

SAT Solver :

Algorithm : The algorithm used in the program to implement the SAT-Solver is “Backtracking”. In the Backtracking algorithm, first of all, the clause has the smallest size is found out. This is done to make sure that in case a clause has a single literal, it is always assigned “true” value. Then, the literal (contained in the smallest clause found) having maximum occurrence in the whole formula is found. This literal is the decision literal which is assigned “true” first and checked. In case, any contradiction arises, it is backtracked and the literals are assigned “false” and then checked. Thus, whenever any contradiction arises, backtracking is done and finally the model is given in case it is Satisfiable else, it is returned that the formula is Unsatisfiable.

Implementation : The code is written in C++ language.

- The program takes input from the user in the form of a cnf file which contains the formulas in the DIMACS format.
- It then stores each clause in a vector of vectors, clauses. No. of literals is stored in the variable literals and number of clauses is stored in clausesno variable.
- The “solve” function is called which is the main function which checks if the clauses are SAT or UNSAT. The “Solve” function first checks the number of clauses present. If no clause is present, it returns 1 as it shows that all the clauses have been satisfied and the result is SAT.
- Then it calls the function s_clause which returns the index of the clause having the smallest size. If the size is given as 0, it signifies that a clause remains unsatisfied and thus, the “solve” function returns 0.
- Else, another function m_o_p is called which returns the literal (from the clause which has the smallest size), m_prop, which has maximum occurrence in the clauses.
- Then, all the clauses having m_prop are deleted from clauses (signifying that clause is satisfied) and m_prop is deleted from all the clauses where it is present in its negation form in the “trim” function.
- Then “solve” function is called recursively. First, consider that m_prop is true. If solve returns 1, it implies that m_prop is true, else again the solve function is called considering m_prop is false. If the function returns 1, it implies m_prop is false. Otherwise, if it returns 0 in both the cases, it implies that the formula is UNSAT.
- In case of SAT, a model is given which is stored in the result vector.

Assumptions :

- The input is given in the DIMACS format.

Limitations:

- Time taken to run the code increases with increasing number of literals and number of clauses.
- In case of UNSAT, the time taken might get longer than usual as each literal is checked for both true as well as false values.