Given this 3-SAT problem:

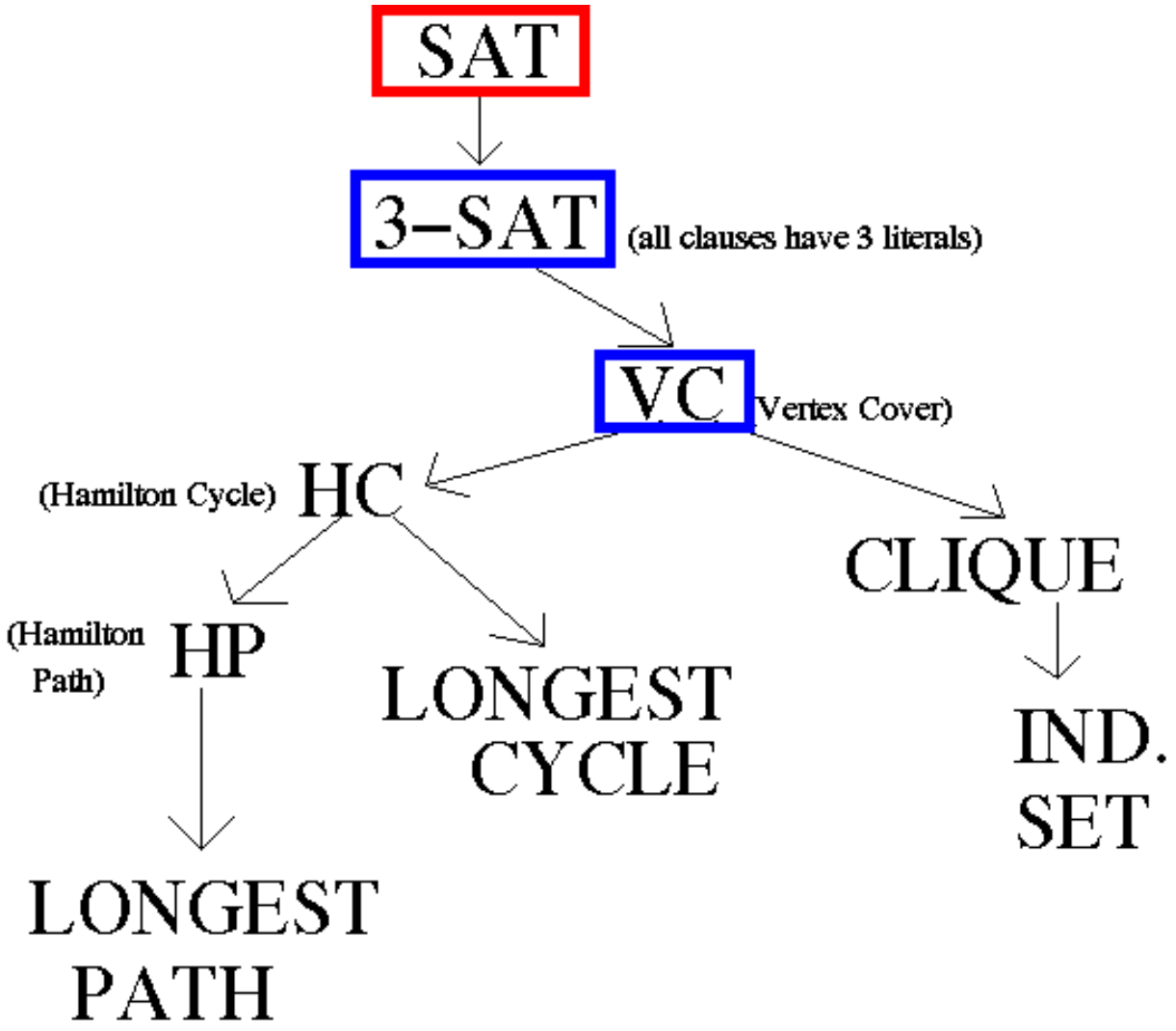
(x1 or x2 or x3) AND

(¬x1 or ¬x2 or ¬x2) AND

(¬x3 or ¬x1 or x2)

1. Draw the graph that you would use if you want to solve this problem using an algorithm for vertex cover.

2. What size of vertex cover would you look for in order to decide if this 3-SAT system is satisfiable?



History of NP−completeness Reductions

A problem Q in NP is NP-complete if the existence of a polynomial time algorithm for Q implies the existence of a polynomial time algorithm for all problems in NP.

How do we prove SAT is NP-complete?

Theorem: **SAT (Satisfiability) is in NP.**

Proof.

Variables: $u_1, u_2, u_3, \ldots u_n$.

Certificate: sequence $b_1, b_2, b_3, \ldots b_n$ of true/false values.

Check for each clause if at least one literal is true.

Time: $O(n+k)$ where k is the number of occurrences of a literal in a clause.

Every problem Q which is in NP has a non-deterministic TM M which:

1. Guesses a certificate of length n.

2. Checks the certificate in p(n) time.

Start state- q_0

Final state- the computation will terminate in a special state q_N when the certificate is not correct and in a state q_y if the certificate is correct.

Cook's Theorem: SAT is NP-complete.

Idea of proof:

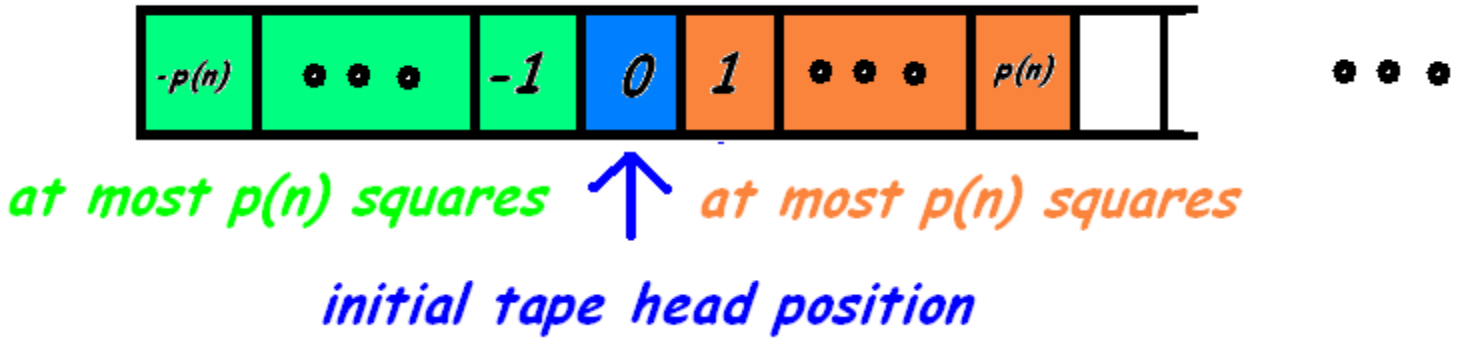
Construct a collection of clauses whose length is polynomial in p(n) and hence is also polynomial in n.

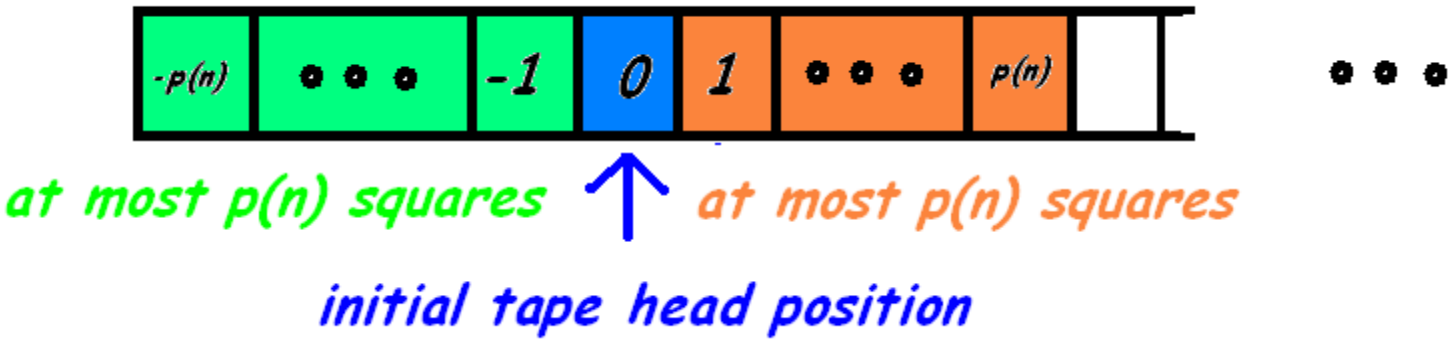
A satisfying truth assignment corresponds to a computation of M which non-deterministically guesses a correct certificate and checks it.

The number of tape squares used is bounded:

p(n)= maximum number of steps that M takes on an input of size n.

Tape squares used in this computation:





Variable	Range	Meaning
Q[i, k]	$0 \le i \le p(n)$, $1 \le k \le$ # states	At time i, M is in state q_k
H[i,j]	$0 \le i \le p(n)$, $-p(n) \le j \le p(n)$	Scanning square j at time i
S[i, j, k]	$0 \le i \le p(n)$, $-p(n) \le j \le p(n)$, $1 \le k \le$ # symbols	Time i, square j contains symbol s_k

Clauses to make sure that satisfying assignments correspond to valid computations:

Recall: Q[i, k]- At time i, M is in state q_k

Group G1: At time i, M is in exactly one state:

(a) M is in at least one state at time i:

for each i, $0 \le i \le p(n)$, add a clause (s= # states):

(Q[i, 1] or Q[i, 2] or ... or Q[i, s])

(b) M is in at most one state at time i: For each time i and pair j, k of states,

$0 \le i \le p(n)$, $1 \le j < k \le s$, add a clause

(not Q[i, j] or not Q[i, k])

Clause Group	Restriction
G1	At time i, M is in exactly one state.
G2	At time i, the tape head scans exactly one square.
G3	At time i, each square contains exactly one symbol.
G4	At time 0, M is in initial checking stage.
G5	By time p(n), M enters state q_y.
G6	Configuration at time i+1 follows the one at time i by one application of the transition relation.

Original problem: Input of size n.

The size of this SAT system has length which is at most some polynomial q(n).

If there is an $O(n^c)$ algorithm for SAT then there is an $O(q(n)^c)$ algorithm for this other problem which is in NP.

Therefore, if there is a polynomial time algorithm for SAT, there is a polynomial time algorithm for every problem in NP.

Conclusion: SAT is NP-complete.

Summary of class contents: