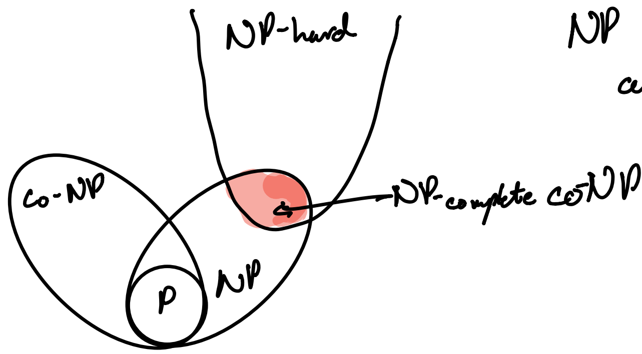
Lecture 25 Scribble

Chat Moderators: Eliot & Tanvi

Topics: Cook-Levin Theorem

3SAT to Subset-Sum

3SAT to CSAT



coNP

NP if it has a poly-time certifier for all YES instances

" for all No instances

tautology if a statement is always true

"y = x or y != x"

A problem (X) is NP-hard if for every problem (Y) ∈ NP Y is reducible to X.

NP-complete = NP ∧ NP-hard

So far:

Cook-Levin Theorem: SAT is NP-complete

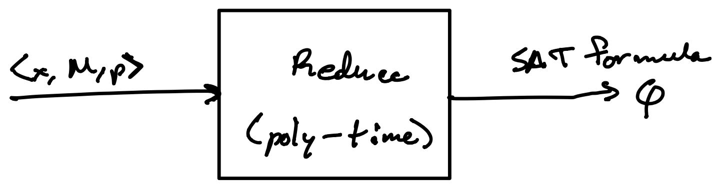
For SAT to be NP-complete

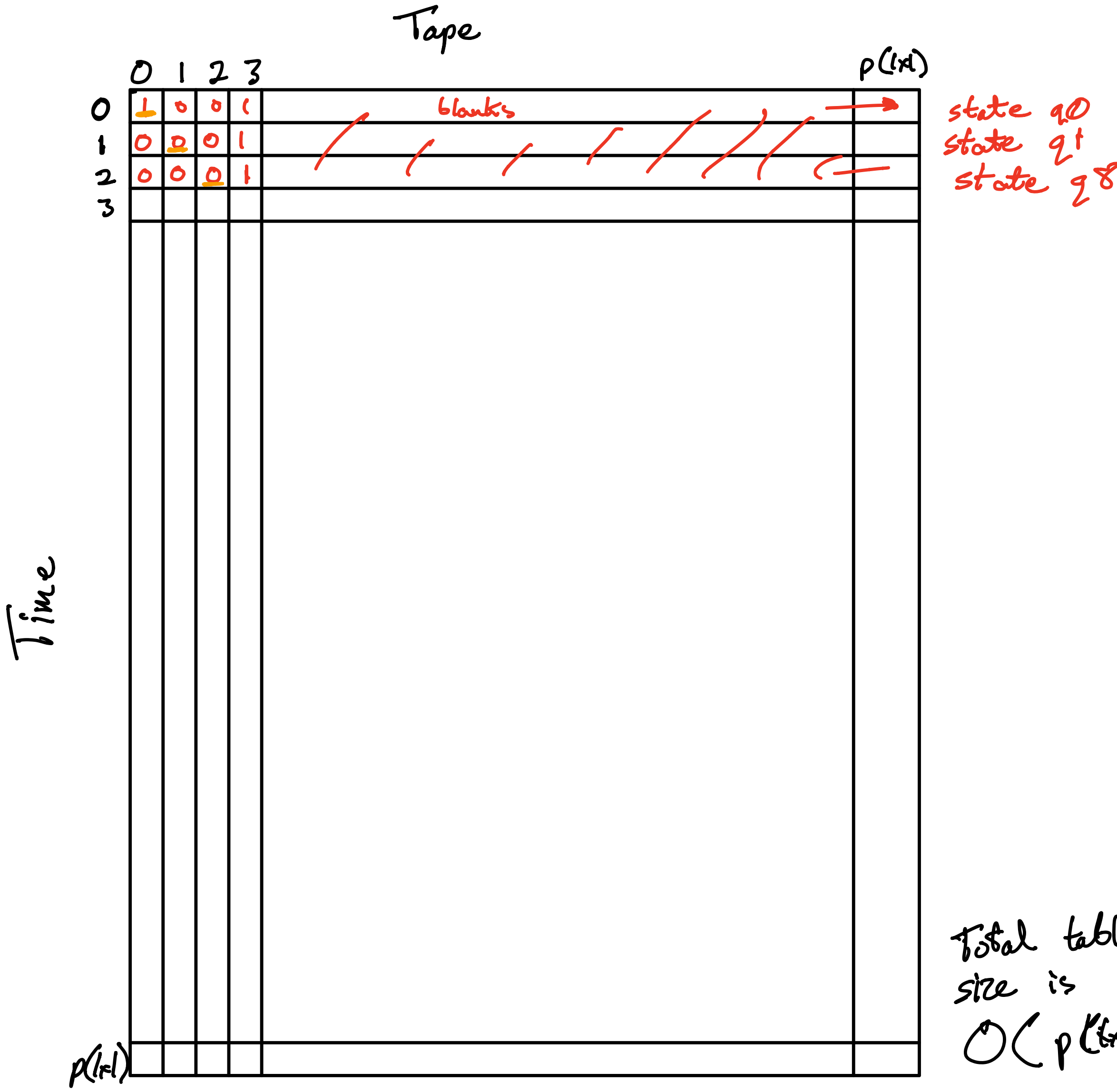
- Must be in NP ← Easy: poly-time certifier
- Must be NP-hard ←
  - ↳ Every NP problem reducible (in poly-time) to SAT

What does it mean for $L \in NP$?

- $L \in NP$ it means that there is a nondeterministic TM (M) that will halt on a input (x) in a polynomial # of steps ($p(|x|)$)

$$L = \{x \in \Sigma^* \mid M \text{ accepts } x \text{ in at most } p(|x|) \text{ steps}\}$$





Total table size is $O(p(|x|)^2)$

Four types of computation that describe M on x

- $T(b, h, i)$ tape cell at position "h" holding character "b" at time "i"
- $H(h, i)$: head at position h at time i
- $S(q, i)$: state "q" of M at i
- $I(j, i)$:instruction j is executed at time i

Notation:

$\oplus(x_1, x_2, x_3, \dots, x_m)$ means exactly one variable is true.

$$\equiv \bigwedge_{1 \le i \le j \le k} (\overline{x_i} \vee \overline{x_j}) \wedge (x_1 \vee x_2 \vee x_3 \vee \dots \vee x_m)$$

No two variables are true

atleast one variable is true

φ_1 : Input is encoded correctly

$$= S(q_0, 0)$$

$$\bigwedge_{h=1}^{n} T(x_N, h, 0) \bigwedge_{h=n+1}^{k(n)} T(\sqcup, h, 0)$$

$$\wedge H(1, 0)$$

$$\varphi = \bigwedge_{i=1}^{12} \varphi_i$$

SAT is NP-hard

Knowing 3SAT is NP-hard

If we show 3SAT $\le_p$ X, X is also NP-hard

NP $\le_p$

Example: 3SAT $\le_p$ Subset-Sum Problem

$\{3, 34, 4, 12, 5, 2\}$ sum = 9 ? Yes

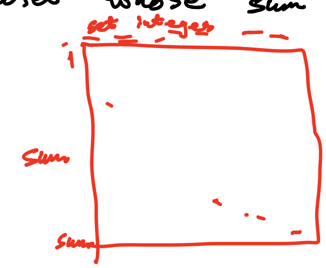
sum = 30 ?

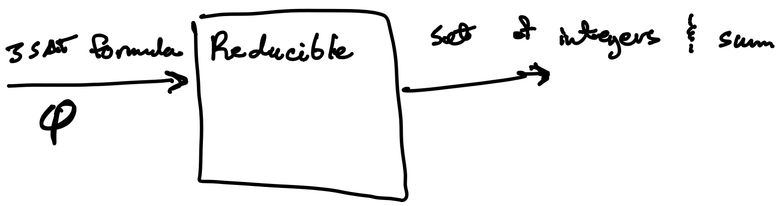
Given a set of ints, is there a non-empty subset whose sum is equal to sum?

Brute Force: $O(2^n \cdot n)$

Recursive: $O(2^n)$

Dynamic: $O(n \cdot \text{sum})$





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

3 variable assignments

4 clauses that need to be satisfied

φ = Any assignment

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

Sum = 1 11 11

$\varphi = x_1 \wedge x_2$

$\overline{x_3}$ 0 0 1 0 0

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

	i			j			
#	1	2	3	1	2	3	4
t_1	1	0	0	1	0	0	1
f_1	1	0	0	0	1	1	0
t_2	0	1	0	1	0	1	0
f_2	0	1	0	0	1	0	1
t_3	0	0	1	1	1	0	1
f_3	0	0	1	0	0	1	0

	i			j			
#	1	2	3	1	2	3	4
x_1	0	0	0	1	0	0	0
y_1	0	0	0	1	0	0	0
x_2	0	0	0	0	1	0	0
y_2	0	0	0	0	1	0	0
x_3	0	0	0	0	0	1	0
y_3	0	0	0	0	0	1	0
x_4	0	0	0	0	0	0	1
y_4	0	0	0	0	0	0	1

SS will use x_j if # of true literals in c_j if at most 2

SS will use y_j if # of true literals in c_j are at most 1

sum = 1 1 1, 3 3 3 3

If 3SAT is Yes then Subset-Sum is also Yes

	i	j
#		

	i	j
#		