KTH Computer Science
and Communication

Homework II, Complexity Theory 2008

Due on May 6 at 8.15. The general rules on homework solutions available at the course home-page apply. In particular, discussions of ideas in groups of up to at most three people are allowed but solutions should be written down individually. There is a bonus problem outside the standard total of 100 points. Points on this problem are counted as normal points. Try to be mathematically correct in your arguments.

1 (15p) Nondeterminism can in many circumstances be thought of as an existential quantifier. In particular a language A belongs to NP iff there is a language of pairs $B \in P$ and a constant k such that

$$x \in A \Leftrightarrow \exists y(|y| \leq |x|^k) \wedge (x, y) \in B. \qquad (1)$$

Let us consider the class of languages, A, that can we written on the form (1) but with the requirement that $B \in$ L. Do we in such a case get NL or do we get some other class?

2 (20p) Show by an explicit program that multiplication belongs to L. You need not design a Turing machine but can describe a high level algorithm but you should keep careful count of the space it uses.

2a (10p) Show that this is true when bits are output in the natural order with least significant bit first.

2b (5p) Show that this is true when bits are output in the natural order with most significant bit first.

2c (1p) Is the above fact true for any function in L? In other words is it always true that if outputting the answer in one order is in L so is outputting the answer in the reverse order?

2d (4p) To what extent can you output the bits in any order? Can you first output the odd numbered bits in order and then the even numbered bits in order? What conditions do you need? You need not prove that your conditions are necessary only that they are sufficient.

3 (17p) In class we briefly discussed a modified type of Vertex Cover where each edge as to have exactly one end point in the chosen set S.

3a (5p) Show that not all graphs have a vertex cover in this sense.

3b (10p) How hard is it to decide if a graph has such a cover? Polynomial time or NP-complete?

3c (2p) Do you think it is an interesting computational problem to optimize the size of the set S for this variant of Vertex Cover? Motivate!

4 (20p) Consider the problem of Acyclic Geography (AC) which is just like Generalized Geography, but the graph is required to be acyclic, i.e. it has no directed cycle. What is the complexity of AC? It is complete for one of the classes L, NL, P, NP or PSPACE. For which class is it complete? Prove your claim!

5 (20p) Consider the problem of solving systems of linear equations over the field of three elements. Put more concretely "+" is addition modulo 3 and variables take values in $\{0, 1, 2\}$. En example would be to

$$\begin{aligned} x_1 + x_2 &= 0 \\ x_1 &= 1 \\ x_2 &= 1 \end{aligned}$$

Suppose you are given a system in n variables with m equations and an integer k and you are asked whether there is an assignment that satisfies at least k equations.

5a (10p) Show that the sub-case of $k = m$ is in P.

5b (10p) Show that the general problem is NP-complete.

In the second problem you may only assume NP-completeness of a problem showed NP-complete in class, i.e. Clique, Independent Set, Vertex Cover, Hamiltonian Circuit or 3Sat.

6 (15p) Reachability in directed graphs was showed in class to be NL-hard. The problem is given a directed graph G and vertices s and t to determine whether there is a directed path from s to t. Let us for simplicity assume that the out-degree of each vertex in G is two (otherwise we can add some extra vertices to make it two) and that we have n vertices. Let us look at a randomized algorithm that behaves similarly to non-deterministic algorithm that moves along edges and at each point randomly chooses which of the two edges to use. We want to reach t with probability at least one half if it is reachable at all.

6a (5p) Prove that starting a walk at s and moving as described above does not always achieve this goal even if you continue moving for ever.

6b (10p) Suppose that you keep a counter and every T steps you restart the walk from s. Prove that for a suitable value of T (which?) you reach t with probability 1/2 if you run for long enough. Make an upper and lower bound for how long you may need to run this process to reach t with this probability.

7 (**Bonus, 25p**) The free group generated by $\{a, b\}$ is the set of formal products (strings) of symbols a, b, a^{-1}, b^{-1}.

A sequence can be simplified by the rule that if a symbol is next to its inverse then this pair of symbols can be removed (a^{-1} is the inverse of a and b^{-1} is the inverse of b). For instance $aba^{-1}ab^{-1}a^{-1}ab$ can be simplified to $abb^{-1}b$ which can be simplified to ab. If a product can be simplified to the empty string then it represents the identity in the free group generated by $\{a, b\}$. Your task is to prove that given such an expression the computational problem of deciding whether it represents the identity is in L.

Again you can argue at a fairly high level and need not describe a Turing machine.

Hint: The only proof I know uses the following useful fact. The matrices

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

and and their inverses generate exactly the free group with two generators (prove this (10p)). Consider a product of n matrices (where each matrix is one of A, B, A^{-1} and B^{-1}) and consider the problem of deciding whether such a product is the identity.