Technische Universität München
Theoretische Informatik
Dr. J. Kreiker / Dr. M. Luttenberger, J. Kretinsky

Summer term 2010
August 2, 2010

Complexity Theory – Final Exam

***Please note*: If not stated otherwise, all answers have to be justified.**

Last name: ____________________________________

First name: ____________________________________

Student ID no.: ____________________________________

Signature: ____________________________________

Remarks

- If you feel ill, let us know immediately.
- Fill in all the required information and don't forget to sign.
- Write with a non-erasable pen, do not use red or green color.
- You have 10 minutes to read the questions and, after that, 120 minutes to write your solutions.
- You are not allowed to use auxiliary means other than your pen.
- You may write your answers in English or German.
- Check if you received *8 sheets of paper*.
- Write your solutions directly into the exam booklet. Please ask if you need additional (scrap) paper. Only solutions written on official paper will be corrected.
- You can obtain *40 points*. You need *17 points* to pass *including* potential bonuses awarded.
- Don't fill in the table below.
- Good luck!

Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Ex7	$\sum$

Exercise 1 **True/False** **each 1P=10P**

Points are awarded as follows:

	unsure ☐	unsure ☒
correct answer	1P	0.5P
wrong answer	-1P	-0.5P
no answer	0P	0P

Remarks

- All claims are definitely either true or false.
- Tick *unsure* ☐ in case of doubt to reduce a potential point deduction.
- A negative total implies zero points awarded for this exercise.

	true	false	unsure
PP has complete problems.	☐	☐	☐
$\bigcup_{k\in\mathbb{N}} \mathbf{IP}[n^k] = \bigcup_{k\in\mathbb{N}} \mathbf{AM}[n^k]$.	☐	☐	☐
If $\mathbf{L} \neq \mathbf{P}$, then $\mathbf{L}$ is closed under $\leq_p$.	☐	☐	☐
Every probabilistic TM running in expected polynomial time decides a language in $\mathbf{ZPP} := \mathbf{RP} \cap \mathbf{coRP}$.	☐	☐	☐
If graph isomorphism is **NP**-complete, then $\mathbf{\Sigma}_2^p = \mathbf{\Pi}_2^p$.	☐	☐	☐
3SAT has no polynomially-sized, interactive, zero-knowledge proofs.	☐	☐	☐
For every $L \in \mathbf{IP}$ there is an alternating TM deciding L in polynomial time.	☐	☐	☐
If every unary language $L \in \mathbf{NP}$ is already in $\mathbf{P}$, then $\mathbf{EXP} = \mathbf{NEXP}$.	☐	☐	☐
If $\mathbf{PH} = \mathbf{PSPACE}$, then $\mathbf{\Sigma}_k^p \subsetneq \mathbf{\Sigma}_{k+1}^p$ for all $k \in \mathbb{N}$.	☐	☐	☐
$\mathbf{AC} = \mathbf{NC}$.	☐	☐	☐

Exercise 2 "One-liners" **each 1P=8P**

Give a short answer. For **Claims** state clearly whether they hold or not.

Notation: Let **polyL** $:= \bigcup_{k\in\mathbb{N}} \mathbf{SPACE}((\log n)^k)$ be the class of languages decidable in poly-logarithmic space.

Claim: **polyL** $\subsetneq$ **EXP**.

Answer: ______

Question: State Ladner's theorem:

Answer: ______

Claim: Probably, $\mathbf{NP} \neq \mathbf{PCP}(1/n \cdot \log n, 1)$ as otherwise ...

Answer: ______

Question: State a **coNL**-complete problem.

Answer: ______

Question: By what kind of certificates is **NL** characterized?

Answer: ______

Question: We know that unless $\mathbf{P} = \mathbf{NP}$ there is some $\rho \in (0, 1]$ s.t. MAX3SAT cannot be ρ-approximated in polynomial time. State a problem for which we know that this holds for every $\rho \in (0, 1]$.

Answer: ______

Question: The reduction in the proof by Cook-Levin takes a Turing machine M and an input x and maps it on some CNF formula $\phi_{M;x}$. Let $N_{M;x}$ be the number of certificates u s.t. $M(x, u) = 1$. How is $N_{M;x}$ related to $\phi_{M;x}$?

Answer: ______

Question:: State a language in $\mathbf{NC}^2 \setminus \mathbf{AC}^0$ other than PARITY.

Answer: ______

Exercise 3 **2P+2P+2P=6P**

Consider the following set $\mathcal{C}$ of complexity classes.

$$\mathcal{C} = \{\mathbf{NC}^2, \mathbf{L}, \mathbf{IP}, \mathbf{ZPP}, \mathbf{PCP}(\log n, 1), \mathbf{\Sigma}_2^p \cap \mathbf{\Pi}_2^p, \mathbf{BPP}, \mathbf{PH}\}$$

(a) Draw a directed graph with nodes $\mathcal{C}$ such that there is a path from A to B if $A \subseteq B$.

(b) The "deterministic space hierarchy theorem" says that for any space-constructible functions f, g with $f(n) \in o(g(n))$ we have

$$\mathbf{SPACE}(f(n)) \subsetneq \mathbf{SPACE}(g(n)).$$

Show that $\mathbf{NL} \subsetneq \mathbf{IP}$.

(c) It is unknown whether $\mathbf{P} = \mathbf{L}$ and whether $\mathbf{P} = \mathbf{PSPACE}$. However, not both equalities can hold at the same time.

Which classes in $\mathcal{C}$ coincide

i) under the assumption that $\mathbf{P} = \mathbf{L}$?

ii) under the assumption that $\mathbf{P} = \mathbf{PSPACE}$?

Exercise 4 **2P+2P+2P+1P=7P**

Given an undirected graph $G = (V, E)$ we call $C \subseteq V$ a *clique* of G if for any two distinct nodes u, v of C there is an edge $(u, v) \in E$. A clique C is *maximal* if $C \cup \{u\}$ is not a clique for any $u \in V \setminus C$.

Consider the following decision and function problems related to cliques.

- $\textsc{Clique}_D := \{\langle G, k\rangle \mid G \text{ has a clique of size at least } k\}$.
- $\textsc{ExactClique}_D := \{\langle G, k\rangle \mid \text{Every maximal clique in } G \text{ has size exactly } k\}$.
- $\textsc{cliqueSize}(G) := \max\{\ |C| \mid C \text{ is a clique of } G\}$.

Remark: You may assume that $\textsc{Clique}_D$ is **NP**-complete.

(a) Show how to calculate $\textsc{cliqueSize}(G)$ in time $\mathcal{O}(T(n) \cdot \log n)$ under the assumption that $\textsc{Clique}_D$ can be decided in time $T(n)$ for any graph G of size n.

(b) Recall the definition of **DP**:

$$\mathbf{DP} := \{L_1 \cap L_2 \mid L_1 \in \mathbf{NP}, L_2 \in \mathbf{coNP}\}.$$

Show that $\textsc{ExactClique}_D$ is in **DP**.

(c) You may assume that the following language, $\textsc{ExactIndset}_D$, is **DP**-complete:

$$\textsc{ExactIndset}_D := \{\langle G, k\rangle \mid \text{Any independent set of maximal size in } G \text{ has size exactly } k\}.$$

Show that $\textsc{ExactClique}_D$ is **DP**-complete.

(d) Show that for no $\rho \in (0, 1]$ there exists a poly-time ρ-approximation of $\textsc{cliqueSize}$ unless $\mathbf{P} = \mathbf{NP}$.

Exercise 5 **4P**

A path $v_0 v_1 \dots v_k$ in a directed graph $G = (V, E)$ with nodes V and edges $E \subseteq V \times V$ is a *cycle* if $v_k = v_0$. Define

$$\textsc{Cycle} := \{\langle G, v \rangle \mid v \text{ lies on a cycle of } G\}.$$

Show that $\textsc{Cycle}$ is **NL**-complete.

Exercise 6 **2P**

Explain the error in the following proof of $\mathbf{P} \neq \mathbf{NP}$:

1) Suppose $\mathbf{P} = \mathbf{NP}$.
2) Then for some $k \in \mathbb{N}$, $\textsc{Sat} \in \mathbf{DTIME}(n^k)$.
3) As every language in $\mathbf{NP}$ is reducible to $\textsc{Sat}$, $\mathbf{NP} \subseteq \mathbf{DTIME}(n^k)$.
4) Due to the assumption $\mathbf{P} = \mathbf{NP}$, $\mathbf{P} \subseteq \mathbf{DTIME}(n^k)$.
5) Then $\mathbf{DTIME}(n^k) \subsetneq \mathbf{DTIME}(n^{k+1})$ is a contradiction with $\mathbf{P} \subseteq \mathbf{DTIME}(n^k)$.

Exercise 7 **1P+2P=3P**

Consider the following kind of computation:

Starting from a deterministic polynomial-time TM V, we extend V as follows:

- V has two special tapes:
  - a query-tape on which it can write Boolean formulae and
  - a response-tape.
- At any step of the computation, V can send the formula on the query tape to a SAT-prover that immediately tells V if the formula is satisfiable or not by writing either 1 or 0 to the response-tape.

Let $\mathbf{SAT}[k]$ be the class of all languages L which are decided by deterministic polynomial-time TMs asking at most k SAT-questions (where k may depend on the input length).

(a) The Boolean closure of $\mathbf{NP}$ is the class of all languages L of the form:

$$L := \bigcup_{i=1}^{m} \bigcap_{j=1}^{n} L_{i,j} \text{ where } L_{i,j} \in \mathbf{NP} \cup \mathbf{coNP}$$

Show that for any such language L there exists a constant k s.t. $L \in \mathbf{SAT}[k]$.

(b) Show that $\mathbf{SAT}[\text{poly}(n)] \subseteq \mathbf{\Sigma}_2^p \cap \mathbf{\Pi}_2^p$.

Remark: $\mathbf{SAT}[\text{poly}(n)]$ refers to the class of languages decided by deterministic polynomial-time TMs which can query the SAT-prover a number of times polynomial in the length of the input.