pen with good contrast from the paper, and **start the answer to each problem on a new sheet of paper**. You can write down your answers in either Finnish, Swedish, or English.

3. **Number each answer sheet** at the top right corner of the sheet with the problem number as 1, 2, 3, 4. If you use several sheets per problem, then number those as 1.1, 1.2, 2.1, 2.2, 2.3 etc. (This is important to collate the sheets in correct order from the online submission server.) **Sign each answer sheet at the top.**

4. Scan/photograph each answer sheet and upload it as a separate image file (.jpg, .tif, .pdf etc.) to the MyCourses Assignment submission server. **Indicate the number of the sheet in the name of the file** (e.g. 1-1.jpg, 1-2.jpg etc.) Make sure that you have enough time at the end of the exam to complete this task, even if the server is busy or some other technical adversities arise. Remember that you can update your submissions until the server closes, so it is better to start uploading early rather than late. If you leave uploading to the end, reserve at least 30 minutes for this task.

5. To make sure that any possible issues arising from the arrangements can be resolved, please be present throughout the exam in the Zoom meeting `https://aalto.zoom.us/j/110829034`. For any questions during the exam, first contact the meeting host privately via the Zoom chat.

Materials, code of conduct

1. The exam is "open (course)book", i.e. all course material available via the course's MyCourses page can be used for reference. This includes your personal solutions to the homework problems as uploaded on MyCourses, but no other materials, notes or discussions. If you have any questions about the materials, please contact the host as above.

Problems

1. (a) Prove (clearly but without going into too much detail) that the following two definitions of complexity class **coNP** are equivalent:
   (i) $\mathbf{coNP} = \{L \mid \overline{L} \in \mathbf{NP}\}$.
   (ii) **coNP** is the class of all languages L for which there exist a polynomial-time Turing machine M and a polynomial function $p\colon \mathbb{N} \to \mathbb{N}$ such that:
      1. M halts on all inputs,
      2. M always produces output 0 or 1, and
      3. for any string x, $x \in L$ if and only if for all u with $|u| \leq p(|x|)$ it holds that $M(x,u) = 1$. *3p.*

   (b) Recall that a Boolean expression ϕ is *valid* or a *tautology*, if it is satisfied by every possible truth assignment to its variables. Prove that the language
   $$\text{TAUT} = \{\phi \mid \phi \text{ is a tautology}\}$$
   is **coNP**-complete. *3p.*

2. Prove that the CIRCUIT SAT problem remains **NP**-complete even when restricted to monotone circuits, i.e. circuits containing only AND- and OR-gates. *8p.*

3. (a) Show that the following EXACT INDEPENDENT SET problem is in class Δ_2^p of the polynomial-time hierarchy: Given a graph $G = (V,E)$ and an integer $k \geq 1$, is the size of the largest independent set in G exactly k? *4p.*

   (b) Prove that if for some $i \geq 1$, $\Sigma_i^p = \Pi_i^p$, then for all $j > i$, $\Delta_j^p = \Sigma_j^p = \Pi_j^p = \Sigma_i^p$. (To prove the result, you may use either the oracle- or the quantifier-hierarchy characterisation of the polynomial-time hierarchy.) *4p.*

4. A *kernel* of a directed graph $G = (V,E)$ is a subset of vertices $K \subseteq V$ such that (i) no two vertices $u, v \in K$ are connected, i.e. neither $(u,v) \in E$ nor $(v,u) \in E$, and (ii) for any $w \notin K$ there is a directed edge $(u,w) \in E$ from some $u \in K$. Show that it is **NP**-complete to determine if a given directed graph G has a kernel. (*Hint:* A pair of vertices u, v that are interconnected to each other, but have no other incoming edges, can be used as a choice gadget. Think of **NP**-complete problems that you know where such a gadget would come in handy.) *8p.*

Total 30p.