Prove that 2SAT is in P

Shobhit Chaurasia (11010179), Harshil Lodhi (11010121), Hitesh Arora (11010122)

We propose the following polynomial time algorithm to decide whether a given 2SAT expression is satisfiable or not.

Consider a 2CNF formula Ψ with *n* variables and *m* clauses. We will show that 2SAT is polynomial-time decidable by constructing a graph and using path searches in the graph.

CONSTUCTION

Create a graph G = (V, E) with *2n* vertices. Intuitively, each vertex resembles a true or not true literal for each variable in Ψ. For each clause $(a \vee b)$ in Ψ, where 'a' and 'b' are literals, create a directed edge from '$\neg$ a' to 'b' and from '$\neg$ b' to 'a'. These edges mean that if 'a' is not true, then 'b' must be true and vice-versa. That is, there exists a directed edge (α, β) in G iff there exists a clause $(\neg \alpha \vee \beta)$ in Ψ.

For example, the following 2CNF Ψ leads to the graph in Fig.1

$$\Psi = (\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (z \vee y)$$

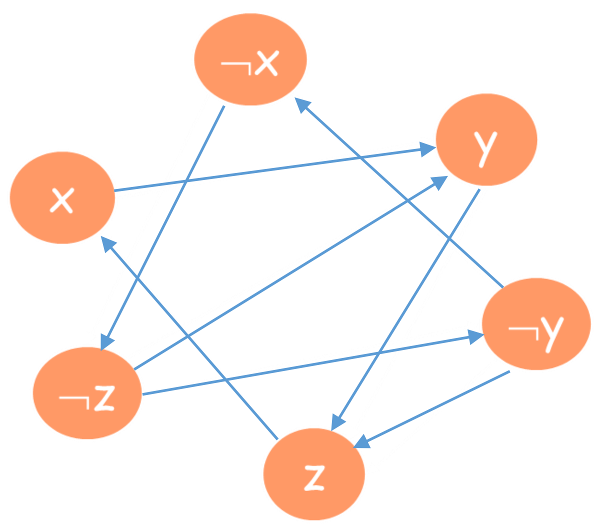

Fig. 1

CLAIM 1 – If G contains a path from α to β, then it also contains a path from $\neg\beta$ to $\neg\alpha$.

PROOF – Let the path from α to β be $\alpha \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots\ldots \rightarrow P_k \rightarrow \beta$.

Now, by construction of G, if there's an edge (X, Y), then there's also an edge ($\neg$Y, $\neg$X). Hence, the edges ($\neg\beta$,$\neg P_k$), ($\neg$Pk, $\neg P_{k-1}$), ..., ($\neg P_2$, $\neg P_1$), ($\neg P_1$,$\neg\alpha$). Hence, there is a path from $\neg\beta$ to $\neg\alpha$.

CLAIM 2 – A 2CNF formula Ψ is **unsatisfiable** iff there exists a variable x, such that:

1. there is a path from x to $\neg$x in the graph
2. there is a path from $\neg$x to x in the graph

PROOF – *(by contradiction)*

Suppose there are path(s) x to $\neg$x and $\neg$x to x for some variable x in G, but there also exists a satisfying assignment $\rho(x_1, x_2 \ldots x_n)$ for Ψ.

Case#1: Let $\rho(x_1, x_2 \ldots x_n)$ be such that x=TRUE.

Let the path x to $\neg$x be x-> .... -> α -> β -> .... -> $\neg$x.

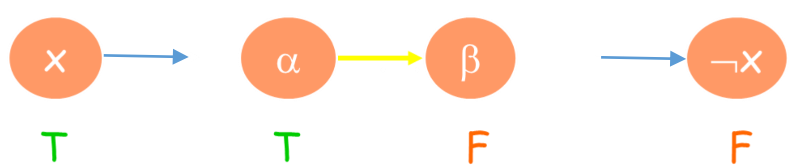

Fig. 2

Now, by construction, there is an edge between A to B in G iff there is a clause ($\neg$ A V B) in Ψ. The edge from A to B represents that if A is TRUE, then B must be TRUE (for the clause to be TRUE). Now since x is true, all literals in path from x to α (including α) must be TRUE. Similarly, all literals in the path from β to $\neg$x (including β) must be FALSE (because $\neg$x=FALSE). This results in an edge between α and β, with α = TRUE and β = FALSE. Consequently the clause ($\neg\alpha \vee \beta$) becomes FALSE, contradicting our assumption that there exists a satisfying assignment ρ(x1, x2.... xn) for Ψ.

Case#2: Let $\rho(x_1, x_2 \ldots x_n)$ be such that x=FALSE. *(Similar analysis)*

Hence, by checking for the existence of a x to $\neg$x and/or $\neg$x to x path in the G, we can decide whether the corresponding 2CNF expression Ψ is satisfiable or not. The existence of a path from one node to another can be determined by trivial graph traversal algorithms like **BREADTH FIRST SEARCH** or **DEPTH FIRST SEARCH**. Both BFS and DFS take polynomial time of O(V + E) time, where V = #vertices and E = #edges in G. *Hence proved that 2SAT is in P.*

COROLLARY

The same graph construction can be used to construct a satisfying assignment for Ψ (if it is satisfiable). The following Pseudo code highlights the algorithm.

1. Construct the graph G as described above and check if given 2CNF is satisfiable or not.
2. If given 2CNF is not satisfiable, return
3. Pick an unassigned literal α, with no path from α to $\neg\alpha$, and assign it TRUE.
4. Assign TURE to all reachable vertices of α and assign FALSE to their negations.
5. Repeat 3, 4 and 5 until all the vertices are assigned.