Nov 8 ①

- Reductions
- SAT, 3SAT, SUBSET-SUM are NP-complete

Cook-Levin Theorem:

Given L recognized by a NTM in time n^k, there is a poly-time Turing machine computes

$$f : \Sigma^* \longrightarrow (\Sigma')^*$$

Σ alphabet of L $\qquad$ Σ' alphabet of SAT

String

s.t. $\forall w \in \Sigma^*$

$$w \in L \iff f(w) \in SAT.$$

(Real Thm: If $SAT \in P$ then $P = NP$)
($\Leftarrow$)

Really: Weaker notion of using SAT:

Best (weakest):

Given a Turing machine that can call SAT as a oracle, if $SAT \in P$ then

$L \in \text{Poly-Time}^{SAT}$ or P^{SAT} = poly time with oracle SAT

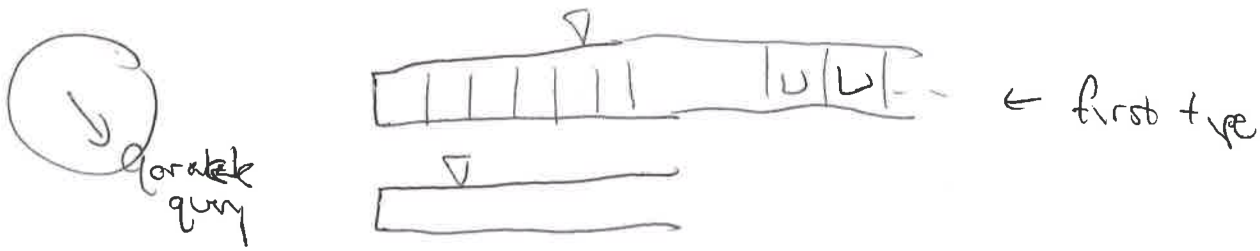
If S is a language (think of SAT),

we write

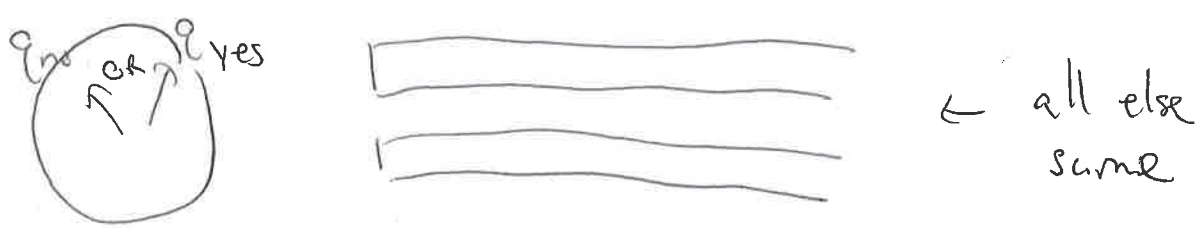
$$L \in P^S = \text{Polytime}^S$$

if there is an oracle Tm., M^S that recognizes L is poly-time with oracle S

M is usual Tm. but $Q = \{ \ldots q_{\text{oracle query}}, q_{\text{oracle yes}}, q_{\text{oracle no}} \ldots$



next step:



q_{yes} if word on 1-tape is in S

q_{no} " " " — — not in S

And you can apply this oracle call — enter $q_{\text{oracle query}}$ as often as you like ...

=

Reduction: L is poly-time reduced to S if $\exists$ poly-time $f: \Sigma^* \to \Sigma'^*$ s.t. $\forall w \in \Sigma^*$, $w \in L \Leftrightarrow f(w) \in S$.

Reduction: $L \leq_{\substack{\text{poly}\\ \text{time}}} S$ stronger notion.

Cook-Levin: $L \in NP$ then $L \leq_p SAT$.

Definition: We say S is NP-complete if (we can prove that)

① $S \in NP$, ② $\forall L \in NP$, $L \leq_{\substack{\text{poly}\\ \text{time}}} S$

E.g. $L \in NP$ $\underset{\text{time}}{\overset{\text{procedure poly}}{\rightsquigarrow}}$ encode membership in L into a SAT problem

SUBSET-SUM:

Given 2, 3, 17, 21, 47, 11, 16 target 51

Given $n_1, \ldots, n_k$ target t

is there $I \subset \{1, \ldots, k\}$ s.t. $\sum_{i \in I} n_i = t$

Looks like an arithmetic question

Claim SUBSET-SUM is NP-complete.

Given $f(x_1, \ldots, x_n) = (x_1 \text{ or } \neg x_2) \text{ and } (x_3 \text{ or } x_4) \ldots$ 4

$\rightsquigarrow$ subset-sum problem

3CNF

Say $((x_1 \text{ or } x_2 \text{ or } (\neg x_3)))$ and $((\neg x_1) \text{ or } x_4 \text{ or } (\neg x_2))$ and ...

Is this true

Make subset-set sum problem:

1st clause	etc.		2nd	1st digit	~~number we use~~ n_1
1		0	0	1	$n_1 \leftrightarrow x_1 = T$
0		0	0	1	$n_2 \leftrightarrow x_1 = F$
1		0	1	0	$n_3 \leftrightarrow x_2 = T$
0		0	1	0	$n_4 \leftrightarrow x_2 = F$
0		1	0	⋮	$x_3 = T$
1		1	0	⋮	$x_3 = F$
0		0	⋮		
0		.			
⋮		1			
0				0	← fiddle
		1	1	1	← target

for some subset to sum to target:

choose n_1 or n_2 but not both

choose n_3 or n_4

at least 1 maybe 2, ... 3

How can we deal with maybe 1, maybe 2, maybe 3