Propositional Logic
CDCL: Conflict Driven Clause Learning

CDCL: goal and idea

Goal: Combine DPLL and resolution into an algorithm oriented towards both satisfiability and unsatisfiability.

Idea: At every unsuccessful leaf of DPLL (called conflict), compute a conflict clause, and add it to the formula we are deciding about.

Conflict clauses "cache" previous search results, so we "learn from previous mistakes".

Conflict clauses also determine backtracking.

We present a particular way of computing a conflict clause using resolution. There are other ways.

DPLL + CDCL algorithm

Given formula F and partial assignment $\mathcal{A}$:
$F|_{\mathcal{A}}$ denotes the result of deleting any clause containing a true literal, and deleting all false literals from each remaining clause.

Input: CNF formula F.

1. Initialise $\mathcal{A}$ to the empty assignment
2. While there is unit clause $\{L\}$ or pure literal L in $F|_{\mathcal{A}}$, update $\mathcal{A} \mapsto \mathcal{A}[\top/L]$
3. If $F|_{\mathcal{A}} = \emptyset$, stop and output $\mathcal{A}$.
4. If $F|_{\mathcal{A}} \ni \Box$, add new clause C to F by learning procedure.
   If $C = \Box$, stop and output UNSAT; otherwise backtrack to highest level where C is unit clause.
   Go to line 2.
5. Apply decision strategy to update $\mathcal{A}$.
   Go to line 2.

Terminology

- State of algorithm is pair $(F, \mathcal{A})$, where F is CNF formula and $\mathcal{A}$ is partial assignment.
  Successful state when $\mathcal{A} \models F$. Conflict state when $\mathcal{A} \not\models F$.

  (Note: conflict state if $F|_{\mathcal{A}} \ni \square$, successful state if $F|_{\mathcal{A}} = \emptyset$)
- Each assignment $A_i \mapsto b_i$ classifies as decision assignment or implied assignment.
- $A_i \mapsto b_i$ denotes decision assignment with decision variable A_i.
- $A_i \overset{C}{\mapsto} b_i$ denotes an implied assignment arising through unit propagation on clause C.
- Decision level of assignment $A_i \mapsto b_i$ in a given state $(F, \mathcal{A})$ is number of decision assignments in $\mathcal{A}$ that precede $A_i \mapsto b_i$.

Example: start with set of clauses $F = \{C_1, \ldots, C_5\}$, where

$$\begin{aligned} C_1 =& \{\neg A_1, \neg A_4, A_5\} \\ C_2 =& \{\neg A_1, A_6, \neg A_5\} \\ C_3 =& \{\neg A_1, \neg A_6, A_7\} \\ C_4 =& \{\neg A_1, \neg A_7, \neg A_5\} \\ C_5 =& \{A_1, A_4, A_6\} \end{aligned}$$

Say current assignment is $(A_1 \mapsto 1, A_2 \mapsto 0, A_3 \mapsto 0, A_4 \mapsto 1)$.
Notice $F|_{\mathcal{A}}$ contains unit clause $\{A_5\}$.
Unit propagation further generates $(A_5 \overset{C_1}{\mapsto} 1, A_6 \overset{C_2}{\mapsto} 1, A_7 \overset{C_3}{\mapsto} 1)$.
This leads to a conflict, with C_4 being made false.

Conflict analysis

After unit propagation:

- If not in conflict nor successful, make decision (line 5)
- If in conflict, learned clause is added (line 4)

Learned clause desiderata: If unit propagation from state $(F, \mathcal{A})$ leads to conflict, clause C is learned such that:

1. $F \equiv F \cup \{C\}$
2. C is conflict clause: each literal of C is made false by $\mathcal{A}$
3. C mentions only decision variables in $\mathcal{A}$

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \ldots, A_k \mapsto b_k)$ leads to conflict.
Find associated clauses $D_1, \ldots, D_{k+1}$ by backward induction:

1. $D_{k+1} :=$ any conflict clause of F under $\mathcal{A}$.
2. If $A_i \mapsto b_i$ is decision assignment or A_i not mentioned in D_{i+1}, set $D_i := D_{i+1}$.
3. If $A_i \overset{C_i}{\mapsto} b_i$ is implied assignment and A_i mentioned in D_{i+1}, define D_i to be resolvent of D_{i+1} and C_i with respect to A_i.

$C := A_1$, that is, the final clause A_1 is the learned clause .

Clause learning: example

Conflict of example above:

$C_1 = \{\neg A_1, \neg A_4, A_5\}$	$D_8 := \{\neg A_1, \neg A_7, \neg A_5\}$	(clause C_4)
$C_2 = \{\neg A_1, A_6, \neg A_5\}$	$D_7 := \{\neg A_1, \neg A_5, \neg A_6\}$	(resolve D_8, C_3)
$C_3 = \{\neg A_1, \neg A_6, A_7\}$	$D_6 := \{\neg A_1, \neg A_5\}$	(resolve D_7, C_2)
$C_4 = \{\neg A_1, \neg A_7, \neg A_5\}$	$D_5 := \{\neg A_1, \neg A_4\}$	(resolve D_6, C_1)
$C_5 = \{A_1, A_4, A_6\}$	$D_4 := \{\neg A_1, \neg A_4\}$	
$A_1 \mapsto 1, A_2 \mapsto 0,$	$D_3 := \{\neg A_1, \neg A_4\}$	
$A_3 \mapsto 0, A_4 \mapsto 1,$	$D_2 := \{\neg A_1, \neg A_4\}$	
$A_5 \overset{C_1}{\mapsto} 1, A_6 \overset{C_2}{\mapsto} 1,$	$D_1 := \{\neg A_1, \neg A_4\}$	
$A_7 \overset{C_3}{\mapsto} 1$		

Learned clause D_1 is conflict clause with only decision variables, including top-level one A_1.

Clause learning: example

Intuitively:

- D_1 records that conflict due to decision to make A_1, A_4 true.
- Adding D_1 ensures search does not explore assignments with $A_1 \mapsto 1, A_4 \mapsto 1$.
- DPLL backtracks to highest level where D_1 is unit clause (after $A_1 \mapsto 1$), unit propagation leads to $A_4 \mapsto 0$.

Clause learning

Proposition: The clause learning procedure satisfies the three desiderata.

Proof sketch: Observation: If $A_i \overset{C_i}{\mapsto} b_i$, then the only literal of C_i true under $\mathcal{A}$ is the literal for A_i (that is, C_i contains either A_i or $\neg A_i$, and b_i is chosen to make the literal true).

1. $F \equiv F \cup \{C\}$
Because C is obtained from clauses of F through resolution steps.

2. C is conflict clause: each literal is made false by $\mathcal{A}$.
We show by induction that $D_{k+1}, D_k, \cdots D_1 = C$ are conflict clauses.
D_{k+1} is conflict clause by definition.
If D_{i+1} is conflict clause and $D_i = D_{i+1}$, then so is D_i.
If D_{i+1} is conflict clause and $D_i \neq D_{i+1}$, then D_i is the result of resolving D_{i+1} and C_i. By the observation, all literals of D_i are made false by $\mathcal{A}$.

3. C mentions only decision variables in $\mathcal{A}$.
Because every other variable, say A_i, disappears after resolving with D_{i+1} w.r.t. A_i.
Indeed, since $\mathcal{A}$ makes D_{i+1} false, by the observation A_i has opposite signs in D_{i+1} and C_i.

Example (without PLR)

$$\{\neg A_1\}\{A_1, A_3, A_4\}\{\neg A_2, \neg A_5\}\{A_3, \neg A_4, A_5, \neg A_6\}\{A_1, \neg A_2, \neg A_4, A_6\}$$

OLR:	$A_1 \mapsto 0$	$\{A_3, A_4\}$	$\{\neg A_2, \neg A_5\}$	$\{A_3, \neg A_4, A_5, \neg A_6\}$	$\{\neg A_2, \neg A_4, A_6\}$
DE:	$A_2 \mapsto 1$	$\{A_3, A_4\}$	$\{\neg A_5\}$	$\{A_3, \neg A_4, A_5, \neg A_6\}$	$\{\neg A_4, A_6\}$
OLR:	$A_5 \mapsto 0$	$\{A_3, A_4\}$		$\{A_3, \neg A_4, \neg A_6\}$	$\{\neg A_4, A_6\}$
DE:	$A_3 \mapsto 0$	$\{A_4\}$		$\{\neg A_4, \neg A_6\}$	$\{\neg A_4, A_6\}$
OLR:	$A_4 \mapsto 1$			$\{\neg A_6\}$	$\{A_6\}$
OLR:	$A_6 \mapsto 1$			$\{\ \}$	

$D_7 := \{A_3, \neg A_4, A_5, \neg A_6\}$ (conflict clause)

$D_6 := \{A_1, \neg A_2, A_3, \neg A_4, A_5\}$ (resolve D_7, $\{A_1, \neg A_2, \neg A_4, A_6\}$)

$D_5 := \{A_1, \neg A_2, A_3, A_5\}$ (resolve D_6, $\{A_1, A_3, A_4\}$)

$D_4 := \{A_1, \neg A_2, A_3, A_5\}$

$D_3 := \{A_1, \neg A_2, A_3\}$ (resolve D_4, $\{\neg A_2, \neg A_5\}$)

$D_2 := \{A_1, \neg A_2, A_3\}$

$D_1 := \{\neg A_2, A_3\}$ (resolve D_2, $\{\neg A_1\}$)

Backtracking to $\{A_1 \mapsto 0, A_2 \mapsto 1\}$. Unit propagation: $A_3 \mapsto 1$.