Solution notes

Complexity Theory 2005 – Paper 6 Question 12 (ACN)

(*a*) When defining space-complexity we use two-tape Turing Machines, with a read-only tape for input-data and a read-write working tape. We only count the space used on the work tape. What difference would be made if we worked with standard one-tape Turing machines, loaded the input data onto the tape to start with and measured space in terms of total tape cells touched by the end of the computation?

About the only issue would be that if you only had one tape then you could not talk about complexities less than $O(n)$ since the size of the input data is that.

(*b*) Comment on the following: "The problem of finding a factor of a number N is NP, because if we have a factor P or N we can do a simple trial division and check it in time related to $\log(N)$. Thus finding factors of numbers of the form $2^p - 1$ (these are known as Mersenne Numbers) is a problem in the class NP."

If you specify $2^n - 1$ by giving n then it is characterised in $\log(n)$ bits but the division witness is checked in time related to $log(2^n - 1)$ which is not polynomial in that, so we have a problem!

(*c*) Define the class co-NP. State an example of a problem that lies in in.

Languages whose complements are in NP.

(*d*) What is a witness-function for an NP problem? Why might some problem such as 3-SAT have many different witness functions associated with it?

A function that, given a language defined in NP in the form

$$L = \{x | \exists y R(x, y)\}$$

then if $x \in L, f(x) = y$ for some y, otherwise $f(x)$ must report that no solution exists.

(*e*) Give and justify a relation between $NTIME(f(n))$ and $SPACE(f(n))$.

The time-bounded one is within the space-bounded one. That is because any one path through the computation of the time-bounded computation clearly uses no more space than time, and one can then just implement a simple back-tracking search that simulates the non-determinism - doing that does not need any extra space.

(*f*) Matchings on bi-partite graphs can be found in polynomial time. The matching problem on tri-partite graphs is known to be NP-complete. Does this suggest that the corresponding problem with a graph whose nodes are partitioned into four sets ("quad-partite" matching) is liable to be exponential in complexity?

Silly! For 4-partite you could still obviously use nondeterminism to "guess' a matching, and then checking it is P. So we are in NP. Also we are at least as bad as 2-partite. So we are NP-complete again.

(g) Comment on the following proposition: "Determining which player can force a win from a given starting position in the game of Chess is an NP problem because given any sequence of moves it will be easy to verify that they are all legal moves and easy to see who wins at the end of them."

This too is garbage, in at least 2 ways! There are only a finite number of chessboard positions so we can not usefully speak in terms of asymptotic complexity, and also seeing who wins after a sequence of moves does not tell us who can force a win.