

## Question 1: Scheduling

Seminar problem: Given an input  $\{D, L\}$ ,  $D = (D_1, \dots, D_n)$ ,  $L = (L_1, \dots, L_M)$ , The goal is to determine whether there is an assignment of seminars to days that respects the faculty requests while also ensuring that every student gets at least one of their requests satisfied as well.

First, let's show that this problem is in NP.

Given an assignment  $A = (a_1, a_2, \dots, a_n)$  where  $a_i \in \{M, T, W, TH, F\}$  is the day where  $i$ -th faculty will do the seminar, our verifier will check that  $A$  indeed assigns a day for every faculty that meets his availability, and that every student gets at least one of their requests satisfied as well. For a yes instance  $(D, L)$ , our certificate is any Assignment  $A$  that satisfies faculties and students preference, our verifier will accept  $(D, L, A)$ . Conversely, if our verifier accepts  $(D, L, A)$ , then  $A$  is an assignment that satisfies faculties and students preference, and  $(D, L)$  will be a yes instance.

Checking a faculty's preference is satisfied will take  $O(1)$  time (so  $O(n)$  time in total for all faculties). Checking a students preference is satisfied will take  $O(n)$  time (so  $O(n^2)$  time in total for all students). So the running time for this verifier is  $O(n^2)$ .

Therefore the problem is in NP.

We now show that SEMINAR is NP-hard by showing that  $3\text{-SAT} \leq_P \text{SEMINAR}$

Given an instance of 3SAT (a formula  $\phi = C_1 \wedge \dots \wedge C_m$ , where each  $C_i$  is the disjunction of 3 terms drawn from variables  $x_1, x_2, \dots, x_n$  and their negations  $\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}$ ), we construct an instance  $(D, L)$  of SEMINAR problem as follow:

For every variable  $x_i$ , we construct a corresponding faculty  $D_i$  who is willing to teach on Monday and Tuesday ( $D_i = \{M, T\}$ ). And for a clauses  $C_j = (y_{j1} \vee y_{j2} \vee y_{j3})$ , we construct a corresponding student  $L_j$  who wants to have  $j$ -th faculty's seminar on Monday if  $y_{jk} \in \{x_1, x_2, \dots, x_n\}$  and wants to have  $j$ -th faculty's seminar on Tuesday if  $y_{jk} \in \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}$

( For example, if  $C_j = \{x_1, \overline{x_2}, x_3\}$  then  $L_j = \{(1, M), (2, T), (3, M)\}$  )

We now run our black box for SEMINAR on  $(D, L)$  and return the same result it gives.

To prove this answer is correct, we simply need to show that the original instance  $\phi$  of 3SAT is a yes instance (i.e. it is satisfiable) if and only if the the algorithm returns “yes”.

Claim: if there is an assignment for  $(D, L)$  then there is a satisfying assignment for  $\phi$ .

Suppose our algorithm returns “yes,” that is, that there is an assignment A for SEMINAR problem  $(D, L)$ . We show there is satisfying assignment B of  $\phi$  by constructing one: set  $B(x_i) = T$  if in assignment A the  $i$ th faculty will have his seminar on Monday and  $B(x_i) = F$  otherwise.

Since each faculty will do seminar only in one day, each variable is assigned either T or F, but not both, and the assignment B is valid. Moreover, since each student,  $L_j$ , will have at least one of his preference satisfied, at least one variable in  $C_j$  evaluates True.

Claim. If there is a satisfying assignment for  $\phi$  then there is an assignment for SEMINAR problem  $(D, L)$

Suppose  $\phi$  has a satisfying assignment B. We show that our algorithm returns “yes” by constructing an assignment A for SEMINAR problem  $(D, L)$ . Add  $(i, M)$  to A if  $x_i$  is true and add  $(i, T)$  to A if  $x_i$  is False. Each faculty will be assigned one day. And since each clause in 3SAT will be satisfied, at least one of the preference of student will be satisfied. Therefore A is an assignment for SEMINAR  $(D, L)$

Runtime: For each variable in 3SAT problem, we create a faculty for SEMINAR problem. This will take  $O(n)$  time. And for each clause in 3SAT problem, we create a student for SEMINAR problem. This will take  $O(n)$  time as well. So the algorithm takes  $O(n)$  time.

In conclusion, we have shown that SEMINAR is NP and  $3SAT \leq_P SEMINAR$ . Thus, SEMINAR is NP-complete.

## Question 2: Closure Properties

a. Claim:  $A$  is regular iff  $A$  is co-regular.

$\Rightarrow$ :  $A$  is regular  $\Rightarrow \overline{A}$  is regular (regular language is closed under complement)

$\Rightarrow A$  is co-regular (by definition)

$\Leftarrow$ :  $A$  is co-regular  $\Rightarrow \overline{A}$  is co-regular (by definition)

$\Rightarrow A$  is regular (regular language is closed under complement)

b. No. We show this by contradiction:

Suppose this is true. Then we will have context free grammar is closed under complement.

Since we know that context free grammar is closed under union and  $A \cap B = \overline{\overline{A} \cup \overline{B}}$ , we will now have context free grammar is closed under intersection, which is not true.

Therefore, our assumption that context-free equal to co-context-free is not True.

c. Yes.

Given a decidable language  $A$ , we can construct a Turing Machine  $M$  that accepts  $A$  and always halt. If we construct a new TM  $M'$  by swapping the accept states and reject states of  $M$ .  $M'$  will accepts  $\overline{A}$  and always halt. Therefore, decidable language is closed under complement.

Therefore,  $A \in \text{Decidable languages} \Leftrightarrow \overline{A} \in \text{Decidable languages} \Leftrightarrow A \in \text{Co-Decidable languages}$ . i.e. co-decidable equal to decidable.

d. Yes.

Given a problem  $A$  in  $P$ , we will have an algorithm  $L$  that solve this problem in polynomial time. Now for problem  $\overline{A}$ , we could run algorithm  $L$  and return "yes" if  $L$  return "No", and return "no" if  $L$  return "yes". Therefore,  $P$  is closed under complement.

Therefore,  $A \in P \Leftrightarrow \overline{A} \in P \Leftrightarrow A \in \text{Co-P}$ . i.e. co-P equal to  $P$ .

e. If  $P = NP$ , since we know that  $\text{Co-P} = P$ , we will have  $\text{Co-NP} = \text{Co-P} = P = NP$ .

Therefore, if  $\text{co-NP} \neq NP$ ,  $P \neq NP$