TIME COMPLEXITY

Let $t : \mathcal{N} \to \mathcal{N}$ be a function. Time complexity class
$TIME(t(n)) = \{ L \mid L$ is a language decided by an $O(t(n))$ TM$\}$

$A = \{ 0^k 1^k \mid k \geq 0 \}$

We know TM such that $A \in TIME(n^2)$

M_2:

Repeat as long as some 0^s & some 1^s remain on tape

- Scan the tape. If total # 0^s & 1^s remaining is odd, REJECT
- Scan the tape, crossing off every other 0 starting with the first, & then crossing off every other 1 starting with the first.

If no 0^s & no 1^s remain on tape, ACCEPT else REJECT.

With M_2, $A \in TIME(n \log n)$.
With a 2-tape TM M_3, $A \in TIME(n)$.

Computability : all reasonable models of computation are $\leftrightarrow$
Complexity : choice of model affects time complexity

<u>Th</u>: Each $t(n)$ time multitape TM has an $\leftrightarrow$ $O(t^2(n))$ time single tape TM.

<u>Th</u>: Each $t(n)$ time non-det. ~~TM~~ single tape TM has an $\leftrightarrow$ $2^{O(t(n))}$ time deterministic single tape TM.

Def: P is the class of languages that are decidable in poly. time on a deterministic 1-tape TM.

$$P = \bigcup_k \mathrm{TIME}(n^k)$$

1) P is invariant for all models of computation that are polynomially $\Leftrightarrow$ det. 1-tape TM.
2) $\longleftrightarrow$ class of problems realistically solvable on a computer.

PATH $= \{\langle G, s, t\rangle \mid$ directed path in G from s to t$\}$

PATH $\in$ P

$M(\langle G, s, t\rangle)$:

1) Mark s
2) Repeat until no additional nodes are marked
3) Scan edges of G. If edge $(\mathring{a}, b)$, mark b
4) If t is marked, then ACCEPT, else REJECT.

This M uses a breadth-first search. Running time ?

RELPRIME $= \{\langle x, y\rangle \mid x$ and y are relatively prime$\} \in$ P

Euclid's GCD $\langle x, y\rangle$:

1) Repeat until $y = 0$
2) $x \leftarrow x \bmod y$
3) Exchange x & y
4) Output x.

R for RELPRIME $\langle x, y\rangle$:

1) Run Euclid's GCD $\langle x, y\rangle$
2) If result = 1, ACCEPT else REJECT

Running time : $\min(\log_2 x, \log_2 y) \times$?

THE CLASS NP

eg.
Hamiltonian path ≡ directed path that goes thru' each node <u>once</u>.

HAMPATH ≡ $\{\langle G, s, t\rangle \mid$ Hamiltonian path from s to t in G$\}$

- polynomial time soln unknown
- has <u>polynomial verifiability</u>

$\overline{\text{HAMPATH}}$ not polynomially verifiable

eg COMPOSITES $= \{x \mid x = pq,$ for integers $p, q > 1\}$

<u>Defn</u>: A verifier for a language A is an algorithm V, where

$A = \{w \mid V$ accepts $\langle w, c\rangle$ for some (certificate) string $c\}$

A poly-time verifier runs in poly-time in length of w

A is poly-verifiable if it has a poly time verifier.

Verifier uses a certificate (proof) c to verify $w \in A$.

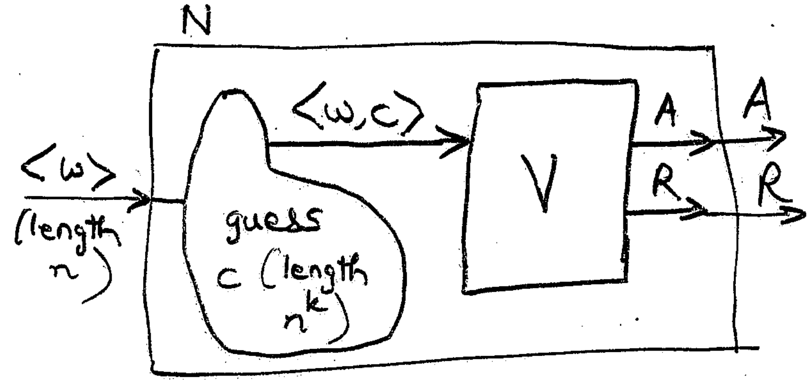
eg For $w = \langle G, s, t\rangle \in$ HAMPATH, c = the Hamiltonian path

For COMPOSITES, c = a divisor of x

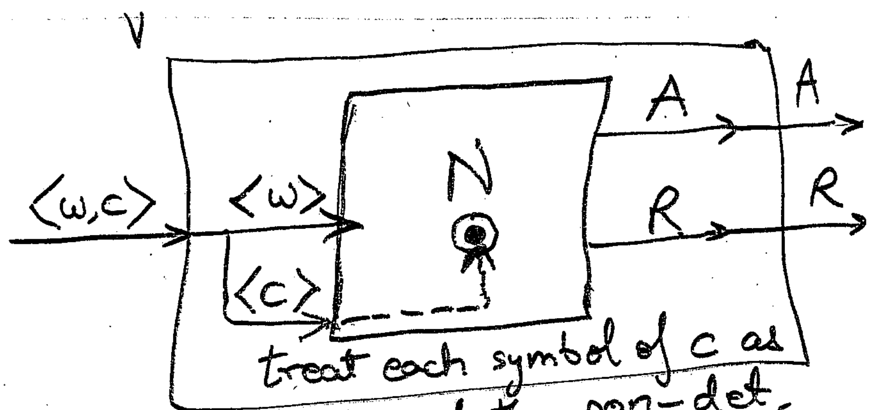
<u>Defn</u>: NP ≡ class of languages that have poly-time verifiers.
(Non-deterministic polynomial)

<u>Theorem</u>: A language is in NP iff it is decided by a non-deterministic poly-time TM.

$\Rightarrow$ NTM simulates verifier by guessing certificate



$\Leftarrow$ Verifier simulates the NTM by using the accepting branch as the certificate



treat each symbol of c as a description of the non-det. choice at each step

<u>Defn</u>: $NTIME(t(n)) = \{L \mid L$ decided by a $O(t(n))$ time NDTM$\}$.

<u>Cor</u>: $NP = \bigcup_k NTIME(n^k)$

<u>Defn</u>: Clique in an undirected graph $\equiv$ subgraph where each pair of nodes has an edge connecting them

Th: CLIQUE $\in$ NP // $V(\langle G,k\rangle, c)$

SUBSET–SUM $= \{\langle S,t\rangle \mid S = \{x_1, x_2, \cdots x_k\}$ and for some $\{y_1, y_2, \cdots y_l\} \subseteq S$, we have $\sum y_i = t\}$

// multi-sets allowed

Th: SUBSET–SUM $\in$ NP // $V(\langle S,t\rangle, c)$

$\overline{\text{CLIQUE}}$, $\overline{\text{SUBSET–SUM}}$: not obviously members of NP

NP = co-NP ?

P: membership can be decided polynomially
NP: " " " verified "

$P \subseteq NP$; $P = NP$?

$NP \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$

SAT $= \{\langle \phi\rangle \mid \phi$ is a satisfiable Boolean formula$\}$
3SAT $= \{\langle \phi\rangle \mid$ " " " " 3cnf "$\}$

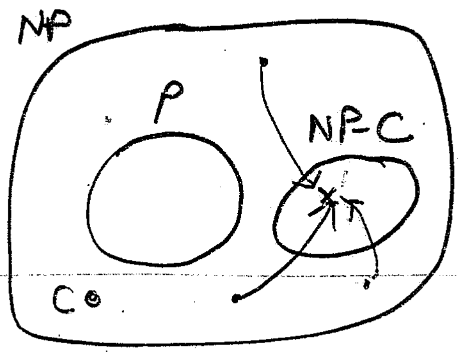
Defn: Language A is poly time reducible to lang B, $A \leq_p B$, if a poly time computable func $f: \Sigma^* \rightarrow \Sigma^*$ exists, where for each w, $w \in A \iff f(w) \in B$

Th: If $A \leq_p B$ and $B \in P$, then $A \in P$.

NP- Completeness

<u>Defn</u>: Lang B is NP-Complete if 1) $B \in NP$ and
2) $\forall A \in NP$, $A \leq_p B$

<u>Th</u>: If B is NP-complete & $B \in P$, Then $P = NP$



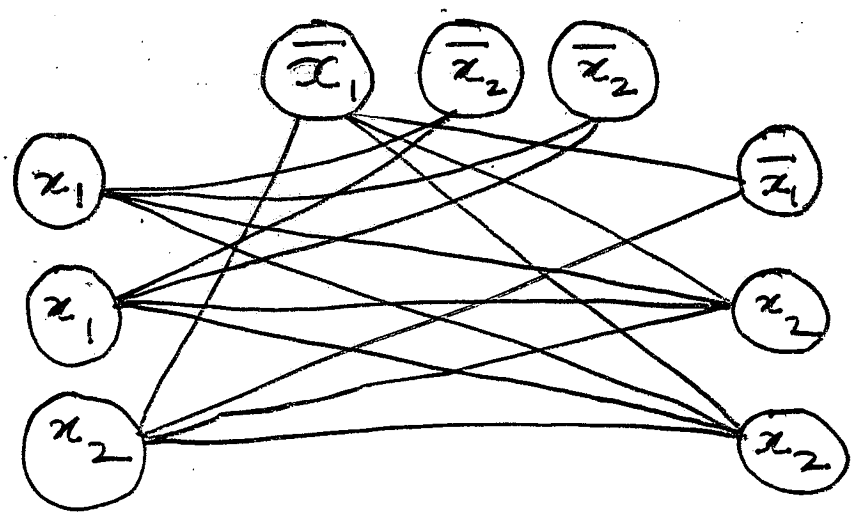
SAT, 3-SAT $\in$ NP-Complete

<u>Th</u>: If $B \in$ NP-Complete and $B \leq_p C$ for C in NP,
then $C \in$ NP-Complete

<u>E.g.</u> CLIQUE $\in$ NP- Complete

Clique $\in$ NP. Show 3-SAT $\leq_p$ CLIQUE
(k clauses) (k-)

<u>eg</u>: $\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2 \vee x_2)$



Show ϕ is satisfiable $\Rightarrow \Leftarrow$ G has a k-clique