Universitat de Lleida

Advanced Programming in Artificial Intelligence

Complete Solver for SAT

Authors:

Francisco Javier Roig Gregorio - 47433543E

Jose Miguel Avellana Lopez - 18068091G

May 28, 2020

The data structure that the solver will use will be a list of clauses, with the literals of each clause (`clauses`), and another list of literals (`vars`) where each literal has its own interpretation, *True* if the literal is true for the interpretation and *False* if the negated literal is true for the interpretation.

From there, the main idea of the algorithm is to obtain an interpretation that is satisfiable or through all the possible interpretations to demonstrate that it is unsatisfiable.

The solver is based on backtracking search. This search begins with an interpretation with all the literals initialized to *None* and in each call (recursive) check if the interpretation for the literal (*True* or *False*) satisfies or not the formula.

Different methods are used to check if the interpretation satisfies the formula, and, if is not, could reduce the charge of the search excluding interpretations.
To make this, the following methods are used:

- `Simplify`, test for each clause if there is a satisfactory literal inside it, if this is true this clause is removed. If the literal that exists within the clause is denied, this literal is removed from the clause, and if this clause is left empty, we can conclude that the current interpretation for the formula is unsatisfactory.

- `Check_unit`, for each clause that have only one literal, if it dont have a value yet, it assign the value to satisfy this clause and it remove the clause, elif it have a value and the clause satisfy it, it remove the clause, else, this set of interpretations can't be satisfiables so return False.

To verify that the formula is satisfactory, the following method is used:

- `is_complete`, it check if all the literals have their interpretation, if this is true, check if it is satisfiable.

- `check_if_satisfiable`, it check if an interpretation is satisfiable checking for each clause if it is satisfactory. If one of them are nor satisfiable, it return false, otherwise, it return true.

We could have use Davis and Putman method with Simplify and Check_unit, but our implementation of it consumed a lot of time checking in what clauses owned every literal. So we exclude this method of our algoritm. We put it comented.