HW7: 3 sol-SAT

In order to prove the 3 sol-SAT Problem is NP-complete, I will prove that a solution can be verified in polynomial time and I can reduce SAT → 3 sol-SAT in polynomial time. We know that SAT is NP-complete.

NP Proof:

First, we need to show that the verification of this problem is in NP. Given an input I (n variables and m clauses), we look for 3 distinct assignments of the variables if they exist, and return NO if not.

Given an input I to 3 sol-SAT, and 3 assignments of the variables, what we need to do is (1) Check whether the 3 assignments of variables are distinct. (2) Check the 3 assignments in each clause in I.

(1) When I have 3 assignments of variables (S1, S2, S3), I need to iterate all the value of variables in S1, S2, S3 to check that S1, S2, S3 are distinct. There are n variables, so the time complexity is O(n) + O(n) + O(n) → O(n).

(2) For S1, S2, and S3, we can take each T/F assignment from them and check them in each clause of I. It is O(n) to check each clause. 3 solutions, m clauses result in O(3nm) →O(nm) to verify.

Overall, it takes O(nm) to verify, which is polynomial.

Input:

The original input for SAT is input I. I add a clause c = A∨B to I, where A and B are new variables. There are 3 assignments of (A, B) would make c to be true: (T, T), (T, F), (F, T). Thus, adding the new clause to I would ensure that the number of solutions is to become three times the original. The input for 3 sol-SAT is I ∧ (A∨B), adding 2 variables and 1 clause to I takes O(1) time overall, which is polynomial.

Output:

If the algorithm for 3 sol-SAT Problem returns NO then the SAT also returns NO.

If the algorithm for 3 sol-SAT Problem returns 3 assignments for (n + 2) variables, we return the SAT solution without the created 2 variables as the SAT solution.

This takes O(n) to assigning n variables, and O(1) to return NO, which is polynomial.

Correctness：

If there is a solution S in the SAT, because we added A∨B to form new input for 3 sol-SAT, then the 3 solution would be S + (T, T), S + (T, F), S + (F, T). Therefore, a solution S exists in SAF IFF it corresponds to that 3 solutions could be found in 3 sol-SAT. We could assign n variables without including the 2 created variables, and pass it to the output of SAT.

If there is no solution in the SAT, then 0 * 3 is still zero, which means where would be no solution for 3 sol-SAT neither. So, if 3 sol-SAT Problem returns NO, then so does the SAT.

Thus, as 3 sol-SAT is the class NP, and at least as hard as SAT, we conclude that 3 sol-SAT is NP-complete.