
Problem 7.26. Let ϕ be a 3cnf-formula. An ***≠-assignment*** to the variables of ϕ is one where each clause contains two literals with unequal truth values. In other words, an *≠-assignment* satisfies ϕ without assigning three true literals in any clause.

Part a. Show that the negation of any *≠-assignment* to ϕ is also an *≠-assignment*.

Proof. Let ϕ be a 3cnf-formula with n variables. Let $A = (v_1, v_2, \cdots, v_n)$ be an *≠-assignment* to the variables $x_1, x_2, \cdots, x_n$ of ϕ. The negation of A is $A' = (\overline{v_1}, \overline{v_2}, \cdots, \overline{v_n})$. Every clause in ϕ contains at least one literal that is assigned $false$ value by A, and at least one literal that is assigned $true$. Therefore, A' also assigns at lease one literal $false$ and at least one literal $true$ in each clause. Thus, A' is also an *≠-assignment*. □

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

$$(y_1 \vee y_2 \vee y_3)$$

with the two clauses

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

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

Proof. We show that $\phi \in 3SAT$, iff $\phi_{\neq} \in \neq SAT$.

Suppose that ϕ has a satisfying assignment. In that satisfying assignment, at least one literal is true in every clause. Let c_i be any clause $(y_1 \vee y_2 \vee y_3)$ in ϕ. The corresponding two clauses in $\phi_{\neq}$ are

$$(y_1 \vee y_2 \vee z_i) \textbf{ and } (\overline{z_i} \vee y_3 \vee b).$$

An *≠-assignment* satisfying $\phi_{\neq}$ can be constructed as follows:

1. Assign false to b.
2. If y_3 is false, then assign false to z_i.
3. If y_3 is true, then assign true to z_i.

The other direction is trivially true, since $\phi_{\neq}$ is a satisfiable 3cnf-formula. The reduction from $3SAT$ to $\neq SAT$ consists of only one stage that executes m times, where m is the number of clauses in ϕ. Furthermore, this stage can be easily implemented in polynomial time on any reasonable deterministic model. □

Part c. Conclude that $\neq SAT$ is NP-complete.

Proof. $\neq SAT \in$ NP as the assignment is the short certificate. Therefore, $\neq SAT$ is NP-complete. □