Lecture 11: Some More NP-Complete Problems.

We recalled that by the transitivity of the polynomial reduction relation, to prove a decision problem π complete, we need only to (a) show it is in NP and (b) prove that for some NP-complete problem π' we have $\pi' <_P \pi$.We used this to show that a number of problems were NP-complete.

To begin we noted that the problem of determining if an IP is feasible is NP-complete, even when we restrict all the variables to be zero or one. We noted that the certificate for such a problem was a feasible solution but that we needed the non-trivial result (which we will not even state or prove) that this solution had sized bounded by a polynomial in the input in order to show this certificate could be verified in polynomial time. To complete the proof we noted that our reduction from SAT to IP-feasibility was polynomial.
We had an integer variable y_i for each Boolean variable x_i. We had the constraints: $y_i \geq 0$, $y_i \leq 1$, and for each clause C_j in the formula:

$$\sum_{x_i \in C_j} y_i + \sum_{not(x_i) \in C_j} (1 - y_i) \geq 1$$

These constrains ensured that feasible solutions were in 1 to 1 correspondence with satisfying truth assignments.

We also noted that 3-SAT is NP-complete. Clearly 3-Sat is in NP (the certificate is a satisfying truth assignment- given this we just need to check if every clause contains a true literal which can be done in linear time). We have already proved that SAT can be reduced to 3-SAT by replacing each clause with a set of clauses of size 3 in such a way that the new formula obtained will be satisfiable if and only if the original formula was. This reduction can easily be done in time linear in the sum of the sizes of the clauses.

We then showed the NP-completeness of the decision problem CLIQUE where an instance is a graph G and integer k, and we ask: does G contain a CLIQUE of size k? CLIQUE is clearly in NP (given as a certificate the names of k vertices in a clique we need only ensure that this list really does have k elements and that there is an edge between every two of them- this can be done in $O(|V|^2)$ time). We showed that 3-SAT polynomially reduces to CLIQUE using the reduction of Section 34.5.1 of the text.

The same proof, replacing edges by non-edges shows the NP-completeness of the decision problem STABLE_SET where an instance is a graph G and integer k, and we ask: does G contain a stable set of size k?

We next observed that the decision problem VERTEX COVER is NP-complete. In this decision problem the input is a graph G and an integer k, and we are asked if there is a vertex cover C of size k in G (i.e. a set C such that every edge of G has an endpoint in C). . A polynomially-verifiable certificate for this problem consists of a list of vertices in the cover. To check the cover exists, we then simply need to verify that this set of vertices has size k and that every edge has at least one endpoint in it. We can polynomially reduce an instance of stable set to an instance of vertex cover, since C is a vertex cover in G if and only if V-C is a stable set in G. So, we reduce an instance (G,k) of stable set to the instance (G,|V|-k) of vertex cover, as discussed in Section 34.5.2 of the text.

We next considered the decision problem 3-COL where an input is a graph G and the question is, can we label the vertices of G from {1,2,3} so that no edge has both endpoints the same colour. Clearly, the desired colouring provides a poynomially verifiable certificate for this problem so it is in NP. I provided the last piece in a polynomial reduction from 3-SAT to 3-COL which we had started in a previous class. The graph corresponding to a 3-SAT instance with variable set X and clauses C_1 to C_j has vertex set which is the union of the 2n literals, three special vertices {f,t,o}, and $\coprod_{i=1}^{j}\{v_i, w_i, y_i, z_i, a_i, b_i\}$. Its edge set consists of a triangle on {f,t,o}, edges from o to every literal, an edge between every literal and its negation,and for every i, a triangle on $\{z_i,a_i,b_i\}$, the edges $\{v_iz_i,w_ia_i,y_ib_i\}$, edges from the first literal of C_i to v_i, from the second literal of C_i to w_i, from the third literal of C_i to y_i, and from t to each of v_i,w_i,y_i.

We saw that this graph has a 3-colouring precisely if the 3-SAT instance has a satisfying truth assignment. Clearly we can construct the graph in timepolynomial in the size of the 3-SAT instance.

We then considered Knapsack and Partition. An instance of Partition conists of n integers $a_1,\ldots,a_n$. We are asked if we can partition these integers into two subsets such that the sum of the integers in the two sets is the same.

An instance of Partition reduces to an instance of Knapsack where $w_i=v_i=a_i$ and $W=V=\frac{1}{2}\sum_1^n a_i$ are both exactly half the sum of the weight of the items. As with any other instance of Knapsack, we can certify a yes answer by providing the indices of

the items to be put in the knapsack. In linear time, we can then check if the sum of their weights is at most W and the sum of their values at least V. So, our reduction from Partition to Knapsack shows that if Partition is NP-complete so is Knapsack, and that to prove Partition NP-complete we need only polynomially reduce an NP-complete problem to it. We give here a reduction from 3-SAT to Partition which is similar to that given to the SUBSET SUM problem in Section 34.5.5 of the text.

We considered an instance of 3-SAT with variable $x_1,\ldots,x_n$ and m clauses, $C_1,\ldots,C_m$, each of which contains three literals. . Our instance of PARTITION has 2n+2m+1 items. For i between 1 and n, item i corresponds to x_i and letting I(i) be $\{j|x_i$ is a literal of $C_j\}$ has value $7^{i+m-1}+\sum_{j \text{ in } I(i)} 7^{j-1}$ while item i+n corresponds to $\neg x_i$ and letting J(i) be $\{j|\ \neg x_i$ is a literal of $C_j\}$ has value $7^{i+m-1}+\sum_{j \text{ in } J(i)} 7^{j-1}$. For j between 1 and m, item 2n+j and2n+j+m correspond to clause j and have value 7^{j-1}. Finally the last item has value $\sum_{j=1}^{m} 7^{j-1}$.
Even though the numbers we construct are exponentially big in terms of the input SAT instance we can do this reduction in polynomial time because we only use O(log k) symbols to represent an integer k.
We note that our weights were chosen so that when we do addition in base 7, there is no carry over. This is crucial to the verification that the reduction is correct.
Given, a satisfying truth assignment for the 3-SAT instance, we will choose to put on one side of the partition all of the items which correspond to true literals under the assignment. On the other side of the partition we put the last element and all of the variables corresponding to false literals. Finally for each clause C_j, the number of items corresponding to clause C_j which are put on the same side as the true literals is equal to the number of false literals in the clause (which is between 0 and 2). The sum of the weights of these items is W.
Given a partition, each side of which has weight exactly half of the total weight of all the items, setting the literals corresponding to items on the same side of the partition as the last item to false yields a satisfying truth assignment.

Finally, we showed that Hamilton Cycle is NP-complete using the reduction of Section 34.5.3 of the text.