



Recall A language (problem) is a set of strings. Typically the goal is to (quickly) decide membership.

$P = \{ \text{languages I can decide membership for in polynomial time} \}$
 $NP = \{ \text{languages I can verify "proofs" for} \}$

Notice $P \subseteq NP$ Question: $P = NP?$ $\rightarrow$ reductions!

Def A is polytime reducible to B, denoted by $A \leq_p B$, if for any input x for A I can compute equivalent inputs $f(x)$ for B in polynomial time: $x \in A \iff f(x) \in B$

Claim $A \leq_p B$ and $B \in P \Rightarrow A \in P$

proof take input x for A, compute $f(x)$, run my algorithm for B on $f(x)$, output result.

Def We say B is NP-hard if $A \leq_p B$ for all $A \in P$.

Def We say B is NP-complete if $B \in (NP \cap \text{NP-hard})$

Theorem (Cook-Levin) SAT is NP-Complete

Recall $SAT = \{ \langle f \rangle \mid f \text{ is a boolean formula for which there exists an assignment of the vars which makes the formula evaluate to TRUE} \}$

Def $3SAT = \{ \langle f \rangle \mid f \text{ is a satisfiable boolean formula that is a bunch of clauses ANDed together where each clause is the OR of 3 boolean vars (or their negatives)} \}$

Claim $SAT \leq_p 3SAT$

Def $SubsetSum = \{ \langle x_1, x_2, \dots, x_n, T \rangle \mid \exists S \subseteq \{x_i\}_{i=1}^n \text{ s.t. } \sum_{x \in S} x = T \}$

Claim $3SAT \leq_p \text{SUBSET SUM}$ (implying SUBSET SUM is NP-Hard)

Proof Take arbitrary boolean formula and compute equivalent inputs for SUBSET SUM

Construction

Example

			C_1	C_2	C_3	
x_1	1 0 0 ...		1	0	1	$(x_1 \vee x_2 \vee x_3)$
$\overline{x_1}$	1 0 0 ...		0	1	0	$\wedge (\overline{x_1} \vee x_3 \vee \overline{x_2})$
x_2	0 1 0 ...		1	0	1	$\wedge (x_1 \vee x_2 \vee \overline{x_3})$
$\overline{x_2}$	0 1 0 ...		0	1	0	
$\vdots$			$\vdots$	$\vdots$	$\vdots$	
C_1			1	0	0	
C_2			1	0	0	
C_3			0	1	0	
C_4			0	1	0	
$\vdots$			$\vdots$	$\vdots$	$\vdots$	
1	1 1 1		3	3	3	

Def $\text{PARTITION} = \{ \langle x_1, \dots, x_n \rangle \mid \exists S \subset \{x_i\}_{i=1}^n \text{ s.t. } \sum_{x \in S} x = \sum_{x \notin S} x \}$

Claim $\text{SUBSET SUM} \leq_p \text{PARTITION}$.

$\langle x_1, \dots, x_n, T \rangle \mapsto y_1, \dots, y_m$

Alg 1 $y_1, \dots, y_m = x_1, \dots, x_n, y_{n+1} = T, y_{n+2} = 2X - T$

where $X = \sum_{i=1}^n x_i$

y_{n+1}	has to be	has to be	y_{n+2}
x_{n+1}	$X - T$	T	$2X - T$

Solving PARTITION means getting a subset that sums to T
Solving SUBSET SUM gives a partition $\square$

claim $\text{PARTITION} \leq_P \text{EF} = \{ \langle G \rangle \mid G \text{ is an allocation problem of indivisible goods} \}$
 $\langle x_1, \dots, x_n \rangle \mapsto$ Make a player game with n items where both players value item i at value x_i .