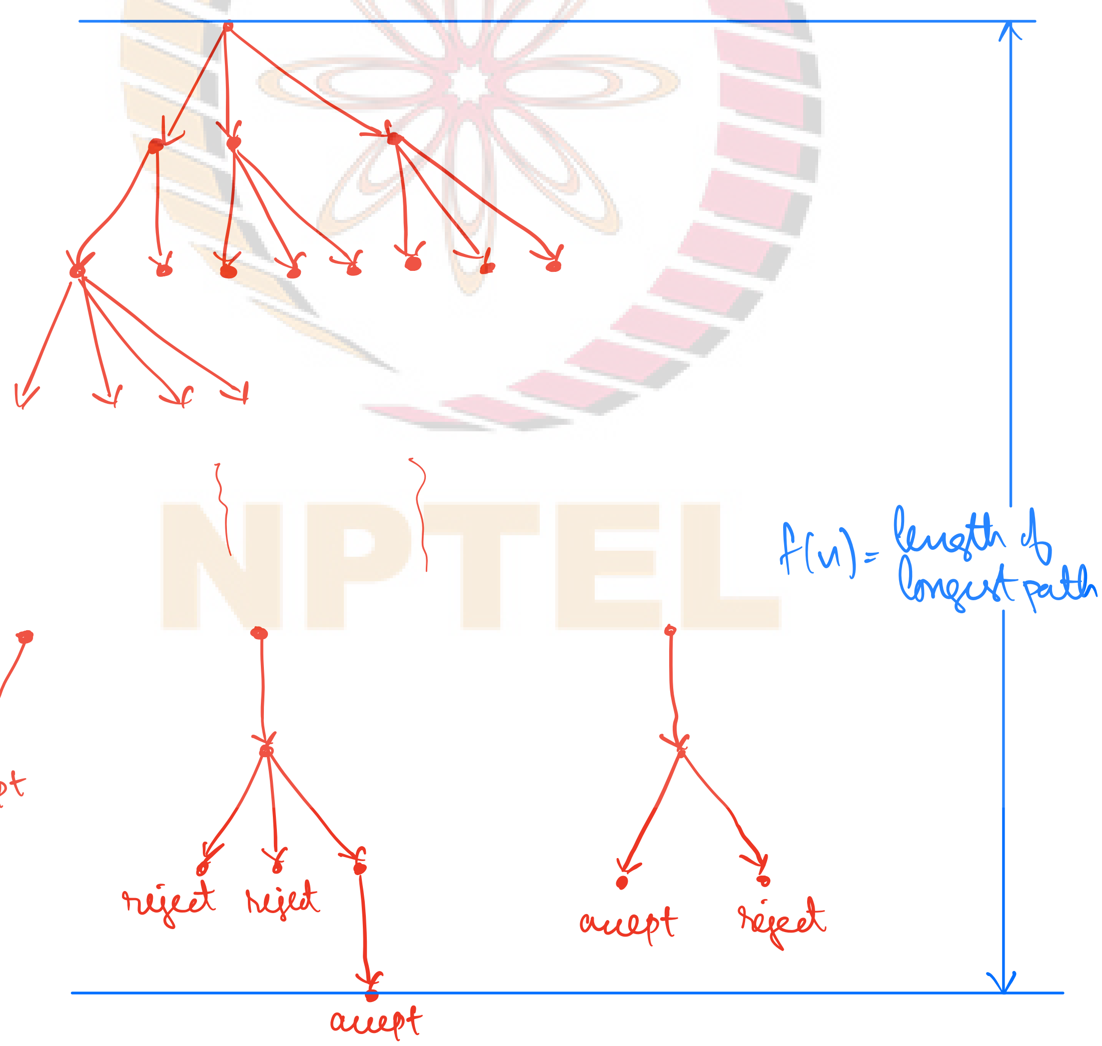
Nondeterministic Polynomial Time

Def 7.9: Let N be a non-deterministic TM that is a decider. The running time of N is $f: \mathbb{N} \to \mathbb{N}$ where $f(n)$ is the maximum no. of steps that N uses on any branch of its computation on an input of length n.



Def 7.21:

$\text{NTIME}(t(n)) = \{ L \mid L \text{ is decided by an NTM in } O(t(n)) \text{ time} \}$

Def 7.22: $NP = \bigcup_{k=1}^{\infty} \text{NTIME}(n^k)$

We will see that this is equivalent to the Guess & Verify model.

Examples:

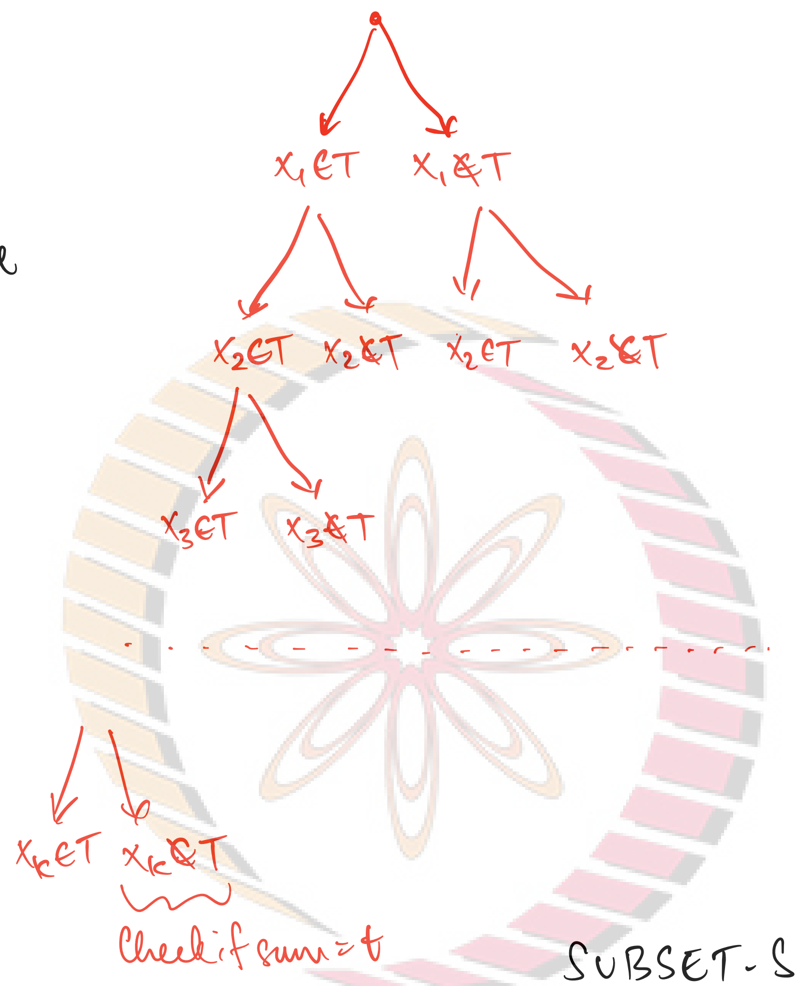
1) $\text{SUBSET-SUM} = \{ \langle S, t \rangle \mid S = \{x_1, x_2, \dots x_k\},\ \exists T \subseteq \{1, 2, \dots k\},\ \text{s.t} \sum_{i \in T} x_i = t \}$

On input $\langle S, t \rangle$:

1. Non deterministically select/reject each of $x_1, x_2, \dots x_k$. $\rightarrow O(k)$ time
2. Add all the selected x_i and verify if they add up to t. $\rightarrow O(k)$

3. If sum $= t$, then accept. Else, reject.

2^k possible branches

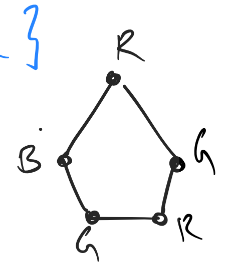


SUBSET-SUM $\in$ NP

---

2) $\underline{\text{3-COLORABLE}} = \{ \langle G \rangle \mid G \text{ is 3-colorable} \}$

→ If G has n vertices, brute force (deterministic) takes $3^n * m$ time



On input $\langle G \rangle$:

1. Go through the vertices $1, 2, \dots n$ and non-deterministically assign colors R, G, B. $\}$ $O(n)$

2. Go through each edge of G and check if $\}$ $O(m)$

it is properly colored.

3. Accept if coloring is valid. Else reject.

3-COLORABLE $\in$ NP.

Note that a DTM can be considered as an NTM.

$$\mathrm{DTIME}(t(n)) \subseteq \mathrm{NTIME}(t(n))$$
$$\mathrm{DTIME}(n^k) \subseteq \mathrm{NTIME}(n^k)$$

So $\quad P \subseteq NP$

P vs. NP question: Is the above containment proper? Is $P = NP$ or $P \subsetneq NP$?

This is a big open question. One of the biggest open questions as of today.

Formally asked by Stephen Cook (1971) and independently by Leonid Levin (1973)

– Also asked by others : 1956 letter by Kurt Gödel to John von Neumann.

SAT : Short for Satisfiability. There is a Boolean formula and we want to know if it is satisfiable

Variables x_i , Negated variables $\overline{x}_i$

Literals

Literals are connected using AND $\wedge$
and OR $\vee$

$x_1 = T$
$x_2 = T$
$x_3 = T$

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

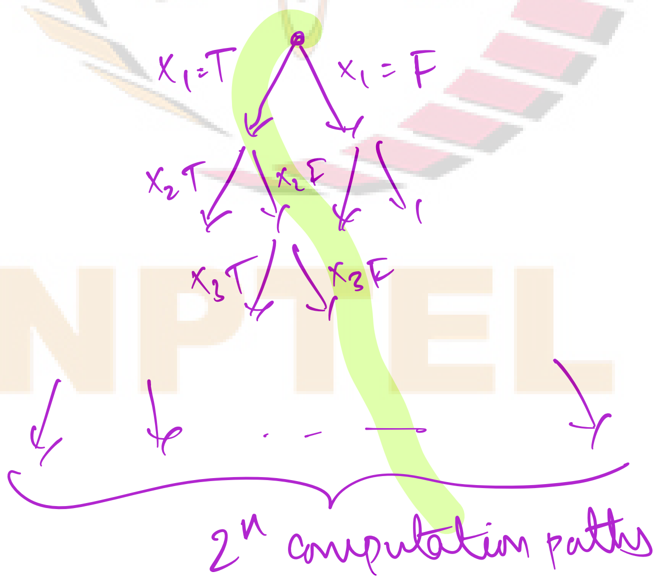
NOT SATISFIABLE.

SAT = $\{ \langle \phi \rangle \mid \phi$ is a Boolean formula that is satisfiable $\}$

There is an assignment of True | False to the Boolean variables such that the formula evaluates to True.

It can be seen that SAT $\in$ NP.

- Nondeterministically assign True | False to each x_i.
- Verify if the assignment evaluates to True. Accept if it does.



CNF formula: Conjunctive Normal Form. $\in$ CNF-SAT

$$\Psi = \underbrace{(x_1 \vee \overline{x_2} \vee x_3 \vee x_4)}_{C_1} \wedge \underbrace{(x_2 \vee \overline{x_3} \vee x_5)}_{C_2} \wedge \underbrace{(\overline{x_4} \vee \overline{x_6})}_{C_3}$$

$C_1 \wedge C_2 \quad \cdots \wedge C_3$

CNF: AND of OR's. AND of clauses where each clause is an OR of literals.

CNF-SAT $= \{ \langle \phi \rangle \mid \phi$ is a CNF formula that is satisfiable. $\}$

Above formula $\Psi \in$ CNF-SAT

3-CNF SAT or 3-SAT $= \{ \langle \phi \rangle \mid \phi$ is a CNF formula where each clause has exactly 3 literals, ϕ is satisfiable $\}$

SAT, CNF-SAT, 3-SAT are all in NP.

In all the above problems, the NP approach was to "guess" a solution and then "verified" it. Are all the NP algorithms of the "guess & verify" type? We will see in the next lecture.