Decision Problem: is $s \in X$?

(s: string; X: set of strings)

Some decision problems easy, others hard.

A (algorithm) **solves** X if $A(s) = \text{yes}$ iff $s \in X$.

P = set of decision problems **solvable** in polynomial time. (By a deterministic machine)

C is a **certifier** for X if $\forall s \in S \; \exists t \; . \; C(s,t) = \text{yes}$.

NP = set of decision problems **certifiable** in polynomial time. (By a deterministic machine)

What is the N in NP?
Alt. def:

NP: set of decision problems **solvable** in polynomial time **on a nondeterministic machine.**

Reduction: X reduces to Y if there exists a reduction A converting instance of X to instance of Y in **polynomial time** s.t.

$s \in X \iff A(s) \in Y$

Notation: $X \leq_P Y$ — "Y (algorithm) gives solution to X"

$Y \in NP$ is **NP-complete** if $\forall X \in NP \; . \; X \leq_P Y$

X ($\in NP$) is NP-complete if $Y \leq_P X$ for some NP-complete problem — important to establish; note order

this is the result you are going to (ab)use ☺

Solved Exercise 1

n nr. of users
t nr. of minutes
S set of n users
$I(u,m)$ IP address accessed by u in minute m; $\perp$ if none.

k size of attacker koalition (:')

suspicious coalition: Let i_j be IP address accessed at minute j. Then $S' \subseteq S$ is a suspicious coalition if

$$\forall m; 1 \leq m \leq t \;.\; \exists u \in S' \;.\; I(u,m) = i_m$$

Suspicious coalition problem:

$$SCOAL = \{\langle S, I, k\rangle \mid S \text{ has suspicious coalition of size } k \text{ in timespan of } I\}$$

Goal:

Show SCOAL is NP-complete.

Strategy:

- Show $SCOAL \in NP$ (poly-time certifier)
- find (suitable) NP-complete problem Y,
- show (prove) $Y \leq_p SCOAL$

(*): We first need to show SCOAL ∈ NP.

We do this by showing SCOAL is verifiable in poly-time.

Certificate = proof which, if valid, when compared to supposed instance s, proves $s \in$ SCOAL.

In our case: The coalition.

certificate: $U \subseteq S$.

Now we construct verifier.

V = "given S, I, k, and U,

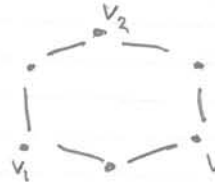
1. if $|U| > k$, return "no". $\mathcal{O}(k)$
2. Foreach minute m,
    1. Check whether there is a user $u \in U$ for which $I(u, m) = 1_m$. If not, return "no". $\mathcal{O}(kt)$
3. return "yes" " $\mathcal{O}(1)$

Total running time: $\mathcal{O}(kt)$, polynomial.

FIRST (*)

Solution: By reduction from VERTEX_COVER (p. 498)

VERTEX_COVER = $\{\langle G, k\rangle \mid G$ has a vertex cover of size at most



$V' = \{v_1, v_2, v_3\}$

Set of vertices V' s.t. each $e \in E$ touches a vertex in V'

$V' = \{v$

$V' = \{v_1, v_2, v_3\}$

Translate instance of VERTEX_COVER to instance of SCOAL.

Given $G = (V, E)$,

- $S = V$
- pick $|E| = t$ IP addresses $i_1, \ldots, i_t$
- map each $(u, u') \in E$ to a minute m ↵ set $I(u, m) := i_m$, $I(u', m) := i_m$

} now we have S, I, k.

(why works: We only need to find a small attack)

Is this sufficient?

Claim: (in this instance (S, G),) S has coalition of size at most k iff G contains vertex cover of size at most k.

proof: Each direction of "iff".

$\Leftarrow$: Let $C \subseteq V$ be the vertex cover of size at most k. We have

$|C| \leq k$
$C \subseteq S$

For each i_m, at least one $u \in C$ accessed i_m at time m. So C is a suspicious coalition. (perhaps even a subset of C is. no matter.)

$\Rightarrow$: Let $C \subseteq S$ be the coalition of size at most k. We have

$|C| \leq k$
$C \subseteq V$

For each i_m, at least one $u \in C$ accessed i_m at time m. Recall, i_m corresponds to an edge $e_m \in E$. So for each edge $e_m \in E$, a user $u \in C$ touches it. So C is a vertex cover. ☒

S.E.2

l guest lectures (one per week)
p hands-on student projects

n speakers
L_i speakers able to give lecture in week i; $1 \leq i \leq l$

P_j speakers capable of speaking about project j; $1 \leq j \leq p$

problem: selecting 1 speaker per week (for first l wee
such that each project will contain at least one
of these speakers? Is this possible?

$$\underbrace{LECTURE\text{-}PLANNING}_{\text{denote } LP} = \{\langle \{L_i\}, \{P_j\} \rangle \mid \ldots\}$$

insert problem here

Goal

Prove that $\underbrace{L\ldots P\ldots NG}_{LP}$ is NP-complete.

Solution: LP is in NP:

certificate: speaker set L

verifier:

V = "on input $L_1, \ldots, L_L, P_1, \ldots, P_P$, and L,

1. Check if $L_i \in L_i$, for each $1 \le i \le L$. If not, return "no".
2. Check if for each P_j, there exists an $L_i \in L$ relevant to P_j. If not, return "no".
3. Return "yes."

V runs in polynomial time.

Time compl. analysis:

$= L_1, \ldots, L_L$

represent L_i by L_j an bool arra $L_j[L_i]$ = true if $L_i \in L_j$. $O(L)$

same with P_j. For each L_i, check $P_j[L_i]$ for all j. If $P_j[L_i]$ = true, no need to check P_j for other L_i. $O(LP)$

$O(L + LP) = O(LP)$

LP is NP-complete:

idea: First "walk down" the list of weeks, assigning a speaker at each. Then "walk down" the projects, checking if each project has a relevant speaker.

similar to: First "walk down" the propositional logic formula, assigning truth values to variables. Then check whether the formula is true.

Using this idea, we can reduce 3SAT to LP.

$3SAT = \{\langle \bigwedge_{i=1}^{k} C_i \rangle \mid$ there exists a variable assignment s.t. $\bigwedge_{i=1}^{k} C_i$ = true$\}$

grammar: $C ::= a_1 \vee a_2 \vee a_3$, $a ::= x \mid \neg x$

Ex: $C_1 = \neg x \vee y \vee z$, $C_2 = \neg x \vee x \vee y$

3SAT example: $(\neg x \vee y \vee z) \wedge (\neg x \vee x \vee y) \wedge (z \vee y \vee x)$

$=: \varphi$ — just a traditional symbol for a propositional formula. You can call it f, or t, if you wish.

another 3SAT example: $(x \vee x \vee x)$.

Assume $s \in 3SAT$. Then, if A computes reduction, $A(s) \in LP$. We do not care what $A(s)$ looks like. Only that

$s \in 3SAT \iff A(s) \in LP$

holds.

Approach: Let $\varphi = \bigwedge^{p} C_i \in 3SAT$. We show how to construct a $s \in LP$ satisfying above-mentioned constraint.

Let $x_1, \dots, x_l$ be the variables in φ.

For each variable x_i, create (out of thin air) two lecturers, L_i and L_i'. L_i corresponds to x_i,
$\mathcal{L}_i = \{L_i, L_i'\}$ L_i' —— " —— $\neg x_i$.

Now let P_j be the set of atoms in C_j.
Illustrative example:

$$\varphi = (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_1 \vee x_2) \wedge (\neg x_3 \vee x_2 \vee x_1)$$

corresp. lecturers	$\{L_1', L_2, L_3\}$	$\{L_1', L_1, L_2\}$	$\{L_3', L_2, L_1\}$
forms our project speaker capability sets	P_1	P_2	P_3

Given a truth assignment of $x_1, \dots, x_l$, the value of x_i determines which speaker L_i, L_i' from $\mathcal{L}_i$ we choose for week i.
If x_i=true, choose L_i'. If x_i=false, then $\neg x_i$=true so we choose L_i'.
(this way, always exactly one speaker gets chosen for each day, since exactly one of $x_i, \neg x_i$ is true, always).

If $L_i \in P_j$, then P_j is "spoken for" if x_i=true.
" $L_i' \in P_j$ —— " —— " x_i=false.

If φ is satisfied by a variable assignment, then each C_j=true. This means at least one of $L \in P_j$ is teaching in some week. } ⊛

In our example, for the satisfying assignment:

x_1 = true		L_1 teaches in week 1	P_2, P_3 covered
x_2 = false	$\Rightarrow$	L_2' —— " —— 2	none
x_3 = true		L_3 —— " —— 3	P_1 covered

all P_j covered.

Claim: In the above construction, All projects can be spoken for iff φ is satisfiable.

proof sketch: each direction of iff: $\Leftarrow$ $\Rightarrow$

$\Leftarrow$: Then there is a satisfying variable assignment. ⊛ completes this argument.

$\neg\Leftarrow\neg$: Then for any assignment, some C_j is false. Thus P_j uncovered.

We are done ▽ o

Note:

☺: You can check that

$p \Leftrightarrow q$ holds iff $p \Rightarrow q$ and $p \overset{q \Rightarrow p}{\Leftarrow} q$

and that

$p \Leftrightarrow q$ holds iff $p \Rightarrow q$ and $\neg p \Rightarrow \neg q$

Book does this proof a little differently.

Note:

There is also a reduction from VERTEX_COVER in book.