Math 3502: Combinatorial Optimisation. Solutions 3.

1. Certificates are: (i) subsets of A and (ii) paths.

2. Suppose $L_i \in NP$ and can be decided by the TM M_i in time p_i where p_i is a polynomial for $i = 1$, 2.

We give a new TM as follows. Given input x and certificates of the form (C_1, C_2) we first run M_1 on the input ignoring C_1 and then run M_2 on it ignoring C_1. For '$\cup$' we need that one of the machines accepts, for '$\cap$' we need that both do for our new TM to accept. The machine runs in $p_1 + p_2$ steps, which is still polynomial.

3. The problem is in NP because the accepting computation is a certificate. In order to show it is complete we have to show that if $L \in NP$ we have $L \leq_p BH$.

Let $L \in NP$, M be a Turing machine for L and p a polynomial. Now consider the function $f(x) = (M, x, 1^{p(|x|)})$. Then f is computable in polynomial time because M is independent of x, x can be written in time $|x|$ and the string of 1s in time $O(p(|x|))$.

Also, $x \in L$ if and only if there is a certificate $y \in \Sigma^*$ such that M accepts the input (x, y) in time $p(|x|)$. But by the definition of BH this is true if and only if $(M, x, 1^{p(|z|)}) \in BH$. So $x \in L$ if and only if $f(x) \in BH$).

4. Let $\phi = \bigwedge_i \bigvee_j l_{ij}$ be in DNF and for each i set $C_i = \bigvee_j l_{ij}$. Now ϕ is satisfiable iff (at least) one C_i is satisfiable. But each C_i is satisfiable iff it does not contain literals l and l' with $l' = \neg l$. For any clause C_i, this can be checked in polynomial time. To check satisfiability of ϕ, it is hence enough to check satisfiability of the Ci separately. This can be done in polynomial time.

(Explanation of the difference with CNF: Since we know that SAT is NP-complete and we don't know whether $P = NP$ this shows us that we do not have a polynomial time algorithm to convert between CNF and DNF.)