DPLL SAT Solver

20170292 박창현

My SAT solver source file consists of ‘Solver’ class and ‘main()’ function. ‘main()’ function first receives cnf file given on the command line. It reads the file content and parse it to get cnf formula as well as the number of variable and clauses. Then, it passes the information to the ‘Solver’ class and ‘solve’ method decide the satisfiability of given formula.

Let’s take a look at the ‘Solver’ class part. This class has four attributes: nbvar, nbclauses, assignment, formula. ‘nbvar’ stores the number of variables, ‘nbclauses’ stores the number of total clauses. ‘assignment’ has each variable’s assignment information as a list of Boolean value. ‘formula’ is a list of list that contains literals.

‘solve’ method conducts the DPLL algorithm, but it has a few deductive guessing steps. This method returns false if current partial assignment is unsatisfiable, or true if satisfiable. ‘check_clause’ method checks whether the formula has empty clause. Empty clause means that all literals of the clause were assigned to false, so current partial assignment can’t be satisfiable. ‘check_con’ method checks consistency of current formula. Consistency means that the formula’s all literals have only one polarity, so we can assign each literal to the according truth value and get satisfiable model. Therefore, if the formula is consistent, then solve method returns true.

Now there are two deductive assigning steps. First one is ‘Unit propagation’ conducted by method ‘unit_propagate’. It detects the clauses that has only one literal, and assign the truth value to set the clauses true. We don’t have any other assigning choices because the ‘unit clause’ would be false for other assignment. Other step is ‘Pure literal elimination’ conducted by method ‘pure_assign’. If propositional variable occurs with only one polarity, it’s called pure. Pure literals can be assigned to true and all clauses containing them can be eliminated. Until this point, if satisfiability wasn’t decided, then we should assign one variable to some truth value just by guessing. If this guess is decided as unsatisfiable, then do assign it to other value. Recursively method ‘solve’ is called, and it stops when it finds its satisfiable or search step is done (so decided as unsatisfiable).