Computational Complexity

Homework Sheet 4

Hand in via Canvas before May 17, 2023, at 23:59

`https://canvas.uva.nl/courses/36220/assignments/409543`

Exercise 1 (3pt). Define:

$$\mathsf{P/log} = \bigcup_{c,d \in \mathbb{N}} \mathrm{DTIME}(n^c)/(d \log n).$$

That is, $\mathsf{P/log}$ is the class of all languages that can be decided in polynomial time with $O(\log n)$ bits of advice. Prove that $\mathsf{SAT} \notin \mathsf{P/log}$, unless $\mathsf{P} = \mathsf{NP}$.

- *Hint:* iterate over all possible advice strings of length $O(\log n)$.
- *Hint:* you may assume that for any string x that represents a propositional formula φ and any truth assignment α to (some of) the variables of φ, one can in polynomial-time encode the formula $\varphi[\alpha]$ as a string x' that is of the same length as x—where $\varphi[\alpha]$ is obtained from φ by instantiating each variable z in the domain of α by $\alpha(z)$.

Exercise 2 (2pt). Let $\mathsf{P/1}$ be the class of languages that can be decided in polynomial time with a single bit of advice (for each input size). That is $\mathsf{P/1} = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}(n^c)/1$. Prove that $\mathsf{P} \subsetneq \mathsf{P/1}$.

- *Hint:* use the following undecidable language UHALT:

$$\begin{aligned}\mathsf{UHALT} = \{\ 1^n \mid\ & n\text{'s binary expansion encodes a pair } (\mathbb{M}, x) \\ & \text{such that } \mathbb{M} \text{ is a Turing machine that halts on input } x\ \}.\end{aligned}$$

Exercise 3 (2pt). Let P_{nu} be the class of all decision problems $L \subseteq \{0,1\}^*$ such that for each $n \in \mathbb{N}$, the problem $L \cap \{0,1\}^n$ is polynomial-time solvable. That is, for each $n \in \mathbb{N}$, there exists a deterministic TM $\mathbb{M}_n$ and a constant c_n such that $\mathbb{M}_n$ decides $L \cap \{0,1\}^n$ and runs in time $O(n^{c_n})$.

In this exercise, you will show that P_{nu} is different from $\mathsf{P/poly}$.[1] You may opt for an easy variant (A) of this exercise, for a total maximum of 1½ points, or a harder variant (B) of this exercise, for a total maximum of 2 points.

(A) Prove that if $\mathsf{P}_{\mathrm{nu}} = \mathsf{P/poly}$, then the Polynomial Hierarchy collapses.

(B) Prove that $\mathsf{P}_{\mathrm{nu}} \neq \mathsf{P/poly}$.

  – *Hint:* for (B), have a look at Sections 6.5 and 6.6 from the book [1].

Exercise 4 (3pt). Consider the following two complexity classes $\mathsf{P}^{\mathsf{NP}[\log]}$ and $\mathsf{P}^{\mathsf{NP}}_{||}$:

- $\mathsf{P}^{\mathsf{NP}[\log]}$ is the class of all decision problems $L \subseteq \{0,1\}^*$ for which there exists a polynomial-time deterministic oracle TM $\mathbb{M}$ and an oracle language $O \in \mathsf{NP}$ such that $\mathbb{M}^O$ decides L, and a function $f(n) : \mathbb{N} \to \mathbb{N}$ that is $O(\log n)$ such that for each input $x \in \{0,1\}^*$, $\mathbb{M}^O(x)$ makes at most $f(|x|)$ queries to the oracle O.
- $\mathsf{P}^{\mathsf{NP}}_{||}$ is the class of all decision problems $L \subseteq \{0,1\}^*$ for which there exists a polynomial-time deterministic oracle TM $\mathbb{M}$ and an oracle language $O \in \mathsf{NP}$ such that $\mathbb{M}^O$ decides L, and for each input, $\mathbb{M}^O$ makes their queries to O *in parallel.*

[1] In fact, the class P_{nu} turns out to be not very interesting. You don't have to prove this, but understanding why will help you solve this exercise.

Making parallel oracle queries works as follows. The machine $\mathbb{M}$ may write an arbitrary number of oracle queries $q_1, \ldots, q_m \in \{0,1\}^*$ on the oracle tape, separated by a designated symbol `#` (i.e., $q_1\texttt{\#}q_2\texttt{\#}\cdots\texttt{\#}q_m$). Then, when the machine $\mathbb{M}$ enters the designated query state $q_{\text{query}} \in Q$, instead of transitioning into q_{yes} or q_{no} depending on the answer of the oracle query, the machine transitions into a designated state q_{done}, and the contents of the oracle tape are replaced by the string $b_1\texttt{\#}b_2\texttt{\#}\cdots\texttt{\#}b_m$ that represents the answers $b_1, \ldots, b_m \in \{0,1\}$ to the oracle queries (e.g., `0#1#1#0`), where $b_i = 1$ if and only if q_i is in the oracle language O. The machine $\mathbb{M}$ is only allowed to make a single such combined query for each input $x \in \{0,1\}^*$. In other words, it must compute (and write down) all the oracle queries that it wants to make before getting an answer for any of them.

Prove that $\mathsf{P}^{\mathsf{NP}[\log]} = \mathsf{P}^{\mathsf{NP}}_{||}$.

- *Hint:* for showing that $\mathsf{P}^{\mathsf{NP}}_{||} \subseteq \mathsf{P}^{\mathsf{NP}[\log]}$, choose an NP-complete oracle O, allowing you to solve different NP problems by querying this single oracle—using the fact that for each $L \in \mathsf{NP}$, there exists a polynomial-time reduction from L to O.

Remark 1. Answers will be graded on two criteria: they should (1) be correct and intelligent, and also (2) concise and to the point.

Remark 2. If you find a solution to one of the exercises in a paper or book, you can use this to inform your solution. Make sure that you write down the solution in your own words, conveying that you understand what is going on.

References

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach.* Cambridge University Press, 2009.