3SAT IS POLYNOMIAL TIME REDUCIBLE TO CLIQUE

Theorem 7.32

Presented by Rohan Doshi

INTRODUCTION

- What is 3SAT?
- What is CLIQUE?

INTRODUCTION

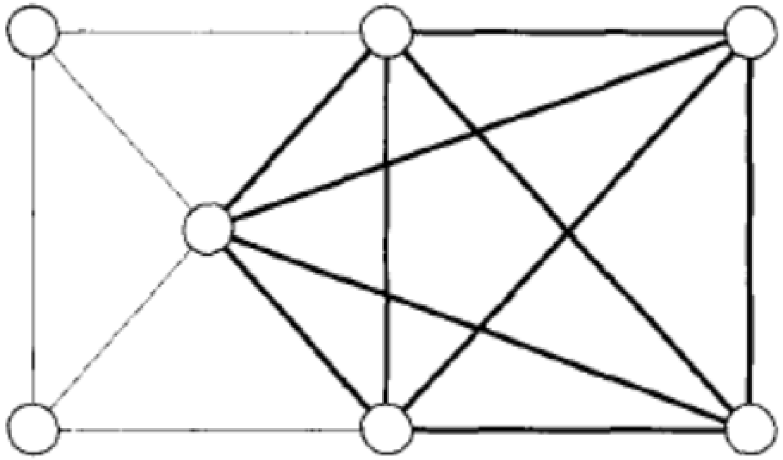
- Boolean formula is an expression involving Boolean variables and operations.
- For example: $\phi = (\overline{x} \wedge y) \vee (x \wedge \overline{z})$
- Boolean formula is satisfiable if some assignment of 0s and 1s to the variables makes the formula evaluate to 1.
- The preceding formula is satisfiable because the assignment x = 0, y = 1, and z = 0 evaluates to 1.
- The satisfiability *problem* is to test whether a Boolean formula is satisfiable.

INTRODUCTION

- A literal is a Boolean variable or a negated Boolean variable, as in x or $\overline{x}$.
- A clause is several literals connected with $\vee$s as in
  $(x_1 \vee \overline{x_2} \vee \overline{x_3} \vee x_4)$
- A Boolean formula is in conjunctive normal form, called a CNF-formula, if it comprises several clauses connected with $\wedge$s, as in
  $(x_1 \vee \overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6})$.
- It is a 3cnf-formula if all the clauses have three literals, as in
  $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_5} \vee x_6) \wedge (x_3 \vee \overline{x_6} \vee x_4) \wedge (x_4 \vee x_5 \vee x_6)$

INTRODUCTION

- A clique in an undirected graph is a subgraph, wherein every two nodes are connected by an edge.
- A k-clique is a clique that contains k nodes.
- For example:



- The clique problem is to determine whether a graph contains a clique of a specified size.

THEOREM

Proof Idea:

- The reduction of 3SAT to clique can be demonstrated by converting the formula into a graph.
- In the constructed graphs, cliques of a specified size correspond to satisfying assignments of the formula.
- Structures within the graph are designed to mimic the behavior of the variables and clauses.

THEOREM

Proof:

- Let the Boolean formula be:

$$\phi = (a_1 \vee b_1 \vee c_1) \ \wedge \ (a_2 \vee b_2 \vee c_2) \ \wedge \quad \cdots \quad \wedge \ (a_k \vee b_k \vee c_k).$$

- We reduce this formula into an undirected graph G.
- The nodes in G are organized into k groups of three nodes each called the *triples*, *t1, . . .*, tk. Each triple corresponds to one of the clauses in the formula, and each node in a triple corresponds to a literal in the associated clause.

THEOREM

Proof:

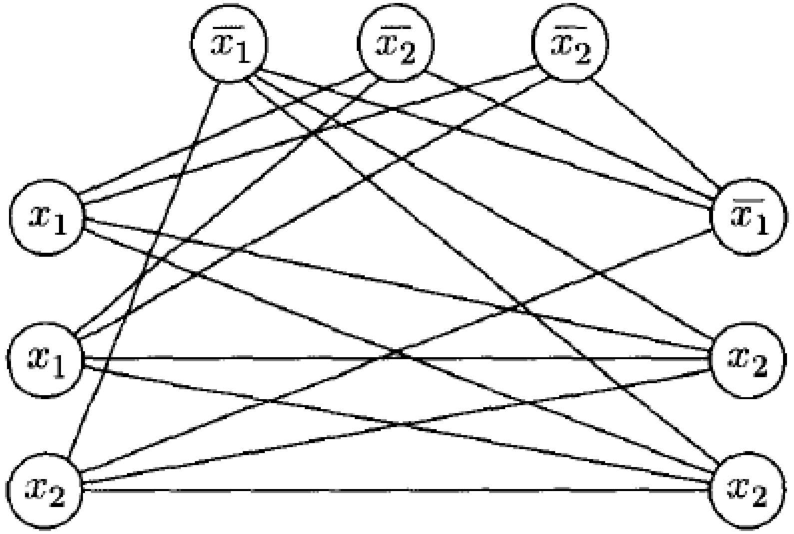
- The edges of G connect all but two types of pairs of nodes in G. No edge is present between nodes in the same triple and no edge is present between two nodes which are complementary.
- For example, let us reduce the following formula into a graph.

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2).$$

THEOREM

Proof:

- Resultant graph is as follows:



THEOREM

Proof:

- Now let us prove that Boolean formula is satisfiable iff G has k-clique.
- Let us consider the first direction i.e. suppose that ϕ is satisfiable.
- In that satisfying assignment, at least one literal is true in every clause. In each triple of *G,* we select one node corresponding to a true literal in the satisfying assignment.
- If more than one literal is true in a particular clause, we choose one of the true literals arbitrarily.
- The nodes just selected form a k-clique. The number of nodes selected is k, because we chose one for each of the k triples.

THEOREM

Proof:

- Each pair of selected nodes is joined by an edge because no pair fits one of the exceptions described previously.
- They could not be from the same triple because we selected only one node per triple.
- They could not have contradictory labels because the associated literals were both true in the satisfying assignment. Therefore G contains a k-clique.

THEOREM

Proof:

- Suppose that G has a k-clique.
- No two of the clique's nodes occur in the same triple because nodes in the same triple aren't connected by edges.
- Therefore each of the k triples contains exactly one of the k clique nodes.
- We assign truth values to the variables so that each literal labeling a clique node is made true.
- This assignment to the variables satisfies ϕ because each triple contains a clique node and hence each clause contains a literal that is assigned TRUE.
- Therefore ϕ is satisfiable.

THANK YOU！