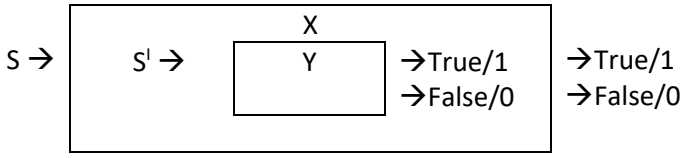
4 SAT: There exists a state of all literals that results in every clause equalling ‘true’, and one clause has, at most, 4 literals.

3 SAT: There exists a state of all literals that results in every clause equalling ‘true’, and one clause has, at most, 3 literals.

Assume Y exists that solves 4 SAT. we will construct machine X that solves 3 SAT using Y.



X takes S, that is the 3 SAT problem (the set of clauses), and constructs S^{I}. S^{I} is the 3 SAT problem that was S but with one of the literals duplicated (making the max number of literals in the problem 4, thus 4 SAT). X passes S^{I} to Y and accepts if Y accepts.

We will show X will accept (say true/1) when run on S iff S results in true/1.

- Assume S results in true/1, then S^{I} will always result in true/1. Because the literals in the 3 SAT problem result in true/1, adding an additional literal that is a duplicate of one of the existing literals will yield the same result. If Y accepts, the X accepts, when run on S.
- Assume X accepts S^{I} when run on S. Then, Y accepts S^{I}, which means that S^{I} always results in true/1 because the additional literal in S^{I} is the same as one of the existing literals from S. Therefore, S will result in true/1 because S^{I} is the same as S with a duplicated literal.

Hence, X solves 3 SAT, and since X is a machine that solves 4 SAT. 3 SAT can be reduced to 4 SAT, making 4 SAT at least as hard as 3 SAT, a NP hard problem. Thus, 4 SAT is NP hard as well.

We can easily check if a solution to 4 SAT is valid by subbing in the values into the clauses. The time to check increases linearly based on the number of clauses that exist.

Thus, 4 SAT is both NP hard and in NP (Can be checked in deterministic polynomial time), thus 4 SAT is NP Complete.

QED