Complexity Theory

Supervision 3

6. NP, co-NP, and UP

1. It is often claimed that a proof of the proposition $\mathrm{P} = \mathrm{NP}$ would have drastic consequences: it would let us solve difficult optimisation problems efficiently, but would also break security and e-commerce by making public-key cryptography impossible. What objections could be made against such a claim?

2. The complexity class NP is closed under which of the following set-theoretic operations: intersection, union, complement? Briefly justify your answers.

3. Prove or disprove the following claims, or show that it is an open problem:

   a) If $L, K \in \text{co-NP}$ then $L \cup K \in \text{co-NP}$.

   b) If $L \in \mathrm{NP}$, $K \subset L$ and $K \in \text{co-NP}$ then $L \setminus K \in \mathrm{NP}$.

   c) If L is NP-complete, then $D = \{ xx \mid x \in L \}$ is NP-complete.

4. Show that a language L is in co-NP if, and only if, there is a nondeterministic Turing machine M and a polynomial p such that M halts in time $p(n)$ for all inputs x of length n, and L is exactly the set of strings x such that *all* computations of M on input x end in an accepting state.

5. Define a *strong* nondeterministic Turing machine as one where each computation has three possible outcomes: accept, reject or maybe. If M is such a machine, we say that it accepts L, if for every $x \in L$, every computation path of M on x ends in either accept or maybe, with at least one accept, *and* for $x \notin L$, every computation path of M on x ends in reject or maybe, with at least one reject.

   Show that if L is decided by a strong nondeterministic Turing machine running in polynomial time, then $L \in \mathrm{NP} \cap \text{co-NP}$.

6. We saw in the lectures that if there is a one-way function, then there is a language L in UP that is not in P. Suppose that the RSA function described in the lecture notes (page 38) is a one-way function. What is the language L that can then be proved to be in $\mathrm{UP} \setminus \mathrm{P}$?

7. Space complexity

1. Show that, for every nondeterministic machine M which uses $O(\log n)$ work space, there is a machine R with three tapes (`input`, `work` and `output`) which works as follows: on input x, R produces on its output tape a description of the configuration graph for M, x, and R uses $O(\log |x|)$ space on its work tape.

   Explain why this means that if Reachability is in L, then $\mathrm{L} = \mathrm{NL}$.

2. Consider the language L in the alphabet $\{ a, b \}$ given by $L = \{ a^n b^n \mid n \in \mathbb{N} \}$. The language L is *not* in $\mathrm{SPACE}(c)$ for any constant c. Why?

3. Consider the algorithm presented in the lecture which establishes that Reachability is in $\mathrm{SPACE}((\log n)^2)$. What is the time complexity of this algorithm?

   Can you generalise the time bound to the entire complexity class? That is, give a class of functions F such that

$$\mathrm{SPACE}((\log n)^2) \subseteq \bigcup_{f \in F} \mathrm{TIME}(f)$$

8. Hierarchy

1. On page 42 of the notes, a number of functions are listed as being constructible. Show that this is the case by giving, for each one, a description of an appropriate Turing machine. Instead of $\lceil \log n \rceil$, you may find it easier to try $n \cdot \lceil \log n \rceil$.

   Prove that if f and g are constructible functions and $f(n) \geq n$, then so are $f \circ g$, $f + g$, $f \times g$ and 2^f.

2. For any constructible function f, and any language $L \in \mathrm{NTIME}(f(n))$, there is a nondeterministic machine M that accepts L and *all* of whose computations terminate in time $O(f(n))$ for all inputs of length n. Give a detailed argument for this statement, describing how M might be obtained from a machine accepting L in time $f(n)$.

Optional exercises

1. POLYLOGSPACE is the complexity class

$$\bigcup_k \mathrm{SPACE}((\log n)^k).$$

   a) Show that, for any k, if $A \in \mathrm{SPACE}((\log \mathrm{n})^{\mathrm{k}})$ and $B \leq_L A$, then $B \in \mathrm{SPACE}((\log n)^k)$.

   b) Show that there are no POLYLOGSPACE-complete problems with respect to $\leq_L$. (/Hint/: use a) and the Space Hierarchy Theorem).

   c) Which of the following might be true: $\mathrm{P} \subseteq \mathrm{POLYLOGSPACE}$, $\mathrm{P} \supseteq \mathrm{POLYLOGSPACE}$, $\mathrm{P} = \mathrm{POLYLOGSPACE}$?

   d) What is the relationship between the class POLYLOGSPACE and the class Quasi-P (see Exercise Sheet 1, Question 3.1)?

2. In the lecture, a proof of the Time Hierarchy Theorem was sketched. Give a similar argument for the following Space Hierarchy Theorem:

   For every constructible function f, there is a language in $\mathrm{SPACE}(f(n) \times \log(f(n)))$ that is not in $\mathrm{SPACE}(f(n))$.

   Use this to show that if $\mathrm{SPACE}((\log n)^2) \subseteq P$, then $\mathrm{L} \neq \mathrm{P}$.