Computational Complexity

Exercise Session 5

Note: these solutions are (often) merely pointers to the right idea that is needed to solve the problems. These are not fully worked-out solutions. So please do not take these solutions as an example for how to write up your solutions for, e.g., the homework assignments. :-)

Note: These exercises are (likely) too much work to solve all during the exercise session.

Exercise 1. A decision problem $L \subseteq \{0,1\}^*$ is *sparse* if there exists a polynomial p such that for every $n \in \mathbb{N}$ it holds that $|L \cap \{0,1\}^n| \leq p(n)$. Show that every sparse decision problem is in $\mathsf{P/poly}$.

Solutions:

- For each n, one can let the advice α_n be a description of all the strings in $L \cap \{0,1\}^n$. Since L is sparse, there exists a polynomial p such that $|L \cap \{0,1\}^n| \leq p(n)$ for each n, and thus we can bound the size of the α_n's by a polynomial (roughly $n \cdot p(n)$). Using this advice α_n, we can decide $L \cap \{0,1\}^n$ in polynomial time, because for each $x \in \{0,1\}^n$, we only have to check if x appears in the list in α_n.

Definition 1. $\mathsf{P}^{\mathsf{NP}[\log]}$ 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.

Exercise 2. Show that the following problem is in $\mathsf{P}^{\mathsf{NP}[\log]}$:

$$\{ \varphi \mid \varphi \text{ is a propositional logic formula, and the maximum number } m \text{ of variables among } \mathrm{var}(\varphi) \\ \text{that are set to true in any satisfying truth assignment of } \varphi \text{ is odd.} \}$$

Solutions:

- Use the oracle to decide the following, for different values k: "there exists a truth assignment α that satisfies φ and that sets at least k variables in φ to true." The inputs are pairs (φ, k).
- This is an NP problem, so we can use an NP oracle to solve this.
- By using binary search, we can find the maximum number m using only a logarithmic number of queries.
- Once we have m, deciding if it is odd or even is easy.

Exercise 3. Prove that $\mathsf{RP} \subseteq \mathsf{BPP}$ and that $\mathsf{coRP} \subseteq \mathsf{BPP}$.

Solutions:

Any RP algorithm is also a BPP algorithm, because it runs in polynomial time and gives the correct answer with probability at least $2/3$.

Exercise 4. Prove that $\mathsf{BPP} \subseteq \mathsf{PSPACE}$.

Solutions:

By using polynomial space, you can iterate over all possible random choices, and give the answer that is given most often. This will then be the correct answer.

Exercise 5. CLIQUE is the problem of deciding, given a graph $G = (V, E)$ and a natural number $k \in \mathbb{N}$, whether there exists a set $C \subseteq V$ such that $|C| = k$ and for all $c_1, c_2 \in C$ with $c_1 \neq c_2$ it holds that $\{c_1, c_2\} \in E$.

For every $\rho < 1$, an algorithm A is called a ρ-approximation algorithm for MAX-CLIQUE if for every graph $G = (V, E)$, the algorithms outputs a clique $C \subseteq V$ of G of size at least $\rho \cdot \mu_G$, where μ_G is the maximum size of any clique of G.

Show that for each $\rho < 1$, if there exists a polynomial-time ρ-approximation algorithm for MAX-CLIQUE, then $\mathsf{P} = \mathsf{NP}$.

Solutions:

- Take the following polynomial-time reduction from IS to CLIQUE:
  - For any instance (G, k) with $G = (V, E)$, produce $(\overline{G}, k)$, where $\overline{G} = (V, \overline{E})$, and $\overline{E} = \binom{V}{2} \setminus E$.
- By using this reduction, any ρ-approximation for MAX-CLIQUE can be used to produce a ρ-approximation for MAX-IS, since the solutions are in one-to-one correspondence (and this correspondence maps solutions of size m to solutions of size m).
- Then, by Theorem 11.15 in the book, the result follows.