Complexity Theory

Homework Sheet 2

Turn in before the lecture of Tuesday 15 Sep.

8 September 2015

Exercise 1. A colouring of a graph with c colours is an assignment of a number from $1, 2, \dots, c$ to each vertex such that no adjacent vertices get the same number.

(a) Show that the following problem is in P.

Two-colouring: $2\text{COL} = \{G : \text{graph } G \text{ has a colouring with two colours}\}$

(b) Contrary to the two-colour version, the three-colouring problem is known to be NP-complete. (You don't have to prove this.)

Three-colouring: $3\text{COL} = \{G : \text{graph } G \text{ has a colouring with three colours}\}$

Consider the following scenario: A mysterious (but trustworthy) wizard gives you a graph, and promises you that it is colourable with three colours.

Assuming $\text{P} \neq \text{NP}$, is it possible to come up with an efficient algorithm that finds a three-colouring of a graph, if you already know that such a colouring exists?

Exercise 2. Let A be an NP-complete language. Let p be a polynomial and M_A a polytime TM such that

$$x \in A \iff \exists\, u \in \{0,1\}^{p(|x|)} : M_A(x,u) = 1.$$

(a) Define the set $B = \{\langle x, z\rangle \mid \exists\, z' \text{ such that } |zz'| \leq p(|x|) \text{ and } M_A(x, zz') = 1\}$. Show that B is in NP.

(b) Assume that we have access to A as an oracle. Basically this means that we have a subroutine which, given a string y, tells in a single step whether $y \in A$. (See Def. 3.4 in the book.) Construct a polytime TM M_{search} that, given $x \in \{0,1\}^*$, if $x \in A$ outputs a string u such that $M_A(x,u) = 1$ and if $x \notin A$ outputs 0. Use **(a)**.

Exercise 3. Let A be a language. When a TM M has access to the oracle A, we write M^A. We say A is *auto-reducible* if there is a polytime TM M such that

$$x \in A \iff M^A(x) = 1$$

with the special requirement that on input x the TM M is not allowed to query the oracle A for x.

Suppose A is NP-complete. Show that A is auto-reducible. Use **Exercise 2**.

Exercise 4. In this exercise we will construct a language that is not auto-reducible, using diagonalization.

(a) Define $b : \mathbf{N} \to \mathbf{N}$ by $b(1) = 2$ and $b(n) = 2^{b(n-1)}$. Show that $b(i) > b(i-1)^{i-1}$ for $i > i_0$ for some i_0.

(b) Let M be a polytime oracle TM that does not query its input to the oracle. Show that there exists an i such that $M = M_i$ and M_i^O runs in time $\leq n^i$ for all oracles O, with n the size of the input. Remember that we can make our representation scheme $i \mapsto M_i$ in such a way that every TM has infinitely many representations.

(c) Suppose the M_i^O of **(b)** is given $0^{b(i)}$ as an input (a string of zeroes of length $b(i)$). What can you say about the size of the queries that M_i^O makes to O?

(d) Construct a set $A \subseteq \{0\}^*$ that is not auto-reducible. You should construct A in stages A_i such that $A = \cup_i A_i$. Recursively define $A_i \subseteq \{0\}^{b(i)}$, such that A is not auto-reducible by construction. Don't forget to prove that the set is not auto-reducible.

Hint: suppose you have constructed $A_1, \ldots, A_{i-1}$. Let $A_{\leq i-1} = \cup_{j \leq i-1} A_j$. Consider the machine $M_i^{A_{\leq i-1}}$ with input $0^{b(i)}$ that does not query $0^{b(i)}$. Based on the output of this machine decide whether $0^{b(i)}$ is in A_i or not.

(e) Show that A is in EXP. **(Bonus)**