# Reduction Algorithm

Given an SAT problem with variables and a formula, turn all in the formula into a new variable in a new formula, which is not negated, and insert a clause . Run the imaginary Monotone SAT code on this new formula with as . The answer from this is the answer for SAT. For example:

Will be reduced to

This reduction algorithm is polynomial because you’re just going through the entire formula and creating a new, non-negated variable for each negated variable (linear, actually). Transforming the output is constant time.

# Proof of Correctness

1. We want to prove that if the answer to is yes, then the answer to is also yes.
2. We want to prove that if the answer to is no, then the answer to is also no.
3. Suppose returns yes, then that means is true by setting at most variables in to true. Since each variable in has a corresponding variable in , setting those corresponding variables to true in also makes true, because the new clauses are obviously true because they’re in actuality disjunctions , and the remaining clauses are also true because it’s essentially the old formula.
4. Suppose returns no, then that means has no satisfiability