CS 228 : Logic in Computer Science

Krishna. S

The DPLL Algorithm

Input : CNF formula F.

1. Initialise α as the empty assignment
2. While there is a unit clause L in $F|\alpha$, add $L = 1$ to α (unit propagation)
3. If $F|\alpha$ contains no clauses, then stop and output α
4. If $F|\alpha$ contains the empty clause, then apply the learning procedure to add a new clause C to F. If it is the empty clause, output UNSAT. Otherwise, backtrack to the highest level at which C is a unit clause, go to line 2.
5. Decide on a new assignment $p = b$ to be added to α, goto line 2.

$c_1 = \neg p_1 \vee p_2$

$c_2 = \neg p_1 \vee p_3 \vee p_5$

$c_3 = \neg p_2 \vee p_4$

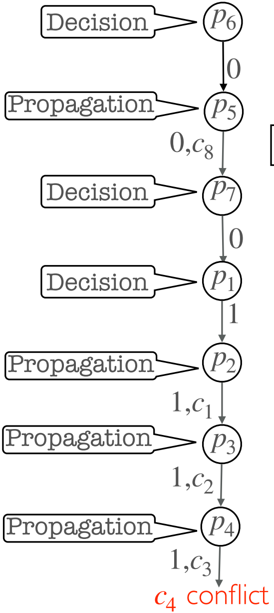
$c_4 = \neg p_3 \vee \neg p_4$

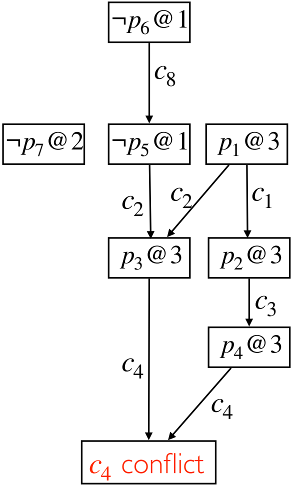
$c_5 = p_1 \vee p_5 \vee \neg p_2$

$c_6 = p_2 \vee p_3$

$c_7 = p_2 \vee \neg p_3 \vee p_7$

$c_8 = p_6 \vee \neg p_5$





Conflict Clause

If (F, α) is the state of the algorithm, then we say that a clause C is a conflict clause if all literals in C have been made false by α.

Learnt Clause

Traverse the implication graph backwards to find the set of decisions that created a conflict. The negations of the causing decisions is the learnt clause.

$c_1 = \neg p_1 \vee p_2$

$c_2 = \neg p_1 \vee p_3 \vee p_5$

$c_3 = \neg p_2 \vee p_4$

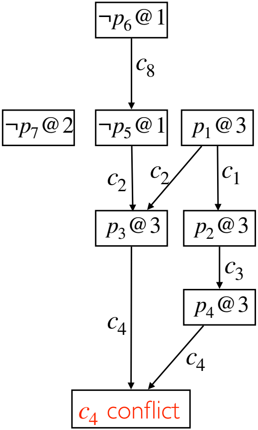
$c_4 = \neg p_3 \vee \neg p_4$

$c_5 = p_1 \vee p_5 \vee \neg p_2$

$c_6 = p_2 \vee p_3$

$c_7 = p_2 \vee \neg p_3 \vee p_7$

$c_8 = p_6 \vee \neg p_5$



Learnt clause : $p_6 \vee \neg p_1$. The algorithm uses resolution to compute learnt clauses (resolve c_4 with c_3, c_1, c_2, c_8)

Learnt Clause

If (F, α) is a conflict state, and a clause C is learnt, then

- C is a conflict clause (all literals in C have been made false by α)
- All variables in C are decision variables
- Adding C to F gives a formula equivalent to F

Clause Learning and Backtracking

- We add the learnt clause to the input set of clauses

Adding the learnt clause

- does not affect satisfiability of the original formula (think of resolution)
- ensures that the conflicting partial assignment will not be tried again

- backtrack to the highest decision level where the learnt clause is unit and then proceed like DPLL

$c_1 = \neg p_1 \vee p_2$

$c_2 = \neg p_1 \vee p_3 \vee p_5$

$c_3 = \neg p_2 \vee p_4$

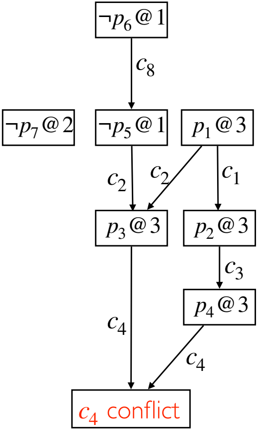
$c_4 = \neg p_3 \vee \neg p_4$

$c_5 = p_1 \vee p_5 \vee \neg p_2$

$c_6 = p_2 \vee p_3$

$c_7 = p_2 \vee \neg p_3 \vee p_7$

$c_8 = p_6 \vee \neg p_5$



Learnt clause is $p_6 \vee \neg p_1$. The highest decision level where this is a unit clause is level 1, with the decision $p_6 = 0$. Unit propagation will force $p_1 = 0$. The combination $p_6 = 0, p_1 = 1$ will not be tried again.

$c_1 = \neg p_1 \vee p_2$

$c_2 = \neg p_1 \vee p_3 \vee p_5$

$c_3 = \neg p_2 \vee p_4$

$c_4 = \neg p_3 \vee \neg p_4$

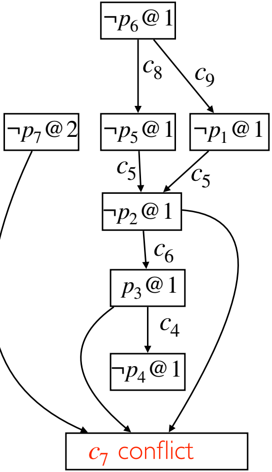
$c_5 = p_1 \vee p_5 \vee \neg p_2$

$c_6 = p_2 \vee p_3$

$c_7 = p_2 \vee \neg p_3 \vee p_7$

$c_8 = p_6 \vee \neg p_5$

$c_9 = p_6 \vee \neg p_1$



Learnt clause : $p_7 \vee p_6$ is added and backtrack

Set $p_7 = 1$ by unit propagation.

$c_1 = \neg p_1 \vee p_2$

$c_2 = \neg p_1 \vee p_3 \vee p_5$

$c_3 = \neg p_2 \vee p_4$

$c_4 = \neg p_3 \vee \neg p_4$

$c_5 = p_1 \vee p_5 \vee \neg p_2$

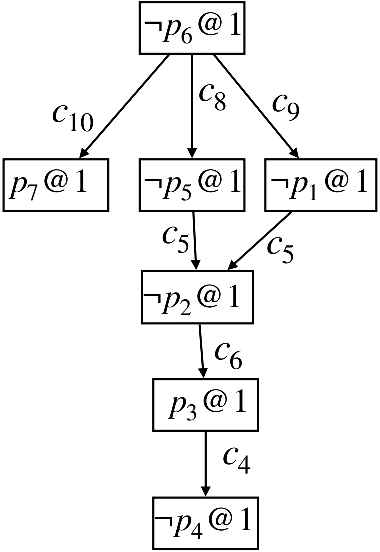
$c_6 = p_2 \vee p_3$

$c_7 = p_2 \vee \neg p_3 \vee p_7$

$c_8 = p_6 \vee \neg p_5$

$c_9 = p_6 \vee \neg p_1$

$c_{10} = p_6 \vee p_7$



DPLL concludes SAT and returns α

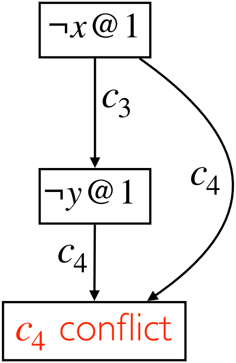
Another DPLL Example

$c_1 = \neg x \vee \neg y$

$c_2 = \neg x \vee y$

$c_3 = x \vee \neg y$

$c_4 = x \vee y$



Clause learnt : x (Resolve c_4 with c_3)

Another DPLL Example

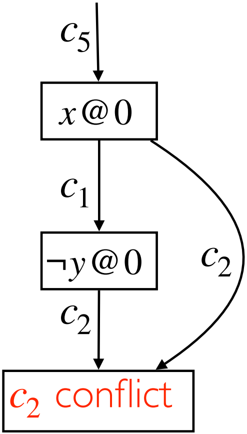
$c_1 = \neg x \vee \neg y$

$c_2 = \neg x \vee y$

$c_3 = x \vee \neg y$

$c_4 = x \vee y$

$c_5 = x$



Clause learnt : Resolve c_2 with c_1, c_5. Empty clause. DPLL concludes UNSAT.

Prove!

Suppose that (F, α) is a conflict state and C be a learnt clause. Then C is a conflict clause, all variables appearing in C are decision variables in α, and F is equivalent to the formula obtained by adding clause C to F.

DPLL Correctness

Termination

A sequence of decisions which lead to a conflict cannot be repeated : the variables in the learned clause are all decision variables. In a future assignment, if all but one of these are set to false, the remaining one will not be a decision variable.

DPLL Correctness

Termination

A sequence of decisions which lead to a conflict cannot be repeated : the variables in the learned clause are all decision variables. In a future assignment, if all but one of these are set to false, the remaining one will not be a decision variable.

Correctness

Correctness is straightforward : $F \vdash$ the learned clause. Thus, if the empty clause is learnt, then F is unsat. Otherwise, if DPLL terminates with a satisfying assignment α, then the input formula is also satisfied by α.

Modern SAT Solvers

Numerous enhancements/heuristics

- Decision heuristics to choose decision variables
- Random restarts

First Order Logic

FOL

Extends propositional logic

- Propositional logic : atomic formulas have no internal structure
- FOL : atomic formulas are predicates that assert a relationship between certain elements
- Quantification in FOL : ability to assert that a certain property holds for all elements or only for some element.
- Formulae in FOL are over some signature.