

CS202-A
MATHEMATICS FOR COMPUTER SCIENCE - II

ASSIGNMENT - 2

SAT SOLVER

SUBMITTED BY-
AVI KUMAR (ROLL NO. - 200229)
PRATHAM JAIN(ROLL NO. - 200712)

Question:

Implement a SAT solver. Given a formula in the DIMACS representation, your implementation should return:

- 1) a model if the formula is satisfiable
- 2) report that the formula is unsatisfiable

We have implemented SAT solver in C++ language using DPLL Algorithm.

The algorithm runs by choosing a literal, assigning a truth value to it, simplifying the formula and then recursively checking if the simplified formula is satisfiable; if this is the case, the original formula is satisfiable; otherwise, the same recursive check is done assuming the opposite truth value.

As it can be seen it splits the problem into two simpler sub-problems. The simplification step essentially removes all clauses that become true under the assignment from the formula, and all literals that become false from the remaining clauses.

We have used these steps to enhance the time complexity of our algorithm.

Unit propagation

If a clause is a *unit clause*, i.e. it contains only a single unassigned literal, this clause can only be satisfied by assigning the necessary value to make this literal true. Thus, no choice is necessary. Unit propagation consists in removing every clause containing a unit clause's literal and in discarding the complement of a unit clause's literal from every clause containing that complement.

Pure literal elimination

If a propositional variable occurs with only one polarity in the formula, it is called *pure*. A pure literal can always be assigned in a way that makes all clauses containing it true. Thus, when it is assigned such a way, these clauses do not constrain the search anymore, and can be deleted.

Finally, Unsatisfiability of a given partial assignment is detected if one clause becomes empty, i.e. if all its variables have been assigned in a way that makes the corresponding literals false. Satisfiability of the formula is detected either when all variables are assigned without generating the empty clause, or, in modern implementations, if all clauses are satisfied. Unsatisfiability of the complete formula is detected after exhaustive search.

In the implementation:

Unit_Propagation(Φ) and *Pure_Literal_Assign(Φ)* are functions that return the result of applying unit propagation and the pure literal rule, respectively, to the literal and the formula Φ .

Unit_Propagation(Φ) detects all clauses of length 1 and applies unit propagation explained above.

Pure_Literal_Assign(Φ) detects all pure literals and applies the Pure literal elimination rule explained above.