3. Assignment on

Computational Complexity Theory

SoSe 2016

Wolfgang Mulzer, Claudia Dieckmann

Due on 17. May 2016 in the tutorial session

Problem 1 Relationships Between Complexity Classes *10 points*

(a) Show that NP $\neq$ DSPACE(n).

Hint: We know that NP is closed under polynomial-time reductions. What about DSPACE(n)?

(b) Show that EXP $\neq$ NEXP $\Rightarrow$ P $\neq$ NP.

Hint: Use padding to prove the contrapositive P = NP $\Rightarrow$ EXP = NEXP.

Problem 2 Single-tape Turing Machines I *10 points*

Let $T : \mathbb{N} \to \mathbb{N}$ be time-constructible, and let $\mathrm{NTIME}_1(T)$ be the set of languages that can be decided in $O(T(n))$ time by a non-deterministic Turing machine *with only one (read-write) tape.* $\mathrm{DTIME}_1(T)$ is defined similarly.

Let $L := \{x0^n y \mid n \in \mathbb{N}, \quad x, y \in \{0,1\}^n, \quad x \neq y\}$. Show that L is contained in $\mathrm{NTIME}_1(n \log n)$.

Hint: First, show that L can be decided in $O(n)$ time by a non-deterministic Turing machine with two tapes, where the second tape contains only $O(\log n)$ cells. Then show how to simulate this machine by a single-tape non-deterministic Turing machine in $O(n \log n)$ time.

Problem 3 Single-tape Turing Machines II *10 points*

Let L be defined as in the second problem. We will now show that any deterministic Turing machine with a single work tape requires $\Omega(n^2)$ time to decide L.

(a) Let $L' := \{x0^n x \mid n \in \mathbb{N}, \quad x \in \{0,1\}^n\}$. Show that if there exists a deterministic single-tape Turing machine that decides L in time $T(n)$, then there is a deterministic single-tape Turing machine that decides L' in time $T(n) + O(n \log n)$.

(b) Let M' be a Turing machine that decides L' in time $T'(n)$. Let $i \in \mathbb{N}$ and $x \in \{0,1\}^n$. We define the i^{th} *crossing sequence* for M' on x: let t_1 be the first time that M' on input $x0^n x$ goes from cell i to cell $i+1$; let t_2 be the first time that M' goes from cell $i+1$ to i; let t_3 be the second time that M' goes from i to $i+1$; let t_4 be the second time that M' goes from $i+1$ to i; etc. Let t_m be the last time that M' crosses from cell i to $i+1$ or back. Clearly, we have $t_1 < t_2 < t_3 < \cdots < t_m$.

The i^{th} crossing sequence is the sequence $(q_1, q_2, \ldots, q_m)$, where q_j is the state of M' at time t_j. We call m the *length* of the crossing sequence.

Argue that for any input $x0^n x$, there has to be an $i \in \{n, \ldots, 2n-1\}$ such that the i^{th} crossing sequence on x has length at most $T'(n)/n$.

(c) Let $x \neq y \in \{0,1\}^n$. Show that for every $i \in \{n, \ldots, 2n-1\}$ the i^{th} crossing sequence for M' on x must be different from the i^{th} crossing sequence for M' on y. Conclude that for every $x \in \{0,1\}^n$ and for every $i \in \{n, \ldots, 2n-1\}$, the length of the i^{th} crossing sequence on x must be $\Omega(n)$.

(d) Conclude that $T'(n) = \Omega(n^2)$, so $\text{DTIME}_1(n \log n) \subsetneq \text{NTIME}_1(n \log n)$.

Remark: For multi-tape Turing machines, Paul, Pippenger, Szemeredi, and Trotter showed that $\text{DTIME}(n) \subsetneqq \text{NTIME}(n)$, but the proof is much harder, and the gap is minuscule (about $\log^* n$).