- SAT to 3-SAT reduction : Analysis.
  - Input size can be seen as proportional to number of clauses (m) and no. of variables (n).
  - What is the increase in input size?
- No: of clauses:
  1. literal clause : 1 becomes 4 clauses.

- 2-literal clause becomes 2 clauses.
- 4-literal clause also becomes 2 clauses.
- For $k > 4$, k-literal clause becomes $k-2$ clauses.
- Number of variables:

  1-literal clause: 2 extra variables.

- 2-literal clause: one extra variable.
- $k \geq 4$: $k-3$ extra variables.

- So the size of the input does not blow up exponentially. It remains a polynomial in n, m and k, where k is the max. no. of literals in a clause.

- We proved that

$$SAT \leq_p 3\text{-}SAT.$$

Or, if $3\text{-}SAT \in P$, then $SAT \in P$.

(In other words)

If $SAT \notin P$, then $3\text{-}SAT \notin P$.

- NP-Complete problems:

A language (a decision problem) B is said to be NP-Complete if

(i) B is in NP, and

(ii) if $B \in P$, then $P = NP$.

- The second condition is equivalent to saying that

  every $A \in NP$ is polynomial-time reducible to B.

- Or, $\forall A \in NP$, $A \leq_p B$.

(in that case, if $B \in P$, then A also in P).

• Another term for this (second) condition is "B is NP-hard".

Def^n: B is NP-hard if

$B \in P$ means $P = NP$.

[Alternatively, If B has a poly. time algo, every problem in NP has poly. time algos.]

• "NP-hard" problems are, thus, problems that are "unlikely" to have polynomial time algorithms. Or, "hard" problems asfar as having poly. time algorithms are concerned.

- But how to prove?

For SAT, we will do this proof, that if $SAT \in P$, then every $A \in NP$ is also polynomial time solvable.

- For now, we will assume this fact.

ie, "SAT is NP-hard."

. <u>Theorem:</u> SAT is NP-complete.

It means SAT $\in$ NP. (Easy).

And SAT is NP-hard. (Proof Later).

[Second part means that

If SAT $\in$ P, then P = NP].

- Once we have this theorem, we can figure out which other problems are NP-complete, and which other problems are NP-hard.
- For instance, we can immediately say that 3-SAT is NP-complete!

SAT
3-SAT

(NP-hard problems)

- We can "grow" this box by giving polynomial-time reductions.

  Say, if 3-SAT $\leq_p$ CLIQUE, we can add CLIQUE to it.

· NP-hard:

- Hard, to the extent that if it can be solved in polynomial time, then any NP problem can be solved so. Or, the class NP 'collapses' to P.

· We will see more such problems.