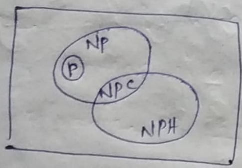
→ $P \subseteq NP$ but $P \neq NP$

→ vandiagram



→ If the problem $\boxed{x} \in NP$ then the problem $\boxed{\bar{x} \in \text{CO-NP}}$

Q proof that $P \subseteq \text{CO-NP}$ (Given that $P \subseteq NP$)

→ P is closed under complementation that is if $x \in P$ then $\bar{x} \in P$.

→ let $x \in P \Rightarrow \bar{x} \in P$ ∵ (P is closed under complimentation)

$\Rightarrow x \in NP \Rightarrow \bar{x} \in NP$

$\Rightarrow x \in \text{CO-NP}$

$\Rightarrow \boxed{P \subseteq \text{CO-NP}}$ (proved.

Q ~~prove~~ that $NP \neq \text{CO-NP}$ then $P \neq NP$.

proofs

This can be proved by contrapositive.

statement:- we have to prove if $P = NP$ then $NP = \text{co-NP}$.

let $x \in NP \Rightarrow x \in P$ (∵ $P = NP$)

$\Rightarrow \bar{x} \in P$ (∵ p is under complementation)

$\Rightarrow \bar{x} \in NP$ (∵ $P = NP$)

$\Rightarrow x \in \text{CO-NP}$

$\Rightarrow \boxed{NP \subseteq \text{CONP}}$ ——— (1)

let $x \in \text{CO-NP}$

$\Rightarrow \bar{x} \in NP$

$\Rightarrow \bar{x} \in P$ (∵ $P = NP$)

$\Rightarrow x \in P$ (∵ P is closed under complementation)

$\Rightarrow x \in NP$ (∵ $P = NP$)

$\Rightarrow \boxed{\text{CO-NP} \subseteq NP}$ ——— (2)

From eqn (1) and (2) we have ~~co~~

$\boxed{NP = \text{CO-NP}}$ if $\boxed{P = NP}$

so, by the contrapositive we have

if $NP \neq \text{CO-NP}$ then $P \neq NP$ (proved.

subset sum problem

→ In the subset sum problem we ~~have~~ are written a finite set $S \subseteq N$ and a target $t \in N$. check whether there is a subset $S' \subseteq S$. whose elements sum is equal to t.

→ Example

let $S = \{1, 2, 5, 15, 14, 20, 1\cancel{8}, 7, 6\}$ and $t = 14$.

$S_1 = \{1, 2, 5, 6\}$

$S_2 = \{2, 5, 7\}$

$S_3 = \{14\}$

$S_4 = \{1, 7, 6\}$

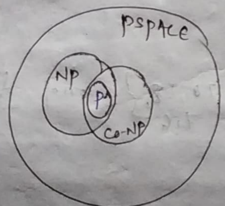
chapter-9 PSPACE (polynomial space)

→ PSPACE is a class of all decision problems that can be solved by a deterministic algorithm, using space limited by n^k for some constant k.

→ That is PSPACE is a set of all decision problems that are solvable in polynomial space.

→ $\boxed{P \subseteq PSPACE}$ that means, a polynomial time algorithm can consume only polynomial space. In the other words, a problem solved in polynomial time is also sovable in polynomial space.

→ $\boxed{NP \subseteq PSPACE}$ then $\boxed{Co\text{-}NP \subseteq PSPACE}$ so we can say that PSPACE is also closed under complimentation. It is means that if the problem belongs to PSPACE then the complement of that problem belongs to PSPACE.



$P \subseteq NP$
then $P \subseteq Co\text{-}NP$

→ There is no prove that

$P \neq PSPACE$,

$NP \neq PSPACE$.

PSPACE - Complete :-

A problem X is PSPACE - complete if

(i) it belongs to PSPACE i.e X is PSPACE.

(ii) For all problem Y in PSPACE, we have

$$Y \leq_p X.$$

Some problems under PSPACE

① Quantified Satisfiability (QSAT) problem

$O(2^n)$ it is solved in polynomial space

→ It is a decision problem.

→ Defn :- Given a CNF formula $\phi(x_1, x_2, \ldots, x_n)$, decides whether the following is true :

$$\boxed{\exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots \exists x_n \, \phi(x_1, x_2, \ldots, x_n)}$$

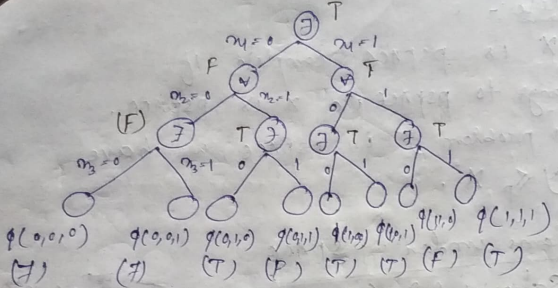
here $\exists$ is existential quantifier and $\forall$ is universal quantifier.

Does There exist a x_1 such that for all x_2 and so on there exist a x_n such that the formula ϕ is true ?

→ In this formula n is considered as odd number.

Exp $\exists x_1 \forall x_2 \exists x_3 \, \phi\left((x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3)\right)$

x_1	x_2	x_3	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\overset{a}{x_1 \vee x_2}$	$\overset{b}{\bar{x}_1 \vee \bar{x}_2 \vee x_3}$	$\overset{c}{x_1 \vee \bar{x}_2 \vee \bar{x}_3}$	$a \wedge b \wedge c$	
0	0	0	1	1	1	0	1	1	0 ×	$\phi(0,0,0)$
0	0	1	1	1	0	0	1	1	0 ×	$\phi(0,0,1)$
0	1	0	1	0	1	1	1	1	1 ✓	$\phi(0,1,0)$
0	1	1	1	0	0	1	1	0	0 ×	
1	0	0	0	1	1	1	1	1	1 ✓	
1	0	1	0	1	0	1	1	1	1 ✓	
1	1	0	0	0	1	1	0	1	0 ×	
1	1	1	0	0	0	1	1	1	1 ✓	



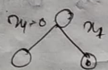
→ For ∃ then we use OR operator that means any one true then the result is true.

→ For ∀ then we use AND operator that means both are the true then the result is true.

→ If the root node is true then the formula is satisfiable.

⇒ The time complexity is $O(2^n)$ for n variables.

⇒ The space complexity :-



space = c + 1 + store the result (1) | then for by reusing the space that is $c = c + 2 = c + 1 + 1$

So for the no. of variables x_i : then the space complexity = $O(2 \cdot i + c)$

Algorithm

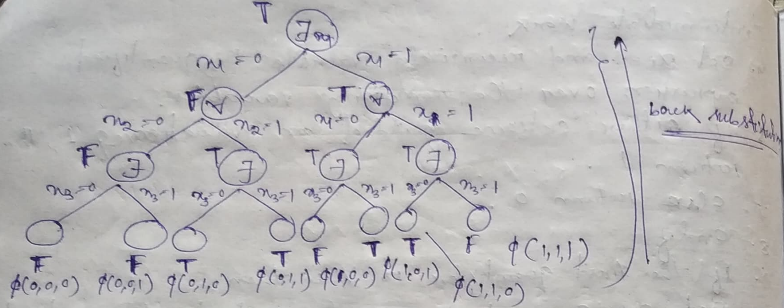
1. If the first quantifier is $\exists x_i$ then
2. set $x_i = 0$ and recursively evaluate the quantified expression over the remaining variables.
3. save the result (0 or 1) and delete all their intermediate work.
4. Set $x_i = 1$ and recursively evaluate the quantified expression over the remaining variables.
5. If either outcome yielded an evaluation of 1, then return 1.
7. Else return 0.
8. Endif
9. If the first quantifier is $\forall x_i$ then
10. Set $x_i = 0$ and recursively evaluate the quantified expression over the remaining variables.
11. Save the result (0 or 1) and delete all their intermediate work.
12. Set $x_i = 1$ and recursively evaluate the quantified expression over the remaining variables.
13. If both outcomes yielded an evaluation of 1, then return 1
14. Else return 0.
15. End if

Ex: Consider an instance of QSAT as

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

Check $\exists x_1 \forall x_2 \exists x_3 \, \phi(x_1,x_2,x_3)$ is True?

x_1	x_2	x_3	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee \bar{x}_3$	$\bar{x}_1 \vee x_2 \vee x_3$	$\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$	$\wedge$
0	0	0	1	1	1	0	1	1	1	0
0	0	1	1	1	0	1	0	1	1	0
0	1	0	1	0	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	1	1
1	0	0	0	1	1	1	1	0	1	0
1	0	1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	0	1	1	1	0	0



→ The root node is true so this formula $\phi(x_1, x_2, x_3)$ is satisfiable.

planning problem

Given a set of condition $C = \{c_1, c_2, ---, c_n\}$ a set of operators $O = \{o_1, o_2, o_3, ---, o_k\}$ initial configuration $C_0 \subseteq C$ and a goal configuration $C^* \subseteq C$. Is it possible to apply sequence of operations to get from initial configuration to goal configuration?

Example

1. 8-puzzle problem.
2. 15-puzzle problem.
3. Rubik's cube.

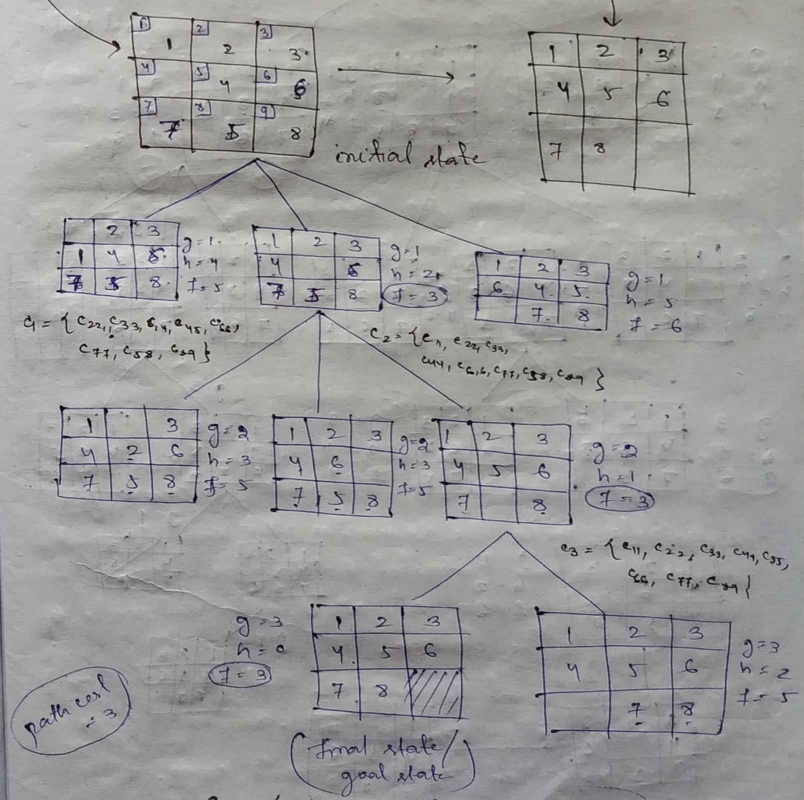
Example . 8-puzzle problem

Conditions : c_{ij} , $1 \le i , j \le 9$

Here c_{ij} means tile 'i' is in square 'j'.

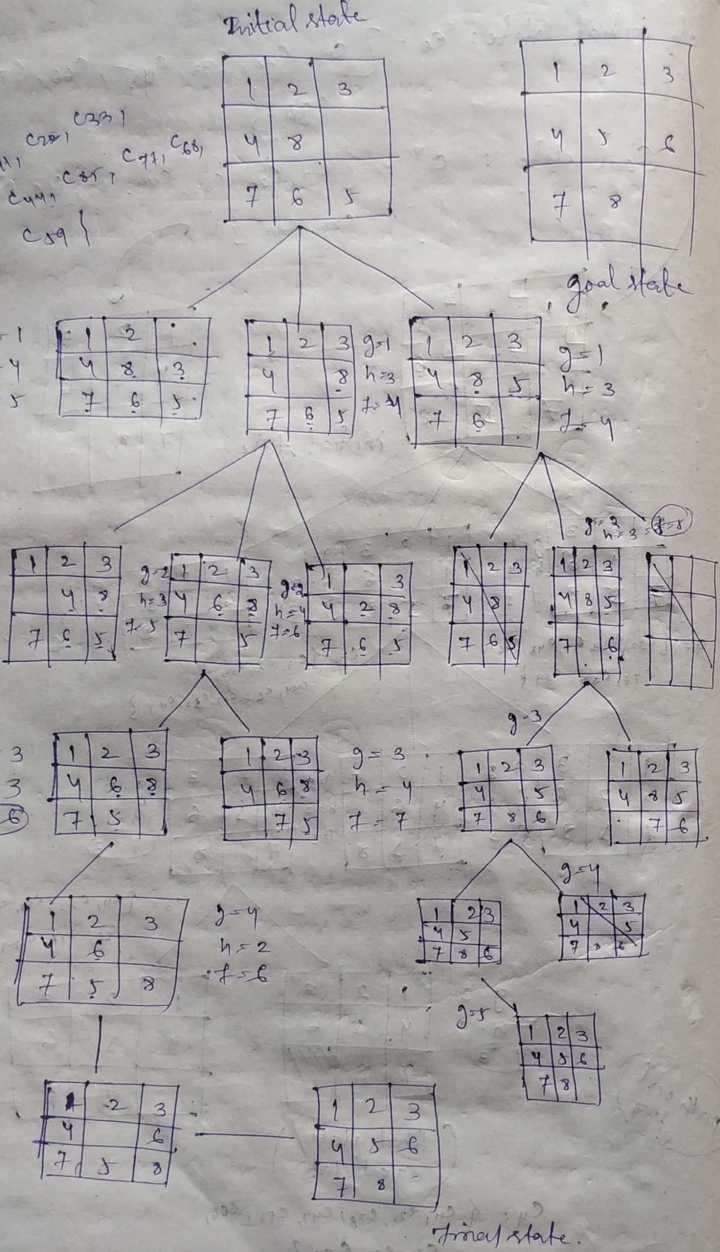
initial state : $C_0 = \{c_{11}, c_{22}, c_{33}, c_{45}, c_{66}, c_{77}, c_{58}, c_{89}\}$

Goal state : $C^* = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{55}, c_{66}, c_{77}, c_{88}, c_{99}\}$.



Q

$C_0 = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{85}, c_{77}, c_{68}, c_{59}\}$



$c = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{85}, \bullet, c_{77}, c_{68}, c_{59}\}$

$c_1 = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{85}, c_{56}, c_{77}, c_{68}, \ , \}$

$c_2 = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{85}, c_{56}, c_{77}, \ , c_{69}\}$

$c_3 = \{c_{11}, c_{22}, c_{33}, c_{44}, \ ; c_{56}, c_{77}, c_{88}, c_{89}\}$

$c_4 = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{55}, \ , c_{77}, c_{88}, c_{69}\}$

$c_5 = \{c_{11}, c_{22}, c_{33}, c_{44}, c_{55}, c_{66}, c_{77}, c_{88}, \ ,\} \rightarrow$ goal state.

Competitive Facility location problem.

→ competitive facilating location problem is PSPACE-complet problem.

→ competitive facility location can solved in polynomial space.

→ competitive facility location works same as that of QSAT problem.

→ only main differents is that a player may have more than just 2 options in each move, because we have a graph that has 'n' nodes that a player can choose from. so a player has 'n' choices in each step rather from just two. so the recursion tree is an 'n-array' tree rather than a binary tree.

~~→ Reduces~~

→ QSAT reduces to Competitive Facility location problem

→ Given an instance of QSAT we construct an instance of Competitive Facility location such that the second ~~pair~~ player can force a win iff the quantified Boolean formula is false.

Construction step

Given a Boolean CNF formula, we construct a graph in the following way:

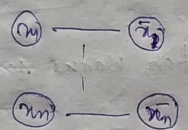
(1) For each variable in the boolean formula, ~~evaluate~~ create two nodes in the graph.

(2) Connect these two nodes by an edge.

→ we have one node corresponding to the first variables x_1, and then the 2nd node which will corresponding to the negation / complement of that variables.

→ we have the same for variable x_2 and $\bar{x}_2$ and again with connect the 2 by an edge.

→ we will continue as same upto the variable $\bar{x}_n$ and x_n.



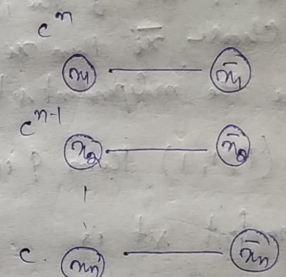
→ The players will be able to choose these vertices and that they will not be able to choose for a variable, both the variables itself and it's negation because they are connected by an edge. so competitive facility location does not allow both end points ~~to~~ of that edge to be choosen.

→ In CFL we want to enforce that the 1st player first picks either x_1 or $\bar{x}_1$ and 2nd player then picks either x_2 or $\bar{x}_2$ and so on.

→ This can be done by simplely assigning a very large value to that nodes associated to the variables x_1 and $\bar{x}_1$. and slightly smaller value to the variables x_2 or $\bar{x}_2$ and so on.

→ The smallest values to the nodes corresponding to the variable x_n and $\bar{x}_n$.

→ specifically we choose a large constant c and we assign a value c^n to the variables x_i and $\bar{x}_1$ and then we assign a value c^{n-1} to the x_2 and $\bar{x}_2$ and so on. and lastly we assign c^1 to the x_n and $\bar{x}_n$.



the profit of each node

$$\boxed{b_{x_i} = c^{1+n-i}}$$

$$b_{x_1} = c^{1+n-1} = c^n$$

$$b_{x_2} = c^{1+n-2} = c^{n-1}$$

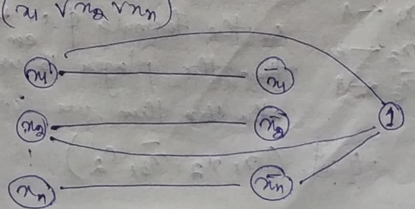
→ For competitive facility location we also have to specify what our threshold target profit. so the first player wants to prevent the second player for getting atleast 'k' units of profit.

→ This 'k' is choosen as

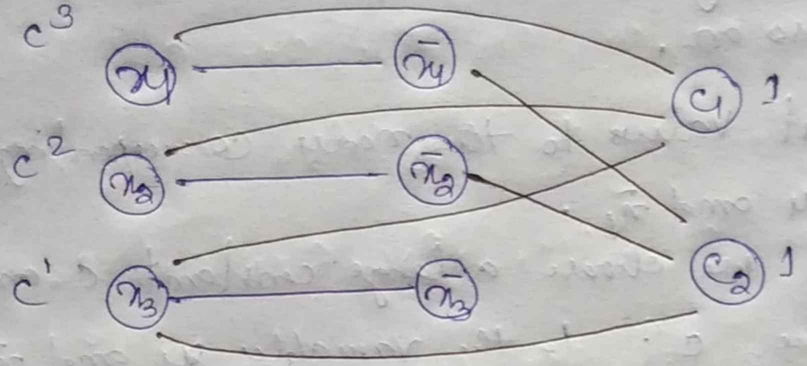
$$k = c^{n-1} + c^{n-3} + c^{n-5} + \dots + c^2 + 1$$

Exp construct the CFL

CNF $= (x_1 \vee x_2 \vee \bar{x}_n)$



Exp $\phi = (\underbrace{x_1 \vee x_2 \vee x_3}_{c_1}) \wedge (\underbrace{\bar{x}_1 \vee \bar{x}_2 \vee x_3}_{c_2})$



Let $\left.\begin{array}{l} x_1 \leftarrow A \to 1 \\ x_2 \leftarrow B \to 1 \\ x_3 \leftarrow A \to 1 \end{array}\right)$ if a player chooses x_i then $x_i = 1$ and if the player choose $\bar{x}_i$ then $x_i = 0$

(i) $\{x_1, x_2, x_3\}$ are the independent set and $(x_1, x_2, x_3) = (1, 1, 1)$ then ϕ is true.

(ii) then another independent set is

$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{array}{l} \bar{x}_1 \leftarrow A \\ \bar{x}_2 \leftarrow B \\ x_3 \leftarrow A \end{array}$ then input is $(0, 0, 1)$

so the formula ϕ is satisfiability, then the player 1 or A wins the game otherwise the second player B wins.

(iii) then another independent set is

$\begin{array}{ll} 1 & x_1 \leftarrow A \\ 1 & x_2 \leftarrow B \\ 0 & \bar{x}_3 \leftarrow A \end{array}$ so the formula ϕ is ~~sati~~ not satisfy can chance to then the player B wins the game ~~othe~~ ~~then~~ then we add some prefix

that $(x_1, x_2, \bar{x}_3, c_2)$