

# Project Proposal: Satisfiability Problem (SAT)

- Title: SAT
- Type of project: CSP

## Project Description:

In our project, we will solve the famous SAT problem we studies in `csc373`. We defined SAT problem as follows:

Given a CNF form logical formula over a set of variables, return a variable assignment that makes the whole formula `True`.

The critical assumption we are making here is that the formula are given in **CNF form**, since we will use this to our advantage when building our project. Because we know that every logical formula can be translated into a CNF formula, we can later reduce every formula to a CNF formula and run our algorithm to solve any generic formula. However, for this project, we will only be working with CNF formulas. If we have extra time, we will reduce the algorithm to solve any generic logical formula. This problem is well suited to run CSP over because each variable logically forms possible combinations that can be tested by an easy formula. Each clause in the CNF formula also logically forms constraints that we can prune over, and of course we know our constraint is satisfied if the overall formula is satisfied.

## Evaluation Plan:

We will run this CSP problem using FC and GAC. More specifically, we will try out different ways of setting up our constraints to make the program efficient. We will analyze the space and time complexities of each strategy (such as one giant constraint or chunks of smaller constraints). We also know that SAT is reducible to 3SAT, and we will also try to reduce the problem to 3SAT in polytime and solve the problem with well-structured constraints.

3SAT is the set of SAT that has the form  $(X1 \text{ or } X2 \text{ or } X3) \text{ and } (\dots)$  where each clause only have three literals.

Of course, there will be a lot to play around with in terms of reducing problems to another NP-hard problem, but the overall plan is to build efficient constraints for FC and GAC and run the routine to get a satisfying assignment.