(1) Show that maximum independent set is polytime reducible to ILP.

Proof:

Maximum Independent set (MIS):

I/p : An undirected graph $G = (V, E)$

O/p : Largest subset $V' \subseteq V$ such that no two vertices in V' are adjacent. i.e. $(u,v) \in V'$ but $(u,v) \notin E$.

Integer Linear Programming (ILP):

I/p : A set of linear inequalities and an objective function with variables, i.e. A, B and objective f^n.

O/p : The values of the variables that maximize or minimize the objective f^n while satisfying the constraints i.e. x such that $Ax \leq B$.

Reduction from MIS to ILP:

→ To represent MIS as ILP we can use binary variables to model the selection of vertices in the independent set.

→ Let's define a binary variable x_i for each vertex $i \in V$

$x_i = 1$ if vertex i is included in j

$x_i = 0$ otherwise

Objective Function

→ Maximize the no. of vertices in Independent set.

∴ Maximize $\sum_{i \in V} x_i$

constraints:

→ Ensure that no two adjacent vertices are both included in Independent Set. For every edge $(i,j) \in E$, we add constraints.

$x_i + x_j \leq 1$ $\quad$ $x_i, x_j \in \{0,1\}$ for all $i,j \in V$

The above constraints guarantees that if $x_i = 1$ (vertex i is in IS) then $x_j = 0$ and vice-versa, thus ensuring no two adjacent vertices are both selected.

Polynomial Time Reduction:

→ To convert an instance of MIS into ILP, we simply need to

① create a variables x_i for each vertex $i \Rightarrow O(V)$

② set up objective $f^n \sum_{i \in V} x_i \Rightarrow O(V)$

③ for each edge $(i,j) \in E$, add constraint

$x_i + x_j \leq 1 \Rightarrow O(E)$

$\therefore$ overall time complexity $= O(V + V + E)$

$\simeq O(V+E)$

$\simeq$ polytime

$\therefore$ MIS $\leq_p$ ILP

(2) Show that independent set is polytime reducible to circuit SAT

Independent-Set

Given an undirected graph G(V,E), an integer k and subset S where $S \subseteq V$ and $|S| = k$. S is IS such that no two vertices in S are adjacent.

Circuit SAT

A decision problem of determining whether there exists an assignment of truth values to the given input-variables of a boolean circuit such that o/p evaluates to 1.

Claim – If circuit outputs 1 then S forms an independent set.

Proof:

(1) Variable Assignment

For each vertex $v_i \in V$ creates a boolean variable x_i such that:

$x_i = 1$ if v_i is part of independent set.

$x_i = 0$ if v_i is not included in independent set.

The variables $x_1, x_2, \ldots x_n$ where $n = |V|$, corresponds to the vertices.

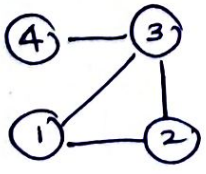
(2) Non-Adjacency constraint

For an edge $(v_i, v_j) \in E$ we need a constraint that includes only one of v_i, v_j. That is both can't come at a time.

∴ Clause is $\sim(x_i \wedge x_j) \Rightarrow (\overline{x_i} \vee \overline{x_j})$ [De Morgan's Law]

These are logical AND and OR operations which can be implemented using logic GATES.

ex. 1)



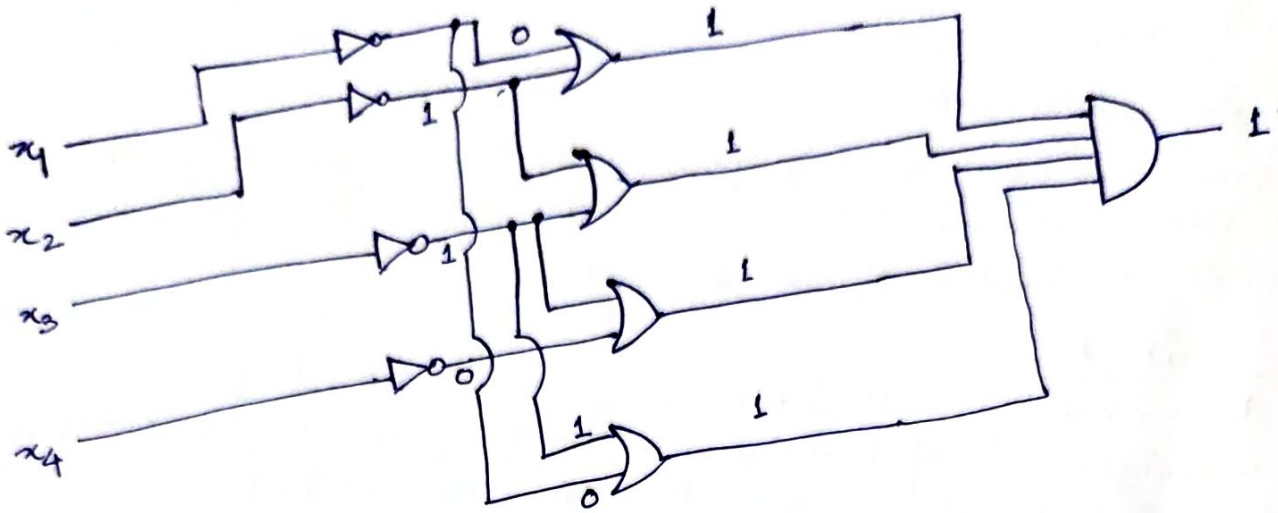
$k = 2$

$G = \{ V(1,2,3,4)$

$E[(1,2)(1,3)(1,2)(4,3)] \}$

$S = \{1,4\}$

x_1	x_2	x_3	x_4
1	0	0	1



circuit C1

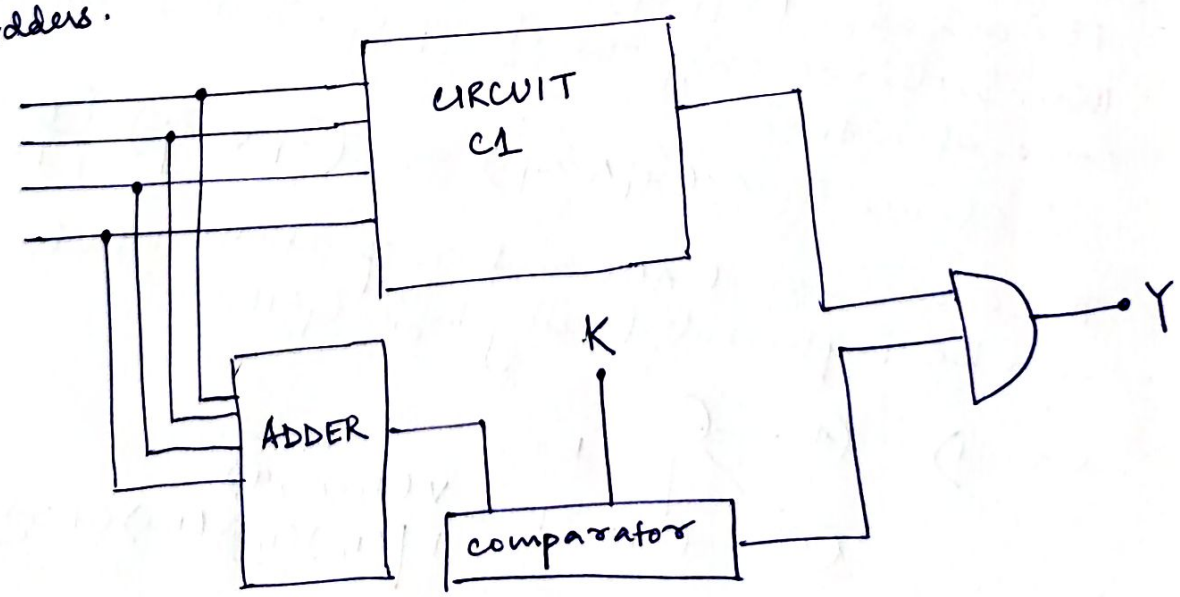
Above is a logical circuit handling non adjacent-vertices in given subset of the graph.

(3) size constraint

$|S| = k$ needs to be ensured in the independent set using Adder circuit.

$$\sum_{v \in V} x_v = k$$

We can build a sub-circuit that counts the no. of x_v variables set to 1 and circuit satisfies iff count = k. Such circuit can be built in polynomial time using Adders.



The output of the circuit will be a single boolean variable ~~variable~~ that determines whether independent set is valid. The output of the gate is satisfied if –

(i) Independent set is of size k

(ii) No two adjacent vertices are part of Independent set

Thus if circuit is satisfiable, this satisfying assignment corresponds to independent set.

Q. (2) Prove that the Quadratic Congruences problem belong to the class NP (Instance : positive integers a, b, c. Question : Is there a positive integer $x < c$ such that $x^2 \equiv a \bmod b$?

X : Instance of positive integers a, b, c

Y : Given x

Verifier

V(X, Y) :

(i) If $x <= 0$ or $x >= c$ output FALSE

as x must be +ve and $x < c$

— O(n) time, comparision assuming

X = n bit number.

(ii) compute $x^2 \pmod b$.

— $O(n^{1.59})$ using Karatsuba's Algorithm

(iii) compare $x^2 \pmod b$ with $a \pmod b$

If equal then return TRUE

— O(n) comparision

(iv) else return FALSE. — O(1)

Hence overall time to verify = $O(n^{1.59})$ which is polynomial time.

Hence quadratic congruence problem is NP.

(4) Prove that clique $\in$ NPC

Clique – A clique is a subset of vertices $V' \subseteq V$ such that every two distinct vertex in V' are connected by an edge in E.

Decision problem

Of a graph $G(V,E)$ and integers k, does G contain a clique of size k?

Proof for NP

X: Instance of graph $G(V,E)$ and k

Y: Set $V' \subseteq V(G)$

V(X,Y): Given a subset of V' of vertices then

(i) If $|V'| \neq k$ then return FALSE

(ii) If $|V'| = k$

```
for u in V':
    for v in V':
        if u≠v and (u,v) ∉ E
            return FALSE
return TRUE
```

So to verify clique we need $O(|V'|^2) = O(k^2)$ which is polynomial time. $\therefore$ clique $\in$ NP

Proof of NPC

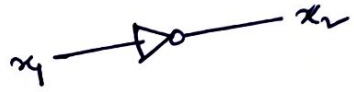
Using cook's Thm. we know that circuit SAT $\in$ NPC, so we know circuit SAT $\leq_p$ SAT $\leq_p$ 3CNF SAT $\leq_p$ clique

Proof circuitSAT $\leq_p$ SAT

1. Assign variable x_i for each input signal of a circuit
2. Assign variable x_o for output
3. Set up an if and only if formula for each gate.
   Let ϕ_k be the formula for k^{th} gate.
4. Let x_o be the final output wire of the circuit,

∴ CNF formula is $x_{o,f} \cdot \phi_1 \cdot \phi_2 \cdot \phi_3 \cdots$

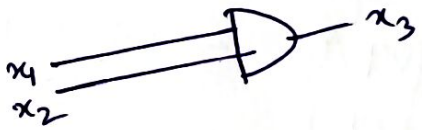
Reduction for NOT Gate



$$\phi = x_1 \leftrightarrow \bar{x}_2$$
$$= (\bar{x}_1 + \bar{x}_2)(x_1 + \bar{\bar{x}}_2)$$
$$= (\bar{x}_1 + \bar{x}_2)(x_1 + x_2)$$

So NOT gate can be reduced to CNF in polynomial time

Reduction for AND Gate



$$\phi = x_3 \leftrightarrow x_1 \cdot x_2$$
$$= (x_3 + \overline{x_1 \cdot x_2})(\bar{x}_3 + x_1 x_2)$$
$$= (x_3 + \bar{x}_1 + \bar{x}_2)(\bar{x}_3 + x_1)(\bar{x}_3 + x_2)$$

Hence AND Gate can be reduced to CNF in polytime as well.

Reduction for OR Gate

$$\phi = x_3 \leftrightarrow x_1 + x_2$$
$$= (x_3 + \overline{(x_1 + x_2)})(\bar{x}_3 + x_1 + x_2)$$
$$= (x_3 + \bar{x}_1)(x_3 + \bar{x}_2)(\bar{x}_3 + x_1 + x_2)$$

So OR gate can also be reduced to CNF in polytime.

Hence each gate in circuit can be reduced to a CNF formula ϕ in polytime.

∴ Circuit-SAT $\leq_p$ SAT

Proving SAT $\leq_p$ 3CNF

case 1: When clause contain one literal

$S = x$

Let $S' = (x + x_1)(x + \bar{x}_1)$

or $S'' = (x + x_1 + x_2)(x + x_1 + \bar{x}_2)(x + \bar{x}_1 + x_2)(x + \bar{x}_1 + \bar{x}_2)$

S'' is satisfiable only if S is satisfiable.

case 2: Clause contains two literals

$S = x_1 + x_2$

Let $S' = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)$

S' is satisfiable only if S is satisfiable.

case 3: clause contains three literals

$S = x_1 + x_2 + x_3$

then we need not do anything

case 4: When clause contain more than 3 literals

$S = x_1 + x_2 + x_3 + \dots x_k$

So we need $k-3$ new variables $y_1, y_2, y_3, \dots y_{k-3}$

$S' = (x_1 + x_2 + y_1)(\bar{y}_1 + x_3 + y_2)(\bar{y}_2 + x_4 + y_3) \dots$

S' is satisfiable only when S is satisfiable

Hence any clause in SAT expression can be replaced by a conjunction of clauses which contain 3 literals each.

Hence SAT problem can be reduced to an instance of 3SAT in polytime.

proving 3SAT $\leq_p$ Clique :

Let input formula be

$\phi = (x_{11} + x_{12} + x_{13})(x_{21} + x_{22} + x_{23}) + \dots (x_{n_1} + x_{n_2} + x_{n_3})$

Steps

1. construct graph G of k clusters with 3 nodes each
2. Each cluster corresponds to a clause
3. Each node in a cluster is labelled with a literal from clause

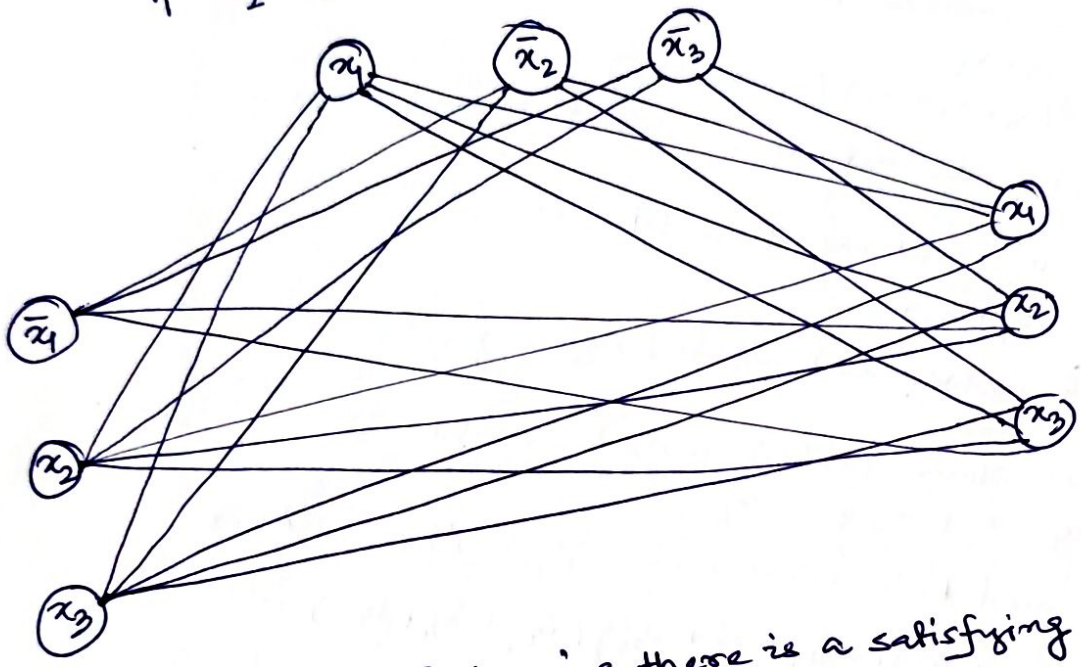
④ An edge is put between all pairs of nodes in different clusters except for pairs of the form $(x_i, \bar{x}_i)$

⑤ No edge is put between any pair of nodes belonging to same cluster.

Example:

$$\phi = (x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + x_3)(x_1 + x_2 + x_3)$$

v_1^1 v_2^1 v_3^1 v_1^2 v_2^2 v_3^2 v_1^3 v_2^3 v_3^3



Suppose ϕ has a solution i.e. there is a satisfying assignment to literals.

$\Rightarrow$ Each clause C_r contains atleast one literal l_i^r set to true.

$l_i^r \rightarrow$ denotes i^{th} literal of clause r

so $l_i^r = 1$ which corresponds to vertex v_i^r in the graph

Picking one true literal from each clause yields a set V' of k vertices.

Claim - V' is a clique

For any two vertices v_i^r and $v_j^s \in V'$ we know their corresponding literals l_i^r and l_j^s are set to 1.

Thus it is not possible that $l_i^r = \overline{l_j^s}$ i.e. They are not complement of one another, also $r \neq s$ thus edge (v_i^r, v_j^s) exists in our graph. Hence V' is clique.

conversely,

Let G contain a clique V' of size K. As no edge in G connect vertices within the same ~~height~~ triplet, so all the K vertices are from different triplets or different clauses. Hence V' contains exactly one vertex per clause.

Now if $v_i^r \in V'$ then assign 1 to the corresponding literal l_i^h. As G contains no edge between inconsistent literals. Thus no literal and its complement are set to 1

Hence each clause is satisfied $\Rightarrow \phi$ is satisfied

Hence $3SAT \leq_p$ Clique.

Thus we have established

circuit SAT $\leq_p$ SAT $\leq_p$ 3-SAT $\leq_p$ Clique

$\Rightarrow$ circuit SAT $\leq_p$ Clique.

Cook's Theorem states that all problem in NP can be reduced to circuit SAT.

As proved circuit SAT $\leq_p$ Clique

So all problems in NP can be reduced to Clique. —(i)

Earlier we have proved Clique $\in$ NP ———— (ii)

using (i) & (ii)

Clique $\in$ NPC