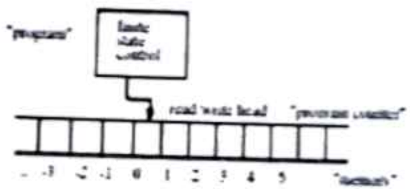
Turing Machines and Cook's Theorem

Cook's Theorem proves that satisfiability is NP-complete by reducing all non-deterministic Turing machines to SAT. Each Turing machine has access to a two-way infinite tape (read/write) and a finite state control, which serves as the program.

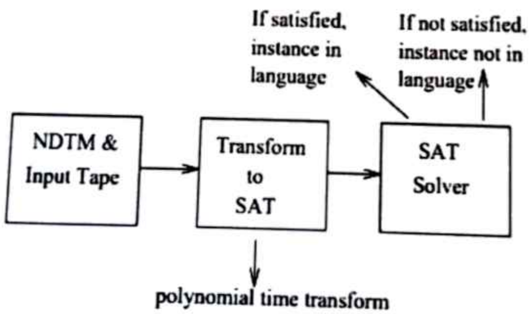


A program for a non-deterministic TM is:

1. Space on the tape for guessing a solution and certificate to permit verification.

2. A finite set of tape symbols
3. A finite set of states Θ for the machine, including the start state q_0 and final states Z_{yes}, Z_{no}
4. A transition function, which takes the current machine state, and current tape symbol and returns the new state, symbol, and head position.

We know a problem is in NP if we have a NDTM program to solve it in worst-case time $p[n]$, where p is a polynomial and n is the size of the input.



If a polynomial time transform exists, then SAT must be NP-complete, since a polynomial solution to SAT gives a polynomial time algorithm to anything in NP.
Our transformation will use boolean variables to maintain the state of the TM:

	Variable	Range	Intended meaning
	$Q[i,k]$	$0 \le i \le p(n)$ $0 \le k \le r$	At time i M is in state q_k
	$H[i,j]$	$0 \le i \le p(n)$ $p(n) \le j \le p(n)+1$	At time i the read-write head is scanning tape square j
	$S[i,j,k]$	$0 \le i \le p(n)$ $p(n) \le j \le p(n)+1$	At time i the contents of tape square j is symbol S_k
		$0 \le k \le v$	

Note that there are $rp(n) + 2p^2(n) + 2p^2(n)v$ literals, a polynomial number if $p(n)$ is polynomial.

We will now have to add clauses to ensure that these variables takes or the values as in the TM computation.

The group 6 clauses enforce the transition function of the machine. If the read-write head is not on tape square j at time i, it doesn't change ....

There are $O(p(^2(n))$ literals and $O(p^2(n))$ clauses in all, so the transformation is done in polynomial time!