Solution notes

Complexity Theory 2005 – Paper 5 Question 10 (ACN)

(*a*) A (decision) problem (class) is NP if it is possible to solve every instance of it in polynomial time using an non-deterministic (Turing) machine. Polynomial time means that there is some polynomial function $p(n)$ such that the time to solve *every* instance of the problem of size n in time bounded by $p(n)$.

To say that a non-deterministic Turing machine solves a problem (in polynomial time) means that some one of the computations it can perform will terminate yielding "true".

An equivalent explanation is that an ordinary deterministic Turing machine can verify a certificate for the problem in the given amount of time.

(*b*) A problem is NP-complete if it is both NP and all other NP problems can be reduced to it in polynomial time.

(*c*) An alphabet, a set of states, an initial state, a subset of states that "accept" and a transfer relation in (states*symbols*states).

In N steps the machine can use at most N tape cells. So you need to know, for each time-step k, the state, the index within the N tape cells potentially used that the read-write head is at and the contents of each tape cell.

This related to solving *arbitrary* NP problems in that an NP problem can (by definition!) be solved in polynomial time on an NDTM, so we can view the behaviour of such a machine as an explanation of how this solution emerges.

(*d*) Consider some computation by an NDTM that takes N steps. I will show how to create an instance of 3-SAT that has size polynomial in N such that if can be solved only if the NDTM terminates properly.

Create (boolean) variables $S_{i,j}$ for i from 0 to N and j ranging over the number of states the TM has. Interpret one of these being true as indicating that the TM is in a given state at a given time. Start by asserting $S_{0,0}$ to force the TM into its starting state (0) at time 0, ans similarly force it to be in an accepting state at time N.

Create more variables P to mark the tape position, and T to say that at time i tape position j holds symbol k.

Observe that *all* behaviour of the NDTM can be expressed in deductions of the form

$$v \wedge w \wedge x \Rightarrow y$$

where the little letters stand for some of the variables just introduced. Eg to specify that between time 0 and 1 tape cell 3 remains holding the same content I can do two things:

$$T_{0,3,k} \Rightarrow T1, 3, k$$

for all relevant values of k, together with a set of rules that prevent the boolean model from suggesting that two distinct symbols lie in the same location on the tape

$$T_{1,3,k} \Rightarrow \neg T1, 3, l$$

listed for all distinct k and l.

What we will then do is "and" together all the implications to get a formula this is satisfiable if and only if there exists a terminating successful computation of the NDTM. Note that $a \wedge b \Rightarrow c$ can be rewritten as $\neg a \vee \neg b \vee c$ so what we have here is in CNF. It can be reduced to 3-CNF by padding out short terms and splitting long ones (easily, perhaps introducing new variable names bit not enlarging the formula much).

For the example I discussed (tape consistency) we need rules for each tape position and each time step, ie N^2, times the square of the number of possible symbols (a constant!). And the bulk of each term so introduced is (essentially) constant. So that only contributes an N^2 to the bulk of our eventual instance of 3-SAT. And because tabulating those terms is a simple loop the TM can clearly do it in polynomial time, perhaps around N^4.