IIIT
BHUBANESWAR
I I I T Bhubaneswar
Imagine, Innovate, Inspire
(A University established by Government of Odisha)

MTech CSE – 1st Semester

Student Name: SIDDHARTH KUMAR
Student ID: A125021

Question 13

Is the 3-SAT (3-CNF-SAT) problem NP-hard? Justify your answer.

Answer:

Direct Answer

Yes, the 3-SAT problem is NP-hard. In fact, it is a classic example of an NP-complete problem.

Relevant Background

The SAT Problem

The Boolean Satisfiability problem (SAT) asks whether there exists an assignment of truth values to variables such that a given Boolean formula evaluates to true. The Cook–Levin Theorem established that SAT was the first NP-complete problem.

Definition of 3-SAT

The 3-SAT problem is a restricted form of SAT in which:

- the formula is in conjunctive normal form (CNF), and
- each clause contains exactly three literals.

A 3-SAT formula can be written as:

$$\varphi = \bigwedge_{i=1}^{m} (\ell_{i1} \lor \ell_{i2} \lor \ell_{i3}),$$

where each literal ℓ_{ij} is either a Boolean variable or its negation.

Understanding NP-Hardness

A problem is said to be *NP-hard* if every problem in the complexity class **NP** can be reduced to it in polynomial time. If a problem is both NP-hard and a member of **NP**, it is classified as NP-complete.

3-SAT Belongs to NP

Given a proposed truth assignment for a 3-SAT instance, each clause can be verified in constant time. Since the total number of clauses is polynomial in the input size, the entire formula can be checked in polynomial time.

Hence,

$$\text{3-SAT} \in \mathbf{NP}.$$

Polynomial-Time Reduction from SAT to 3-SAT

To prove NP-hardness, it is sufficient to show that:

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

That is, any instance of SAT can be transformed into an equivalent instance of 3-SAT in polynomial time.

Transforming Long Clauses

Consider a CNF clause containing more than three literals:

$$(x_1 \lor x_2 \lor x_3 \lor \cdots \lor x_k).$$

By introducing new auxiliary variables, this clause can be rewritten as a conjunction of 3-literal clauses:

$$(x_1 \lor x_2 \lor y_1) \;\land\; (\neg y_1 \lor x_3 \lor y_2) \;\land\; \cdots \;\land\; (\neg y_{k-3} \lor x_{k-1} \lor x_k).$$

This transformation preserves satisfiability and increases the formula size only linearly.

Handling Short Clauses

Clauses containing fewer than three literals can be padded by duplicating existing literals:

- $(x_1 \vee x_2)$ becomes $(x_1 \vee x_2 \vee x_2)$,
- (x_3) becomes $(x_3 \vee x_3 \vee x_3)$.

These modifications do not alter the logical meaning of the clauses.

Correctness and Complexity of the Reduction

The constructed 3-CNF formula is satisfiable if and only if the original CNF formula is satisfiable. Moreover:

- the number of new variables introduced is linear,
- the number of clauses grows linearly,
- the transformation runs in polynomial time.

Therefore,

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

Final Classification

Since:

- SAT is NP-complete,
- SAT can be reduced to 3-SAT in polynomial time, and
- 3-SAT belongs to **NP**,

we conclude that:

$$\text{3-SAT is NP-complete.}$$

As a consequence, 3-SAT is NP-hard.

Significance

The NP-hardness of 3-SAT is fundamental in computational complexity theory. It serves as a standard starting point for reductions to prove NP-hardness of many other problems. Despite its strict structural restrictions, 3-SAT retains the full computational difficulty of general SAT.