The proof of 4-SAT's NP Completeness is by the support of 3-SAT's NP Completeness. Following is the sequence of statements leading to the desired proof:

1) 4-SAT belongs to the class of Nondeterministic Polynomial (NP) type algorithms, hence a NP time algorithm can be written that takes the input of a 4-SAT instance along with a suggested truth assignment.

2) The 4-SAT instance is validated against the suggested truth assignment in polynomial time. The output of the algorithm is 'YES' if the validation evaluates to truth, 'NO' otherwise.

3) To prove the HARD-ness of 4-SAT with reduce 4-SAT from 3-SAT as the following:

4) Let **p** be an instance of the 3-SAT problem.

5) **p** is polynomial time convertable to **q** which is an instance of 4-SAT by turning each clause (a ? b ? c) in **p** to

(a ? b ? c ? **d**) ? (a ? b ? c ?¬**d**), where **d** is a new variable.

6) If a given clause (a ? b ? c) is satisfied by a truth assignment, then (a ? b ? c ? **d**) ? (a ? b ? c ?¬**d**) is also satisfied by the same truth assignment for an arbitrary value of **d**. Thus if **p** is satisfiable, so is **q**.

7) Suppose **q** is satisfied by a truth assignment T. Then (a ? b ? c ? **d**) ? (a ? b ? c ?¬**d**) must be true under T as well. Also, as **d** and **¬d** assume different truth values, then (a ? b ? c) must also be true under T. Thus **p** is satisfiable.

Hence 4-SAT is NP-Complete