Learning from Conflicts (CDCL)

The DPLL algorithm can meet the same conflict several times. Modern SAT solvers extend DPLL with conflict-driven clause learning (CDCL), which avoids this [1]. This sheet recalls BCP and DPLL (Section 1), runs DPLL into a repeated conflict, and then runs CDCL on the same formula (Section 2).

1 Boolean constraint propagation and DPLL

A formula in CNF is a conjunction of clauses $c_1, \ldots, c_m$. Under a partial assignment v, a clause $\ell_1 \vee \ell_2 \vee \cdots \vee \ell_m$ is

- *satisfied* if one or more of its literals ℓ_i are satisfied,
- *conflicting* if all of its literals ℓ_i are false,
- *unit* if it is not satisfied and all of its literals ℓ_i but one are false,
- *unresolved* otherwise.

Boolean constraint propagation (BCP) extends v with the steps below [1]. The formula column of a table lists the clauses that are not satisfied, without their false literals.

(B1) **Conflict or no-unit?:** If a clause is conflicting, stop and report the lowest-numbered conflicting one. If no clause is unit, stop. Otherwise, continue with (B2).

(B2) **Unit:** Take the lowest-numbered unit clause, and give the variable of its unassigned literal the value that satisfies this literal (the *unit clause rule*).

(B3) **Simplify:** Delete satisfied clauses, and false literals from the other clauses. A unit clause then has one literal, and a conflicting clause has zero. Go to (B1).

Giving a fixed order to the variables of a formula, the DPLL algorithm executes the steps below [2, 1], starting from the empty assignment.

(D1) **BCP:** Run BCP, steps (B1)–(B3).

(D2) **Conflict? ⇒ flip, or unsat:** If a clause is conflicting and every decision has been flipped, then stop: the formula is unsatisfiable. If a clause is conflicting otherwise, *backtrack*: undo the most recent unflipped decision (see (D4)) and all later assignments, and *flip* the value of its variable. Go to (D1).

(D3) **All sat?:** If every clause is satisfied, stop: the formula is satisfiable, and the variables that are still unassigned may take any value in its model.

(D4) **Decide:** Otherwise, *decide*: set the first unassigned variable of the fixed order to 1. This value may be flipped later at (D2). Go to (D1).

2 Learning from conflicts

Modern SAT solvers extend DPLL with conflict-driven clause learning (CDCL) [1]. To show why clause learning is needed, we first run DPLL on a small formula.

Exercise 1 (DPLL repeats a conflict). Run DPLL on the clauses below, in the variable order $x_1, \ldots, x_6$.

$$c_1 = (\neg x_1 \vee \neg x_3 \vee x_4) \quad c_2 = (\neg x_1 \vee \neg x_3 \vee \neg x_4) \quad c_3 = (\neg x_1 \vee x_3 \vee x_5)$$
$$c_4 = (\neg x_1 \vee x_3 \vee \neg x_5) \quad c_5 = (x_1 \vee x_2) \quad c_6 = (\neg x_2 \vee x_6)$$

The table below shows the run with the steps of Section 1, and numbers the conflicts #1, #2, ... Complete the missing reasons, and the empty rows.

Step	Reason	$v(x_i)$						Formula
		x_1	x_2	x_3	x_4	x_5	x_6	
start		—	—	—	—	—	—	$c_1, \ldots, c_6$
(D4)	decision	1	—	—	—	—	—	$c_1 = (\neg x_3 \vee x_4)$, $c_2 = (\neg x_3 \vee \neg x_4)$, $c_3 = (x_3 \vee x_5)$, $c_4 = (x_3 \vee \neg x_5)$, $c_6 = (\neg x_2 \vee x_6)$
(D4)	decision	1	1	—	—	—	—	$c_1 = (\neg x_3 \vee x_4)$, $c_2 = (\neg x_3 \vee \neg x_4)$, $c_3 = (x_3 \vee x_5)$, $c_4 = (x_3 \vee \neg x_5)$, $c_6 = (x_6)$
(B1)–(B3)	unit c_6	1	1	—	—	—	1	$c_1 = (\neg x_3 \vee x_4)$, $c_2 = (\neg x_3 \vee \neg x_4)$, $c_3 = (x_3 \vee x_5)$, $c_4 = (x_3 \vee \neg x_5)$
(D4)	decision	1	1	1	—	—	1	$c_1 = (x_4)$, $c_2 = (\neg x_4)$
(B1)–(B3)	unit c_1	1	1	1	1	—	1	$c_2 = ()$
(B1)	conflicting c_2 (#1)	1	1	1	1	—	1	$c_2 = ()$
(D2)	_____	1	1	0	—	—	1	$c_3 = (x_5)$, $c_4 = (\neg x_5)$
(B1)–(B3)	_____	1	1	0	—	1	1	$c_4 = ()$
(B1)	_____	1	1	0	—	1	1	$c_4 = ()$
(D2)	_____	—	—	—	—	—	—	_____
(D4)	_____	—	—	—	—	—	—	_____
(B1)–(B3)	_____	—	—	—	—	—	—	_____
(B1)	_____	—	—	—	—	—	—	_____
(D2)	_____	—	—	—	—	—	—	_____
(B1)–(B3)	_____	—	—	—	—	—	—	_____
(B1)	_____	—	—	—	—	—	—	_____
(D2)	_____	0	—	—	—	—	—	$c_5 = (x_2)$, $c_6 = (\neg x_2 \vee x_6)$
(B1)–(B3)	_____	0	1	—	—	—	—	$c_6 = (x_6)$
(B1)–(B3)	_____	0	1	—	—	—	1	no clause

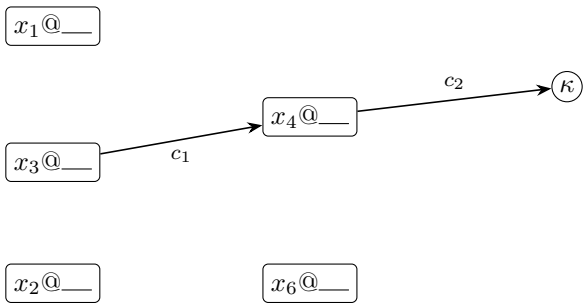
Answer:

- How many conflicts did DPLL meet? ____
- Which earlier conflicts do #3 and #4 repeat, clause and formula included? _____
- Which decided variable has a different value at those earlier conflicts? ____

Exercise 2 (Reading an implication graph). CDCL counts the decisions: the *decision level* is the number of decisions made so far, and the notation $\ell@d$ represents the literal ℓ made true at level d. At a conflict, CDCL builds the *implication graph* of the assignment [1].

- Its nodes are the literals with their levels ($\ell@d$) or the node κ denoting conflicts.
- Edges are labelled with the clause c_i that forced the node they enter:
  - Either $c_i = (\ell_1 \vee \cdots \vee \ell_k \vee \ell)$ was unit and BCP made ℓ true at level d. Then $\neg\ell_j@d_j \xrightarrow{c_i} \ell@d$ for each j,
  - or $c_i = (\ell_1 \vee \cdots \vee \ell_k)$ was conflicting. Then $\neg\ell_j@d_j \xrightarrow{c_i} \kappa$ for each j.
  - Decision nodes therefore have no incoming edges.

The incomplete implication graph below is that of conflict #1 in Exercise 1. Complete it.



Exercise 3 (Running CDCL). CDCL takes a formula φ in CNF, given as a list of clauses $c_1, \ldots, c_m$, and a fixed variable order. It keeps an assignment v, a current level d and a list of clauses $c_1, \ldots, c_n$, starting with the empty assignment, $d = 0$ and $n = m$. Unlike DPLL, it extends the list at each conflict with a *learned clause*, which follows from φ. The list therefore keeps the models of φ. CDCL repeats the steps below [1].

(C1) **BCP:** Run BCP on $c_1, \ldots, c_n$ (Section 1). Each literal ℓ that it makes true is at level d: it is the node $\ell@d$.

(C2) **Conflict ⇒ learn, or unsat:** If a clause is conflicting and $d = 0$, stop: φ is unsatisfiable. If a clause is conflicting and $d > 0$, *learn*: let $\ell_1@d_1, \ldots, \ell_k@d_k$ be the decision nodes (without incoming edges) from which κ can be reached in the implication graph of v. Append $c_{n+1} = (\neg\ell_1 \vee \cdots \vee \neg\ell_k)$ to the list, and increase n by 1.

(C3) **Backjump:** Let d' be the second highest level among $d_1, \ldots, d_k$ in (C2). Undo every assignment at a level higher than d', and set d to d'. Go to (C1).

(C4) **All sat?:** If $c_1, \ldots, c_n$ are all satisfied, stop: φ is satisfiable.

(C5) **Decide 1:** Otherwise, *decide*: increase d by 1, and set $v(x) = 1$ for the first unassigned variable x of the order. It is the node $x@d$. Go to (C1).

Note: In (C2), real solvers may have a different method to choose the $\ell_1@d_1, \ldots, \ell_k@d_k$, e.g. they can be chosen so that every path from a decision to κ passes through one of them (the *first unique implication point* [1]). In this exercise, this choice gives the same clauses c_7 and c_8.

Part 1 — *Learn and backjump.* Run CDCL on the clauses $c_1, \ldots, c_6$ of Exercise 1 ($m = 6$), in the same order. Up to conflict #1, its run is that of DPLL, with the implication graph of Exercise 2. Complete the sentences below.

- The learned clause: $c_7 =$ _____.
- The learned clause is implied by the formula: the clauses c_1 and c_2 differ only in the literals x_4 and $\neg x_4$, so both hold only if _____ holds.
- The literals of c_7 have the levels _____, so CDCL backjumps to level ___.
- The backjump undoes the values of _____.

Part 2 — *Continue.* Continue the run up to the next conflict. Complete the table.

Step	Reason	Level	$v(x_i)$						Formula
			x_1	x_2	x_3	x_4	x_5	x_6	
(B1)	conflicting c_2 (#1)	3	1	1	1	1	—	1	$c_2 = ()$
(C2)–(C3)	learn c_7, backjump to level 1	1	1	—	—	—	—	—	$c_1 = (\neg x_3 \vee x_4)$, $c_2 = (\neg x_3 \vee \neg x_4)$, $c_3 = (x_3 \vee x_5)$, $c_4 = (x_3 \vee \neg x_5)$, $c_6 = (\neg x_2 \vee x_6)$, $c_7 = (\neg x_3)$
(B1)–(B3)	_____	__	__	__	__	__	__	__	_____
(B1)–(B3)	_____	__	__	__	__	__	__	__	_____
(B1)	_____	__	__	__	__	__	__	__	_____

Part 3 — *Learn again.* Draw the implication graph at conflict #2: add the missing node and the labelled edges.



The learned clause: $c_8 =$ _____. Continue the run until CDCL stops.

Step	Reason	Level	$v(x_i)$						Formula
			x_1	x_2	x_3	x_4	x_5	x_6	
(C2)–(C3)	learn c_8, backjump to level ___	__	__	__	__	__	__	__	$c_1, \ldots, c_8$

Step	Reason	Level	$v(x_i)$						Formula
			x_1	x_2	x_3	x_4	x_5	x_6	
(B1)–(B3)									
(B1)–(B3)									
(B1)–(B3)									

Part 4 — *Compare.* Complete the comparison of the two runs on the same clauses.

	DPLL (Exercise 1)	CDCL
Conflicts		
Verdict		

Mark with a checkmark (✓) why CDCL met fewer conflicts:

( ) CDCL propagates more unit clauses among the original clauses $c_1, \ldots, c_6$.

( ) The learned clause c_7 records that $v(x_1) = v(x_3) = 1$ is a conflict whatever the value of x_2, and the backjump skips the decision on x_2, which caused no conflict.

( ) CDCL decides the variables in a better order than DPLL.

References

[1] Daniel Kroening and Ofer Strichman. *Decision Procedures: An Algorithmic Point of View*, 2nd edition. Springer, 2016.

[2] Armin Biere, Marijn Heule, Hans van Maaren and Toby Walsh (editors). *Handbook of Satisfiability*. IOS Press, 2009.