TU Dresden, Fakultät Informatik
Stephan Mennicke, Markus Krötzsch

Winter Term 2021/2022

Complexity Theory
Exercise 6: Diagonalization
8th December 2021

Exercise 6.1. Find the fault in the following proof of $\mathsf{P} \neq \mathsf{NP}$.

1. Assume that $\mathsf{P} = \mathsf{NP}$.
2. Then $\mathsf{SAT} \in \mathsf{P}$ and thus there exists a $k \in \mathbb{N}$ such that $\mathsf{SAT} \in \mathsf{DTime}(n^k)$.
3. Because every language in NP is poly-time reducible to SAT, we have $\mathsf{NP} \subseteq \mathsf{DTime}(n^k)$.
4. It follows that $\mathsf{P} \subseteq \mathsf{DTime}(n^k)$.
5. By the Time Hierarchy Theorem there exist languages in $\mathsf{DTime}(n^{k+1})$ that are not in $\mathsf{DTime}(n^k)$, contradicting $\mathsf{P} \subseteq \mathsf{DTime}(n^k)$.
6. Therefore, $\mathsf{P} \neq \mathsf{NP}$.

Exercise 6.2. Show the following.

1. $\textsc{Time}(2^n) = \textsc{Time}(2^{n+1})$
2. $\textsc{Time}_*(2^n) \subset \textsc{Time}_*(2^{2n})$
3. $\textsc{NTime}(n) \subset \textsc{PSpace}$

Exercise 6.3. Define a function that is computable but not time-constructible.

Exercise 6.4. Consider the function $\mathsf{pad}\colon \Sigma^* \times \mathbb{N} \to \Sigma^*\#^*$ defined as $\mathsf{pad}(s, \ell) = s\#^j$, where $j = \max(0, \ell - |s|)$. For some language $\mathbf{A} \subseteq \Sigma^*$ and $f\colon \mathbb{N} \to \mathbb{N}$ define $\mathsf{pad}(\mathbf{A}, f) = \{\, \mathsf{pad}(s, f(|s|)) \mid s \in \mathbf{A} \,\}$.

Show all of the following satements.

1. Show that, if $\mathbf{A} \in \textsc{DTime}(n^6)$, then $\mathsf{pad}(\mathbf{A}, n^2) \in \textsc{DTime}(n^3)$.
2. Show that, if $\textsc{NExpTime} \neq \textsc{ExpTime}$, then $\mathrm{P} \neq \mathrm{NP}$.
3. Show for every $\mathbf{A} \subseteq \Sigma^*$ and $k \in \mathbb{N}$ that $\mathbf{A} \in \mathrm{P}$ if and only if $\mathsf{pad}(\mathbf{A}, n^k) \in \mathrm{P}$.
4. Show $\mathrm{P} \neq \textsc{DSpace}(n)$.
5. Show $\mathrm{NP} \neq \textsc{DSpace}(n)$.

Exercise 6.5. You are given two oracles and one of them is the set **True QBF**, but you do not know which one. Design a polynomial algorithm that decides **True QBF** using these oracles.