SAT Solver

CS508 Optimization Methods

Vaibhav Gupta, 194101054

Department of Computer Science and Engineering, IIT Guwahati, India

email: vaibha@iitg.ac.in

DPLL based SAT Solver:

In the field of logic and computer science the DPLL (Davis - Putnam – Logemann – Loveland) algorithm is used for deciding the satisfiability of CNF(Conjuctive Normal Form) formulae. DPLL algorithm is a backtracking based algorithm for solving CNF-SAT problem.

Idea of DPLL:

- First apply unit resolution as long as we can perform it on sentence this step is known as unit propagation.
- If you can not apply unit resolution any further then choose a variable x introduce two cases x and case -x and proceed recursively, this step is known as decision step.
- Here apply unit resolution as long as means:
  - As long as there is unit clause(clause with single literal ℓ) occurs in sentence.
  - Remove -ℓ from all other clauses which contains -ℓ .
  - Remove all clauses containing l as they are redundant now.

Algorithm:

```
DPLL( Sentence ):
    sentence: = unit_propagation( Sentence )
    if ( sentence is empty ):
        return SATISFIABLE
    if( sentence does not contain empty clause ):
        choose variable x in sentence
        DPLL( Sentence + ( x ) )
        DPLL( Sentence + ( -x ) )
```

- DPLL algorithm terminates since in every step size of sentence decreases.
- If DPLL algorithm returns SATISFIABLE then all the unit clauses used in unit propagation steps are a satisfying assignment.
- Otherwise it is the case of big case analysis in which if algorithm yielding empty clause for all cases then sentence will be UNSATISFIABLE.
- So DPLL algorithm is complete procedure to decide satisfiability of CNF-SAT problem.

Example of unsatisfiable:

Consider the CNF consisting of the following nine clauses

1. ¬p ∨ ¬s	2. ¬p ∨ ¬r	3. ¬q ∨ ¬t
4. p ∨ r	5. p ∨ s	6. r ∨ t
7. ¬s ∨ t	8. q ∨ s	9. q ∨ ¬r

Solution:

No unit resolution possible: choose variable p

Add ¬p, unit resolution:	Add p, unit resolution:
remove clause 1 and 2 as it contains -p	remove clause 4 and 5 as it contains p
remove p from clause 4 and 5 it generates	remove -p from clause 1 and 2 it generates
r (4), s (5)	¬s (1), ¬r (2)
perform unit resolution using r and s	perform unit resolution using -s and -r
which yields q and t	which yields q and t
q (use r, 9), t (use s, 7)	q (use ¬s, 8), t (use ¬r, 6)
perform unit resolution using q	perform unit resolution using q
which yields -t	which yields -t
¬t (use q, 3)	¬t (use q, 3)
perform unit resolution using -t	perform unit resolution using -t
which yields ⊥(empty clause)	which yields ⊥(empty clause)
⊥ (use t, ¬t)	⊥ (use t, ¬t)

Both branches yield ⊥(empty clause), so original CNF is **unsatisfiable**.

Example of satisfiable:

Consider the CNF consisting of the following eight clauses

1. ¬p ∨ ¬s	2. ¬p ∨ ¬r	3. ¬q ∨ ¬t
4. p ∨ r	5. p ∨ s	6. r ∨ t
7. ¬s ∨ t	8. q ∨ s	

Solution:

No unit resolution possible: choose variable p

Add -p, unit resolution:	Add p, unit resolution:
remove clause 1 and 2 as it contains -p	remove clause 4 and 5 as it contains p
remove p from clause 4 and 5 it generates	remove -p from clause 1 and 2 it generates
r (4), s (5)	¬s (1), ¬r (2)
perform unit resolution using s	perform unit resolution using -s and -r
which yields t	which yields q and t

t (use s, 7)	q (use ¬s, 8), t (use ¬r, 6)
perform unit resolution using t	perform unit resolution using q
which yields -q	which yields -t
¬q (use t, 3)	¬t (use q, 3)
Yields satisfying assignment	perform unit resolution using t
p = q = false, r = s = t = true	which yields ⊥ (empty clause)
	⊥ (use t, ¬t)

As unit resolution on -p yields satisfying assignment p = q = false, r = s = t = true so above CNF is **satisfiable**.

Code Organization:

The whole code of sat solver is divided into five methods which are used in combination to solve the CNF – SAT problem. These methods are -

- unit_resolution()
- unit_propagation()
- jeroslow_wang_2_sided()
- dpll()
- main()

unit_resolution() method:

- This method takes sentence and variable x (by using which we have to prform the unit resolution step) as input.
- It removes all the clauses from sentence which contains x as these clauses are redundant and it removes -x from all the clauses which contains -x.
- This method returns 1 if it encounter any empty clause after removing -x from any clause as it means that this unit resolution step results unsatisfiable.
- If it does not encounter any empty clause than it returns updated sentence which is obtained after performing all the above steps. The whole method is commented thouroughly for better understanding.

unit_propagation() method:

- This method takes sentence as input. And returns updated sentence and all the unit assignments.
- It iterate through whole sentence to find all the unit clauses and after finding list of unit clauses. (step 1)
- It selects first unit clause as this unit clause contains variable x or -x so it will now call unit_resolution() method with sentence and this variable as input. (step 2)
- unit_resolution() method will return 1 or updated sentence. (step 3)
    - If 1 is returned by unit_resolution method then it means unit resolution on variable results unsatisfiable so unit_propagation() method returns 1 and empty list.
    - If updated sentence is returned by unit_resolution() method then execution will again start from step 1.

- If now there is no unit clauses in sentence then it will return updated sentence and all the unit assignments. The whole method is commented thouroughly for better understanding.

jeroslow_wang_2_sided() method:

- This method is implemented for applying Jeroslow – Wang – Two – Sided decision heuristic which is used for selecting the literal which will be used in decide step of dpll() method.
- This method takes sentence as input and returns a literal which is having the highest score.
- This method gives score to each literal which depends on the length of clause in which it belongs. The score is inversely proportional to the length of clause.

$$\text{Score}(x) = \Sigma \left(2^{-|c|} \right) \text{ for each clause c that contains x or -x}$$

- Where |c| is length of clause c.
- So this method gives higher score to the literals which belongs to smaller clauses and literal with maximum score is returned. The whole method is commented thouroughly for better understanding.

dpll() method:

- This method takes sentence and current unit assignment list as input and returns final solution after completing execution of all recursive calls.
- In the starting this method calls unit_propagation() method which returns updated sentence and unit assignments.
- If sentence returned by unit_propagation() method is 1 then that means the sentence is unsatisfiable so dpll method returns empty solution and terminates.
- If sentence returned by unit_propagation() method is empty then it returns assignment list which contains assignment of all leterals and terminates.
- If sentence is not empty then it calls jeroslow_wang_2_sided() method to chosoe literal ℓ for decide step.
- After that dpll() method is called with literal ℓ and if it returns empty assignment then it means that literal ℓ results unsatisfiable. And if empty assignment list is returned than another recursive call of dpll() method is called with literal $-\ell$.
- After completion of all recursive calls it returns a solution list. If solution list is empty then it means sentence is unsatisfiable otherwise solution list contains assignment of each literal. The whole method is commented thouroughly for better understanding.

main() method:

- This method is the driver method of whole code. In this method at start t takes name of input file and output file as input from user.
- After getting the name of input file it opens the input file and parse the whole input file to create sentence.

- After creating the sentence it checks for the empty clause in sentence, and if there is any empty clause is present then it means given sentence is unsatisfiable so it writes unsatisfiable in output file and terminates.
- If sentence does not contain empty clause then it calls dpll() method with sentence and empty list of assignments.
- dpll() method returns solution list after complete execution. If solution list is empty then it means sentence is unsatisfiable so it writes unsatisfiable in output file and terminates.
- If solution list returned by dpll() method is not empty then it means the sentence is satisfiable and solution list contains assignment of each literal. So it writes satisfiable and state of each literal in output file. The whole method is commented thouroughly for better understanding.