1 Problem 1

1.1 Correctness

Lemma 1. *The VERY-SAT problem is NP-complete*

The problem is in NP since, given a sequence of m clauses, we can check that for each clause of at most $2n$ variables there are at least two different true variables in each clause in polynomial time. Now we will show that a SAT problem can be reduced to VERY-SAT. Note that SAT is NP-complete. Suppose we are given an instance of SAT clauses $C_1, ..., C_k$ over the variables $x_1, ..., x_n$. We construct an instance of VERY-SAT as follows. For each clause C_i, we create one variable $x_i = TRUE$ and take the disjoint of all terms in C_i and x_i. Now, if all clauses have at least 2 true variables, then the instance of VERY-SAT returns YES. As in each clause C_i we inserted a term $x_i = TRUE$, there must exist another term $x_j = TRUE$ from the original clause C_i by definition of VERY-SAT. Thus, the instance of SAT would return YES as at least one term is true. If for some clause C_k, VERY-SAT returns NO, this implies that the input clause had less then 2 true variables. Because we inserted $x_k = TRUE$, this implies all terms the original clause C_k had value FALSE by definition of VERY-SAT. Thus, the instance of SAT would return NO by definition as there exists no solution to the original SAT problem.

1.2 Runtime and Space Complexity

The runtime of the above reduction algorithm is bounded by $O(k)$, where k is the number of clauses, as for each clause C_i, we create one variable $x_i = TRUE$ and take the disjoint of all terms in C_i and x_i, which takes constant time assuming. Note, because the runtime is polynomial, the space complexity is polynomial as well.