Second Exam — Algorithm Theory, LVA 703608 — Institute of Computer Science
September 27, 2007 — University of Innsbruck

Name: **Matr.-Nr.**: **Points**:

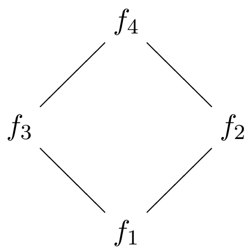
Good luck!

Small o notation

Which of the following functions are small o of which others? (Recall that we are only interested in functions $f\colon \mathbb{N} \to \mathbb{N}$. Hence when we write $f(n)$, we really mean $\max(\{\lceil f(n) \rceil, 0\})$.) [5]

(a)
- $n^{\log n}$
- $n^{\mathcal{O}(1)}$
- $\log n$
- $\log(\log n)$
- $\log^2 n$

Here the answer suffices in the form of a diagram:



(Meaning that $f_1(n) = o(f_2(n))$, $f_1(n) = o(f_3(n))$, $f_2(n) = o(f_4(n))$ and $f_3(n) = o(f_4(n))$ and neither $f_2(n) = o(f_3(n))$ nor $f_3(n) = o(f_2(n))$.)

(b) Explain your answers to (a) convincingly. [5]

Turing Machines

Consider the following Turing machine $\mathsf{M} = (K, \Sigma, \delta, q_0)$ with $K = \{q_0, q_1, q_2, q_3, yes, no\}$, $\Sigma = \{0, 1, \mathsf{X}, \mathsf{Y}, \triangleright, \sqcup\}$ and δ defined as follows. (We assume that if M cannot perform any step, then M halts in state *no*.)

$p \in K$	$\sigma \in \Sigma$	$\delta_M(p, \sigma)$
q_0	$\triangleright$	$(q_0, \triangleright, \rightarrow)$
q_0	0	$(q_1, \mathsf{X}, \rightarrow)$
q_0	Y	$(q_3, \mathsf{Y}, \rightarrow)$
q_1	0	$(q_1, 0, \rightarrow)$
q_1	1	$(q_2, \mathsf{Y}, \leftarrow)$
q_1	Y	$(q_1, \mathsf{Y}, \rightarrow)$

$p \in K$	$\sigma \in \Sigma$	$\delta_M(p, \sigma)$
q_2	0	$(q_2, 0, \leftarrow)$
q_2	X	$(q_0, \mathsf{X}, \rightarrow)$
q_2	Y	$(q_2, \mathsf{Y}, \leftarrow)$
q_3	Y	$(q_3, \mathsf{Y}, \rightarrow)$
q_3	$\sqcup$	$(yes, \sqcup, -)$

(a) Define formally, what is understood by (i) "a TM *decides* a language L" and (ii) "a TM *accepts* a language L". [4]

(b) Define the language $\mathrm{L} \subseteq \{0, 1\}^*$ decided by M. [5]

Complexity Classes

Consider the following tasks.

(a) Define the complexity class **NP**. [4]

(b) Define the problems SAT and 3SAT. [4]

(c) Define the function problem FSAT. [4]

(d) Consider the following assertion: "Given a polynomial algorithm for SAT, we can define a polynomial algorithm for FSAT." Is this assertion correct? Explain your answer briefly. [4]

Reductions and NP-completeness

(a) A *clique* in an *undirected* graph is a subgraph, where every two nodes are connected. A k-*clique* is a clique that contains k-nodes. Set

$$\text{CLIQUE} = \{(G, k) \mid G \text{ is an undirected graph with a } k\text{-clique}\}$$

Show that CLIQUE is in **NP**. [4]

(b) Assume the **NP**-completeness of SAT and show the **NP**-completeness of 3SAT. (Prepare a formal solution.) [5]

(c) Given a graph G and nodes $1, n \in V$, is there a path between 1 and n? This problem is called REACHABILITY. CIRCUIT VALUE is the problem, where given a variable-free circuit C, we ask whether there exists an assignment T, such that $T(C) = \textbf{true}$?

Give the central idea of a reduction R from REACHABILITYto CIRCUIT VALUE. (An informal argument suffices.) [6]