$\{\text{all such languages}\} = \mathrm{TIME}(f(n))$

e.g. $\mathrm{TIME}(n^3) = \{$ languages decidable in time $\leq Cn^3$ for some $C \}$

$$\mathrm{TIME}(n^3) = \mathrm{TIME}(20n^3 + 1000n^2)$$

$$P = \bigcup_{c=1,2,\ldots} \mathrm{TIME}(n^c)$$

= languages decidable in "polynomial" time

Ch 7: P vs NP

Ch 8, 9, ...: " " "

Last week or two: regular languages, context-free "

Remark: P = Polynomial Time

NP = Non-deterministic poly time

PSPACE, N(P)SPACE $\leftrightarrow$ P vs NP

Review:

We say that a language (over Σ), L, is decidable in time $f(n)$ if there is a (multi-tape) Turing machine, M, and a constant C such that on input w, M decides if $w \in L$ or not in $\leq C\, f(n)$ steps, $n = |w|$.

Non-determinism:

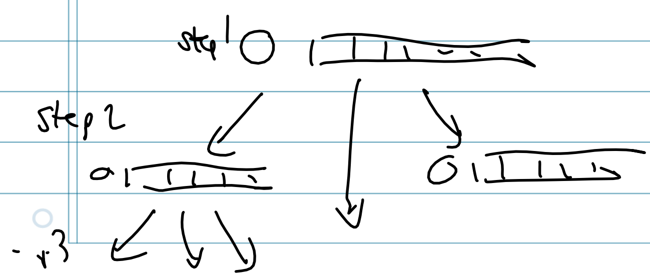
rather than

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

$\delta(q, \gamma) \mapsto$ just one possible new configuration

Non-det:

$$\delta : Q \times \Gamma \to \mathrm{Power}(Q \times \Gamma \times \{L, R\})$$

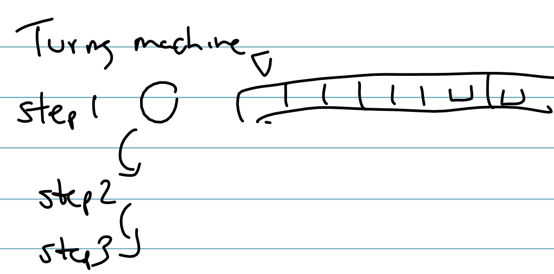


2COLOR, PRIMALITY (recent result), GRAPH-CONNECTED, ... $\in P$

3COLOR, PARTITION, SUBSET-SUM, SAT, 3SAT $\in P$???

NP:

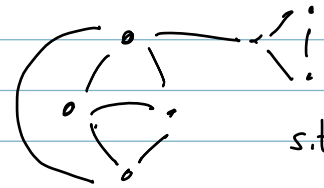
(1) Classical: Non-deterministic TM:



3COLOR:

$$3COLOR = \left\{ \langle G \rangle \;\middle|\; \begin{array}{l} G \text{ is a graph} \\ \text{that has a} \\ 3 \text{ colouring} \end{array} \right\}$$

i.e.



$G = (V, E)$ s.t. the vertices, V, can be assigned "colours" {red, blue, green} s.t. each edge's endpoints are coloured differently

=

Not at all clear how to solve this in polynomial time. But, if we could check all colourings, $3^{|V|}$, we could check if a colouring works in polynomial time.

=

So 3COLOR has a natural algorithm that guesses a colouring ($3^{|V|}$ colourings) and checks.

=

So a language, L, is poly-time-guessable or poly-time verifiable if there is a language L' decidable in poly time s.t.

$$w \in L \iff \left\{ \langle w, v \rangle \in L' \text{ s.t. } \dots \right\}$$

Example: 3COLOR language over Σ s.t. $1,2,3 \in \Sigma$. Then each $v \in \{1,2,3\}^*$ s.t. $|v|$ = # vertices of a graph, G, with vertices $\{1,2,\dots,n\}$

$\iff \{1,2,\dots,n\} \to \{1,2,3\}$

$\iff$ colouring of vertices with $\{1,2,3\}$

=

$w = \langle G \rangle$ is 3-colourable

$\iff \exists\, v = v_1 \dots v_n,\ v_i \in \{1,2,3\}$

Fix a constant c,

$$\left\{ \langle w, v \rangle \;\middle|\; \begin{array}{l} v \in \Sigma^* \\ |w| = cn^c \end{array} \right\}.$$

=

Formally:

$L \in$ poly-time verifiable if L is over Σ, there is c constant, positive integer, language $L' \in P$ s.t. $\forall w \in \Sigma^*$

$$w \in L \iff \exists\, v \in \Sigma^{cn^c}, \text{ s.t. } \langle w, v \rangle \in L'$$

=

x_1 = true or false

x_2 = .. - -

x_n = - - -

we have $f(x_1, \ldots, x_n) = $ true

e.g. $f(x_1) = x_1 \wedge (\neg x_1)$

is not satisficble

$f(x_1, x_2) = x_1 \wedge (\neg x_2)$

then $f(\text{true}, \text{false}) = \text{true}$

so $x_1 \wedge (\neg x_2)$ is satisfiable

Intuitively: given $f(x_1, \ldots x_n)$ and $x_1 = \begin{Bmatrix} T \text{ or } \\ F \end{Bmatrix}, x_2 = \begin{Bmatrix} T \\ F \end{Bmatrix}, x_3 = \begin{Bmatrix} T \\ F \end{Bmatrix}$. -

s.t.

V gives a colouring of the vertices of G...

$$SAT = \left\{ \langle f \rangle \;\middle|\; \begin{array}{l} f \text{ is a Boolean formula} \\ f = f(x_1, \ldots, x_n) \\ \text{s.t. } f \text{ has a satisfying assignment} \end{array} \right\}$$

x_1 and (not x_1) $= f(x_1)$

$f(\text{true}) = \text{false}$

$f(\text{false}) = \text{false}$

$f(x_1, \ldots, x_n)$ has a "satisfying assignment" if for some

Over $\Sigma = \{1, 2, \ldots, 9, 0, X, \#, \ldots\}$

Language

$$L' = \left\{ \langle f, v \rangle \;\middle|\; \begin{array}{l} f \text{ is a Boolean formula,} \\ v \text{ is an assignment of } f\text{'s variables} \\ \text{s.t. } f \text{ is true with that assignment} \end{array} \right\}$$

is in polynomial time.

SAT we are sure if SAT $\in$ P

but $SAT = \{ \langle f \rangle \mid \exists\, v$ as above such that $\langle f, v \rangle \in L' \}$

easy to see if f at these values is true or not:

given:

$x_1 \wedge (\neg x_2) \;\#\; x_1$ is T

$\# \; x_2$ is T

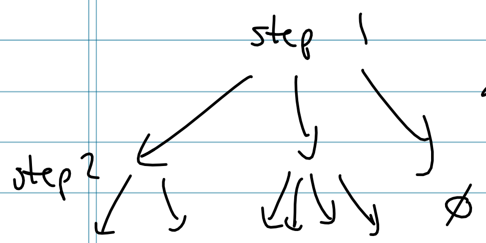
String over

$\{1, 2, \ldots, 9, 0, X, \#, \text{"is"}, T, F, \wedge, \vee, \neg, (,)\}$

X, 1, $\wedge$, (, $\neg$, X, 2,), #, X, 1, is, T, #, X, 2, is, T

Thm: NP is exactly the set of languages that can be decided in non-deterministic poly-time:

$\delta : Q \times \Gamma \to$ any number of possible elements of $Q \times \Gamma \times \{L, R\}$



a few possible transitions

$\delta : Q \times \Gamma \to \text{Power}(Q \times \Gamma \times \{L, R\})$

L is decidable in non-deterministic time g(n) if

Textbook:

$NP = \{\text{languages that can be poly-time verified}\}$

SAT ∈ NP, 3COLOR ∈ NP,

$\left.\begin{array}{l}\text{any problem with}\\ \text{some unknown}\\ \text{information, v,}\\ \text{that you are}\\ \text{trying to find}\\ \text{s.t. if you find}\\ \text{one v then}\\ \text{"success"}\\ \text{otherwise fail}\end{array}\right\}$ in NP

there is a non-deterministic T.M. such that

(1) on input w, all computation paths of M on input w halt within time $\leq C\, g(n)$ (C is independent of n)

(2) $w \in L \iff$ ∃ at least one computation path that ends in q_{acc}