Problem

Let ? be a 3cnf-formula. An ≠-assignment to the variables of ? is one where each clause contains two literals with unequal truth values. In other words, an ≠ -assignment satisfies ? without assigning three true literals in any clause.

- **a.** Show that the negation of any \neq -assignment to ? is also an \neq -assignment.
- **b.** Let \neq SAT be the collection of 3cnf-formulas that have an \neq -assignment. Show that we obtain a polynomial time reduction from 3SAT to \neq SAT by replacing each clause c_i

$$(y_1 \lor y_2 \lor y_3)$$

with the two clauses

$$(y_1 \lor y_2 \lor z_i)$$
 and $(\overline{z_i} \lor y_3 \lor b)$,

where z_i is a new variable for each clause ci, and b is a single additional new variable.

c. Conclude that ≠ SAT is NP-complete.

Step-by-step solution

Step 1 of 4

Assumption: is a 3CNF formula.

 ϕ is a clause that contains two literals with unequal truth values of an \neq – assignment to the variable.

CNF is Conjunctive Normal Form. It has the following rules:

- A literal is Boolean variable or negated Boolean variable in the form.
- · Boolean formula is in CNF called a CNF formula.
- · If all clauses have three literals, then it is called 3cnf formula.
- Clause contains several literals connected with $\ \lor \ s$ and $\ \land \ s$.
- Each clause has at least one satisfied literal and one unsatisfied literal in \neq assignment to ϕ .
- The negation of an \neq -assignment conserve this property.

Comment

Step 2 of 4

b)

To show: The formula ϕ is mapped to ϕ then ϕ is satisfiable if ϕ has an \neq - assignment.

Assumption: \neq SAT is the collection of 3cnf formulas that have an \neq -assignment.

• Obtain a polynomial time reduction from 3SAT to \neq -SAT by replacing each close $(y_1 \lor y_2 \lor y_3)$ with the two clause $(y_1 \lor y_2 \lor z_i)$ and $(z_i \lor y_3 \lor b)$

Where

- z_i is a new variable for each clause c_i
- · b is a single additional new variable.

It is known that $SAT = \{(\phi) \mid \phi \text{ is a boolean formula} \}$

· Let ϕ and ϕ are 3 CNF formulas of input and reduction on input ϕ and ϕ as output.

Suppose that $\phi \in 3 - SAT$ • ϕ is satisfiable. • 1 is true and 0 is false. We get an ≠-assignment to ϕ by extending a satisfying assignment to ϕ in such a way that we assign 1 to $\phi \in k \neq SAT$ • Else if both literals y^1 and y^2 are clauses x^2 are unsatisfied, else we assign 0 to x^2 . • Finally, we assign 0 to b. • Extended assignment satisfies ϕ and it is an ≠-assignment to ϕ . Therefore, $\phi \in x SAT$ Comment Step 3 of 4 Suppose that $\phi \in k \neq SAT$ • ϕ has an ≠-assignment of ϕ as follows: • From part(a) we obtain ≠-assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠-assignment to ϕ as follows: • From part(a) we obtain ≠-assignment assign 0 to all y^1, y^2 and y^3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee x_2)$ and $(x_2 \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, SSAT is polynomial time reducible to ≠SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠-assignment.	• We must prove $\phi \in 3-SAT \Leftrightarrow \phi' \in \ne SAT$ therefore the reduction is correct.
 • \$\text{\tex	
• It is true and 0 is false. We get an ≠ - assignment to ∮ by extending a satisfying assignment to ∮ in such a way that we assign 1 to ∮ ∈ k ≠ SAT. • Else if both literals J ¹ and J ² are clauses c ¹ are unsatisfied, else we assign 0 to ± ⁷ . • Entended assignment satisfies ∮ and it is an ≠ - assignment to ∮. • Extended assignment satisfies ∮ and it is an ≠ - assignment to ∮. Therefore, ∮ ∈ x SAT Comment Step 3 of 4 Suppose that ∮ ∈ k ≠ SAT • ≠ has an ≠ - assignment to ∮ as follows: • From part(a) we obtain ≠ - assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠ - assignment cannot assign 0 to all J ¹ L ¹ J ² and J ³ as doing so would force one of the two clauses, (J ³ 1 ∨ J ² 2 ∨ J ² 3) and (Ē 1 ∨ J ³ 3 ∨ J ³ 4), to have all 0's • Hence restricting this assignment to the variables of ∮ yields a satisfying assignment to ∮. Therefore, 3SAT is polynomial time reducible to ≠ SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ - assignment.	
Else if both literals y^1 and y^2 are clauses e^2 are unsatisfied, else we assign 0 to e^2 . Finally, we assign 0 to b. Extended assignment satisfies e^4 and it is an e^2 -assignment to e^4 . Therefore, e^4 ∈ e^2 SAT Comment Step 3 of 4 Suppose that e^4 ∈ e^4 SAT * e^4 has an e^4 -assignment to e^4 as follows: From part(a) we obtain e^4 -assignment assigns 0 to b, otherwise simply negate the assignment. *This e^4 -assignment cannot assign 0 to all e^4 yields a satisfying assignment to the variables of e^4 yields a satisfying assignment to e^4 . *Hence restricting this assignment to the variables of e^4 yields a satisfying assignment to e^4 . Therefore, SSAT is polynomial time reducible to e^4 SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. # e^4 AT ∈ NP, as it is easy to verify whether an assignment assignment assignment.	
Finally, we assign 0 to b. Extended assignment satisfies ϕ and it is an ≠-assignment to ϕ . Therefore, $\phi \in \neq SAT$ Comment Step 3 of 4 Suppose that $\phi \in k \neq SAT$ * ≠ has an **-assignment Satisfying assignment to ϕ as follows: From part(a) we obtain ≠-assignment assigns 0 to b, otherwise simply negate the assignment. *This ≠-assignment cannot assign 0 to all $Y_{i,Y_{i}}$ and Y_{i} as doing so would force one of the two clauses, $(Y_{i} \vee Y_{i} \vee Z_{i})$ and $(\overline{z}_{i} \vee Y_{i} \vee D_{i})$, to have all 0's *Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, 3SAT is polynomial time reducible to ≠SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions *B is in NP *Every A in NP is polynomial time reducible to B. # $SAT \in NP$, as it is easy to verify whether an assignment is an ≠-assignment.	• 1 is true and 0 is false. We get an ≠- assignment to \$\varphi\$ by extending a satisfying assignment to \$\varphi\$ in such a way that we assign 1 to \$\varphi \in K \neq SAI\$.
Extended assignment satisfies ϕ and it is an ≠-assignment to ϕ . Therefore, $\phi \in x SAT$ Comment Step 3 of 4 Suppose that $\phi \in k \neq SAT$ • \neq has an ≠-assignment Satisfying assignment to ϕ as follows: • From part(a) we obtain ≠-assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠-assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \lor y_2 \lor x_3)$ and $(\overline{x_1} \lor y_3 \lor b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, 3SAT is polynomial time reducible to ≠SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠-assignment.	• Else if both literals y_1 and y_2 are clauses c_i are unsatisfied, else we assign 0 to z_i .
Comment Step 3 of 4 Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq - assignment assign 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses. $(y_1 \vee y_2 \vee z_1)$ and $(\overline{z_1} \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, SSAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	• Finally, we assign 0 to b.
Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq -assignment assigns 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_i)$ and $(z_i \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	• Extended assignment satisfies ϕ and it is an \neq – assignment to ϕ .
Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq -assignment assigns 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_i)$ and $(z_i \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	Therefore $\phi' \in \neq SAT$
Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq -assignment assigns 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all $y_{i_1}y_{i_2}$ and y_{i_3} as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_i)$ and $(z_i \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	melelole, ————
Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq - assignment assigns 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all y_1y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_i)$ and $(z_i \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	Comment
Suppose that $\phi' \in k \neq SAT$ • \neq has an \neq - assignment Satisfying assignment to ϕ' as follows: • From part(a) we obtain \neq - assignment assigns 0 to b, otherwise simply negate the assignment. • This \neq - assignment cannot assign 0 to all y_i, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_i)$ and $(\overline{z_i} \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ' yields a satisfying assignment to ϕ' . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	
• ≠ has an ≠ - assignment Satisfying assignment to ∮ as follows: • From part(a) we obtain ≠ - assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠ - assignment cannot assign 0 to all ∫ 1, 1, 2, 2 and ∫ 3 as doing so would force one of the two clauses, (√1 ∨ √2 ∨ √2, 1) and (√2, 1 ∨ √3 ∨ b), to have all 0's • Hence restricting this assignment to the variables of ∮ yields a satisfying assignment to ∳. Therefore, 3SAT is polynomial time reducible to ≠ SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ - assignment.	Step 3 of 4
• ≠ has an ≠ - assignment Satisfying assignment to ∮ as follows: • From part(a) we obtain ≠ - assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠ - assignment cannot assign 0 to all ∫ 1, 1, 1, 2, 2 and ∫ 2, 3 as doing so would force one of the two clauses, (√1 ∨ √2 ∨ √2, 1) and (√2, 1 ∨ √3 ∨ b), to have all 0's • Hence restricting this assignment to the variables of ∮ yields a satisfying assignment to ∳. Therefore, 3SAT is polynomial time reducible to ≠ SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ - assignment.	$\phi \in k \neq SAT$
Satisfying assignment to ϕ as follows: From part(a) we obtain \neq -assignment assigns 0 to b, otherwise simply negate the assignment. This \neq -assignment cannot assign 0 to all y_i, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \lor y_2 \lor z_i)$ and $(\overline{z_i} \lor y_3 \lor b)$, to have all 0's Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq - assignment.	
• From part(a) we obtain ≠ – assignment assigns 0 to b, otherwise simply negate the assignment. • This ≠ – assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \lor y_2 \lor z_1)$ and $(z_1 \lor y_3 \lor b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, 3SAT is polynomial time reducible to ≠SAT Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment.	•
• This ≠ – assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \vee y_2 \vee z_1)$ and $(\overline{z_1} \vee y_3 \vee b)$, to have all 0's • Hence restricting this assignment to the variables of ϕ yields a satisfying assignment to ϕ . Therefore, 3SAT is polynomial time reducible to ≠SAT Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment.	
o's Hence restricting this assignment to the variables of ∮ yields a satisfying assignment to ∮. Therefore, 3SAT is polynomial time reducible to ≠SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ − assignment.	
Therefore, 3SAT is polynomial time reducible to \neq SAT Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. \neq SAT \in NP, as it is easy to verify whether an assignment is an \neq — assignment.	• This \neq -assignment cannot assign 0 to all y_1, y_2 and y_3 as doing so would force one of the two clauses, $(y_1 \lor y_2 \lor z_i)$ and $(z_i \lor y_3 \lor b)$, to have all 0's
Comment Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. $\neq SAT \in NP$, as it is easy to verify whether an assignment is an $\neq -$ assignment.	$ullet$ Hence restricting this assignment to the variables of ${}^{\phi}$ yields a satisfying assignment to ${}^{\phi}$.
Step 4 of 4 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. $\neq SAT \in NP$, as it is easy to verify whether an assignment is an $\neq -$ assignment.	Therefore, 3SAT is polynomial time reducible to ≠SAT
 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment. 	Comment
 (c) NP-COMPLETE: A language B is NP-complete if it satisfies two conditions • B is in NP • Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment. 	
 NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment. 	Step 4 of 4
 NP-COMPLETE: A language B is NP-complete if it satisfies two conditions B is in NP Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment. 	
 B is in NP Every A in NP is polynomial time reducible to B. ≠ SAT ∈ NP, as it is easy to verify whether an assignment is an ≠ – assignment. 	
• Every A in NP is polynomial time reducible to B. $\neq SAT \in NP$, as it is easy to verify whether an assignment is an $\neq -$ assignment.	• B is in NP
	• Every A in NP is polynomial time reducible to B.
	$\neq SAT \in NP$, as it is easy to verify whether an assignment is an $\neq -$ assignment.
Part(h) also proved $35AI \leq_p \neq 5AI \neq SAT$ is NP-complete	Part(b) also proved $3SAT \leq_p \neq SAT$, \neq SAT is NP-complete.
i ditto) diso proved	Tanto and proved f , FOAT IS NI "Complete.
Comment	Comment