Theory of Computation, CSCI 438 spring 2023
Class nondeterministic polynomial time, NP, pg. 292-298
NP-Completeness, pg. 299-311
April 27

1. Prove that HAMPATH ∈ NP

    HAMPATH = {<G, s, t> | G is a directed graph that contains a directed path from source vertex s to target vertex t and the path passes through every vertex in G exactly once}

Given a path, it is easy to verify if this is a HAMPATH path. (This is called a verifier.)

Verifying the existence of a Hamiltonian path is much easier than finding the path. HAMPATH has polynomial verifiability.

This is not true for all problems. For instance, the complement of HAMPATH does not have polynomial verifiability. If we can somehow determine that a graph does not have a Hamiltonian path, there isn't an easy way to verify this.

A certificate for a string <G,s,t>∈HAMPATH is a path from s to t.

NP Turing decider for HAMPATH (page 294)
N1 = "On input <G,s,t> where G is a directed graph with nodes s and t:

1. Write a list of m numbers, p_1, …. p_m, where m is the number of nodes in G. Each number in the list is nondeterministically selected to be between 1 and m.
2. Check for repetitions in the list. If any are found, reject.
3. Check whether $s=p_1$ and $t=p_m$. If either fail, reject.
4. For each i between 1 and m-1, check whether (p_i,p_{i+1}) is an edge of G. If any are not, reject. Otherwise, all tests have been passed, so accept."

2. Prove that CLIQUE∈NP

    CLIQUE = {<G, k> | G is an undirected graph with a k-clique}

        Clique – all nodes in the clique are completely connected

        k-clique – a clique with k nodes.

A certificate is a k-clique.

Verifier for CLIQUE given a certificate c, a k-clique.
V = "On input <<G, k>,c> where G is an undirected graph, k is an integer, and c is a list of nodes, representing a k-clique:

1. If k exceeds n (the number of vertices in G), reject.
2. If c doesn't contain k unique vertices, reject.
3. For each vertex c_i in c
    If c_i isn't connected to every other vertex in c, reject.
4. Accept"

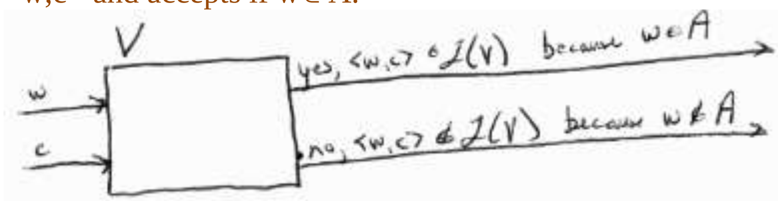
Running time:
Step 1 runs in constant time.
Step 2 Loop through the vertices in c, marking them. At most n steps.
Step 3 Go through the loop at most n times
    Loop through edges, marking vertices in c that are adjacent to c_i. At most n^2.
Step 4 runs in constant time.

$O(c+n+n*n^2+c) = O(n^3)$
Thus, V∈P so CLIQUE∈NP.

3. Theorem 7.20 (page 294) A language is in NP iff it is decided by some nondeterministic polynomial time Turing machine.

Proof:

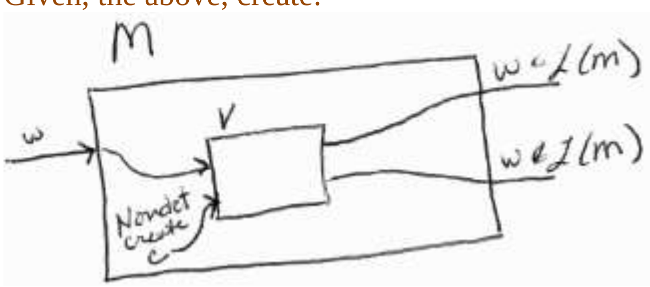
Only if (⇒) Say that a language A is in NP. Then there is a polynomial time verifier for the language A. That is, there is a polynomial time algorithm V which uses a certificate c to determine membership in the language. That is, V takes in <w,c> and accepts if w∈ A.



Define the nondeterministic polynomial time Turing machine for the language as N = “On input <w> of length n:

1. Nondeterministically select a string c.
2. Run V on <w, c>.
3. If V accepts, accept; otherwise, reject.”

Given, the above, create:



Running time:

Step 1 The certificate c, can’t exceed length n^k, for some constant k, since V is a polynomial time verifier and n^k is all the verifier would have time to access.

Step 2 V runs in polynomial time.

Step 3 Runs in constant time.

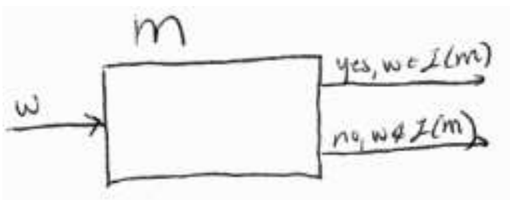
$O(n^k+n^k+c) = O(n^k)$ for some k

Thus, M is a nondeterministic TM that runs in polynomial time.

If (⇐) Say that a language is decided by some nondeterministic polynomial time Turing machine M. Since it is decided, there is some path which runs in $f(n^k)$ time and gives an accept or reject for the language.

Construct a verifier V as follows:
V = "On input <w, c> where w and c are strings:
1. Simulate M on input w. At each nondeterministic choice, use what is given in the certificate c to decide what branch to take. (That is, use the certificate in the same way the progress tape was used in simulating a nondeterministic TM by a deterministic one.)
2. If this branch of M's computation accepts, accept. Otherwise reject."

Have:



Build:

Running time:
Step 1 Runs in polynomial time, because M ran in polynomial time (that is, the maximum number of steps that M used on may branch was of polynomial length).
Step 3 Runs in constant time.

Thus there is a polynomial time algorithm V which uses a certificate c to determine membership in the language. Since ths is a polynomial time verifier, the language is in NP.

Problems in NP:

HAMPATH = {<G, s, t> | G is a directed graph that contains a directed path from source vertex s to target vertex t and the path passes through every vertex in G exactly once}

COMPOSITES = {x | x=pq, for integers p, q > 1}

CLIQUE = {<G, k> | G is an undirected graph with a k-clique}
Clique – all nodes in the clique are completely connected
k-clique – a clique with k nodes.

SUBSET-SUM = {<S, t> | S = {$x_1, \ldots, x_k$}, and for some
{$y_1, \ldots, y_k$}$\subseteq${$x_1, \ldots, x_k$}$\Sigma y_i = t$ }
Note that S and its subset are multi-sets.