CS257: Introduction to Automated Reasoning

DPLL and CDCL

Stanford University

Plan

- DPLL
  - Abstract DPLL
- CDCL (DP Chapter 2)
  - Abstract CDCL
  - Implication graphs

* Some of the slides today are contributed by Clark Barrett, Cesare Tinelli, and Emina Torlak.

The Original DPLL Procedure

- Modern SAT solvers are based on the DPLL procedure
- DPLL tries to build incrementally a satisfying truth assignment M for a CNF formula F
- M is grown by
  - deducing the truth value of a literal from M and F, or
  - guessing a truth value
- If a wrong guess for a literal leads to an inconsistency, the procedure backtracks and tries the opposite value

DPLL as a Proof System

To facilitate a deeper look at DPLL, we present DPLL as a proof system called **Abstract DPLL**.

The procedure described next is a re-elaboration of those in [1,2].

[1] Nieuwenhuis et al, "Solving SAT and SAT Modulo Theories: from an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T).", Journal of the ACM, 53(6).
[2] Krstić and Goel, "Architecting Solvers for SAT Modulo Theories: Nelson-Oppen with DPLL.", FroCos 2007.

Abstract DPLL: A Proof System for DPLL

States:

$$\texttt{Fail} \quad \text{or} \quad \langle M, \Delta \rangle$$

where

- M is a **sequence of literals** and **decision points** $\bullet$ denoting a **partial truth assignment**
- Δ is a **set of clauses** denoting a CNF formula

Def. If $M = M_0 \bullet M_1 \bullet \cdots \bullet M_n$ where each M_i contains no decision points

- M_i is **decision level** i of M
- $M^{[i]} \stackrel{\text{def}}{=} M_0 \bullet \cdots \bullet M_i$

Abstract DPLL: A Proof System for DPLL

States:

$$\texttt{Fail} \quad \text{or} \quad \langle M, \Delta \rangle$$

Initial state:

- $\langle (), \Delta_0 \rangle$, where Δ_0 is to be checked for satisfiability

Expected final states:

- $\texttt{Fail}$ if Δ_0 is **unsatisfiable**
- $\langle M, \Delta' \rangle$ otherwise, where
  - Δ' is **equivalent** to Δ_0 and
  - M **satisfies** Δ'

Some clause terminology

Given a partial assignment: $\{p_1 \mapsto 1, p_2 \mapsto 0, p_4 \mapsto 1\}$

- $\{p_1, p_3, \neg p_4\}$ is **satisfied**
- $\{\neg p_1, p_2\}$ is **conflicting**
- $\{\neg p_1, p_3, \neg p_4\}$ is **unit**
- $\{\neg p_1, p_3, p_5\}$ is **unresolved**
- p_1 is **assigned**
- p_3 is **unassigned**

Some clause terminology

Given a partial assignment: $\{p_1 \mapsto 1, p_2 \mapsto 0, p_4 \mapsto 1\}$

- $\{p_1, p_3, \neg p_4\}$ is **satisfied**
- $\{\neg p_1, p_2\}$ is **conflicting**
- $\{\neg p_1, p_3, \neg p_4\}$ is **unit**
- $\{\neg p_1, p_3, p_5\}$ is **unresolved**
- p_1 is **assigned**
- p_3 is **unassigned**

One characteristic of DPLL-style SAT solvers is that given a partial assignment under which a clause becomes unit, it must be extended so that it satisfies the unassigned literal of this clause. Following this requirement is necessary but not sufficient for satisfying the formula.

Abstract DPLL: Proof Rules for the Original DPLL

Extending the assignment

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

Deduce the values of unassigned literals in unit clauses.
The clause $\{l_1, \cdots, l_n, l\}$ is called the **antecedent clause** of l. Denoted by `Antecedent(`l`)`.

Note: When convenient, treat M as a set

Abstract DPLL: Proof Rules for the Original DPLL

Extending the assignment

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

Deduce the values of unassigned literals in unit clauses.
The clause $\{l_1, \cdots, l_n, l\}$ is called the **antecedent clause** of l. Denoted by `Antecedent`(l).

Note: When convenient, treat M as a set

$$\frac{l \text{ literal of } \Delta \quad \neg l \text{ not literal of } \Delta \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Pure})$$

Make a pure literal true.

Abstract DPLL: Proof Rules for the Original DPLL

Extending the assignment

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

Deduce the values of unassigned literals in unit clauses.
The clause $\{l_1, \cdots, l_n, l\}$ is called the **antecedent clause** of l. Denoted by `Antecedent(`l`)`.

Note: When convenient, treat M as a set

$$\frac{l \text{ literal of } \Delta \quad \neg l \text{ not literal of } \Delta \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Pure})$$

Make a pure literal true.

$$\frac{l \in \texttt{Lits}(\Delta) \quad l, \neg l \notin M}{M := M \bullet l} \ (\textbf{Decide})$$

Guess a truth value for an unassigned literal.

Note: $\texttt{Lits}(\Delta) \overset{\text{def}}{=} \{l \mid l \text{ literal of } \Delta\} \cup \{\neg l \mid l \text{ literal of } \Delta\}$

Proof Rules for the Original DPLL

Repairing the assignment

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad M = M' \bullet l\ N \quad \bullet \notin N}{M := M'\ \neg l} \ (\textbf{Backtrack})$$

There is a conflicting clause and there is a decision point that we can backtrack to.
Backtrack to the last decision point and try the opposite value for the literal than last time.

Note: Last premise of **Backtrack** enforces chronological backtracking

Proof Rules for the Original DPLL

Repairing the assignment

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad M = M' \bullet l\ N \quad \bullet \notin N}{M := M'\ \neg l} \ (\textbf{Backtrack})$$

There is a conflicting clause and there is a decision point that we can backtrack to.
Backtrack to the last decision point and try the opposite value for the literal than last time.

Note: Last premise of **Backtrack** enforces chronological backtracking

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \quad \neg l_1, \ldots, \neg l_n \in M \quad \bullet \notin M}{\texttt{Fail}} \ (\textbf{Fail})$$

There is a conflicting clause and there are no decision points to backtrack to.
So the formula is unsatisfiable.

Proof Rules for the Original DPLL

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Pure})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \ (\textbf{Decide})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \ (\textbf{Backtrack})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}} \ (\textbf{Fail})$$

Note: In DPLL, there are no rules to update Δ, the set of clauses. Such rules are present in CDCL as we will see.

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \text{ (Propagate)}$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \text{ (Decide)}$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \text{ (Pure)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}} \text{ (Fail)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \text{ (Backtrack)}$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \text{ (Propagate)}$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \text{ (Decide)}$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \text{ (Pure)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}} \text{ (Fail)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \text{ (Backtrack)}$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet 1\ \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet 1\ \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet 1\ \neg 2\ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet 1\ \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet 1\ \neg 2\ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ \neg 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}} \ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \ (\textbf{Backtrack})$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet 1 \ \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet 1 \ \neg 2 \ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \ \neg 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack
$4 \ \neg 1 \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \ \neg 1 \ \neg 2 \ \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution example

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

note: We abbreviate p_n as n.

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet 1\ \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet 1\ \neg 2\ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ \neg 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack
$4\ \neg 1 \neg 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ \neg 1\ \neg 2\ \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
		Fail

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
	...	

How many steps (i.e., # of rule applications) does it take to derive **Fail**?

Work with your neighbor. Submit your answer at

`https://pollev.com/andreww095`

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}} \ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \ (\textbf{Backtrack})$$

DPLL execution: exercise

$$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet \neg 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet \neg 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet \neg 3\ 2\ 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet \neg 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet \neg 3\ 2\ 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l} \text{ (**Propagate**)}$$

$$\frac{l \in \mathtt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l} \text{ (**Decide**)}$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l} \text{ (**Pure**)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\mathtt{Fail}} \text{ (**Fail**)}$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l} \text{ (**Backtrack**)}$$

DPLL execution: exercise

$$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet \neg 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet \neg 3\ 2\ 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
4 3	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack
4 3 2	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
4 3 2 1	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

DPLL execution: exercise

$\Delta_0 := \{\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}\}$

M	Δ	rule
	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	
4	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Pure
$4 \bullet \neg 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Decide
$4 \bullet \neg 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4 \bullet \neg 3\ 2\ 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ 3$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Backtrack
$4\ 3\ 2$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
$4\ 3\ 2\ 1$	$\{1, \neg 2\}, \{\neg 1, \neg 2\}, \{2, 3\}, \{\neg 3, 2\}, \{1, 4\}$	Propagate
		Fail

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

$$\frac{l \in \texttt{Lits}(\Delta) \qquad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

$$\frac{l \text{ literal of } \Delta \qquad \neg l \text{ not literal of } \Delta \qquad l, \neg l \notin M}{M := M\ l}\ (\textbf{Pure})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M \qquad \bullet \notin M}{\texttt{Fail}}\ (\textbf{Fail})$$

$$\frac{\{l_1, \cdots, l_n\} \in \Delta \qquad \neg l_1, \cdots, \neg l_n \in M \qquad M = M' \bullet l\ N \qquad \bullet \notin N}{M := M'\ \neg l}\ (\textbf{Backtrack})$$

Transforming DPLL to Resolution

The search procedure of DPLL can be in fact reduced to a resolution proof (a sequence of application of resolution rules).

For details, see Chapter 4.2 of "The Correctness of SAT Solvers and Related Issues" by Lintao Zhang.

DPLL Shortcomings

OK for randomly generated CNFs, but not for practical ones. Why?

- No learning: throws away all the work performed to conclude that the current partial assignment is bad. Revisits bad partial assignments that lead to the conflict due to the same root cause.
- Chronological backtracking: backtracks one level, even if it can be deduced that the current partial assignment became doomed at a lower level.
- Naïve decisions: picks an arbitrary variable to branch on. Fails to consider the state of the search to make heuristically better decisions.

Conflict-Driven Clause Learning (CDCL)

- Learning: Δ is augmented with a conflict clause that summarizes the root cause of the conflict.

Conflict-Driven Clause Learning (CDCL)

- Learning: Δ is augmented with a conflict clause that summarizes the root cause of the conflict.
- Non-chronological backtracking: backtracks b levels, based on the cause of the conflict.

Conflict-Driven Clause Learning (CDCL)

- Learning: Δ is augmented with a conflict clause that summarizes the root cause of the conflict.
- Non-chronological backtracking: backtracks b levels, based on the cause of the conflict.
- Decision heuristics: choose the next literal to add to the current partial assignment based on the state of the search.

From DPLL to CDCL Solvers

To model conflict-driven backjumping and learning, add to states a third component C whose value is either `no` or a clause (often referred to as the conflict clause).

From DPLL to CDCL Solvers

To model conflict-driven backjumping and learning, add to states a third component C whose value is either $\texttt{no}$ or a clause (often referred to as the conflict clause).

States: $\texttt{Fail}$ or $\langle M, \Delta, C \rangle$

Initial state:

- $\langle (), \Delta_0, \texttt{no} \rangle$, where Δ_0 is to be checked for satisfiability

Expected final states:

- $\texttt{Fail}$ if Δ_0 is **unsatisfiable**
- $\langle M, G, \texttt{no} \rangle$ otherwise, where
  - G is **equivalent** to Δ_0 and
  - M **satisfies** G

From DPLL to CDCL Solvers

Replace **Backtrack** with three rules:

From DPLL to CDCL Solvers

Replace **Backtrack** with three rules:

$$\frac{C = \texttt{no} \qquad \{l_1, \cdots l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M}{C := \{l_1, \cdots, l_n\}} \; (\textbf{Conflict})$$

The conflict clause is `no`, and there is a conflicting clause w.r.t. the current partial assignment M. So we set C to the conflicting clause.

From DPLL to CDCL Solvers

Replace **Backtrack** with three rules:

$$\frac{C = \texttt{no} \qquad \{l_1, \cdots l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M}{C := \{l_1, \cdots, l_n\}} \text{ (**Conflict**)}$$

The conflict clause is `no`, and there is a conflicting clause w.r.t. the current partial assignment M. So we set C to the conflicting clause.

$$\frac{C = \{l\} \cup D \quad \{l_1, \cdots, l_n, \neg l\} \in \Delta \quad \neg l_1, \ldots \neg l_n, \neg l \in M \quad \neg l_1, \ldots, \neg l_n \prec_M \neg l}{C := \{l_1, \cdots, l_n\} \cup D} \text{ (**Explain**)}$$

Δ contains a clause $\{l_1, \cdots, l_n, \neg l\}$ such that 1) l is in the conflict clause; 2) $\neg l$ is assigned true; 3) $l_1, \cdots, l_n$ are all assigned false and are assigned before l. We can derive a new conflict clause C by applying resolution.

Note: $l \prec_M l'$ if l occurs before l' in M

From DPLL to CDCL Solvers

Replace **Backtrack** with three rules:

$$\frac{C = \texttt{no} \qquad \{l_1, \cdots l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M}{C := \{l_1, \cdots, l_n\}} \ (\textbf{Conflict})$$

The conflict clause is `no`, and there is a conflicting clause w.r.t. the current partial assignment M. So we set C to the conflicting clause.

$$\frac{C = \{l\} \cup D \quad \{l_1, \cdots, l_n, \neg l\} \in \Delta \quad \neg l_1, \ldots \neg l_n, \neg l \in M \quad \neg l_1, \ldots, \neg l_n \prec_M \neg l}{C := \{l_1, \cdots, l_n\} \cup D} \ (\textbf{Explain})$$

Δ contains a clause $\{l_1, \cdots, l_n, \neg l\}$ such that 1) l is in the conflict clause; 2) $\neg l$ is assigned true; 3) $l_1, \cdots, l_n$ are all assigned false and are assigned before l. We can derive a new conflict clause C by applying resolution.

$$\frac{C = \{l_1, \cdots, l_n, l\} \qquad \texttt{lev}(\neg l_1), \ldots, \texttt{lev}(\neg l_n) \leq \ i < \texttt{lev}(\neg l)}{C := \texttt{no} \quad M := M^{[i]} l} \ (\textbf{Backjump})$$

Compute the backtracking level: find the literals $\neg l, \neg l_n \in C$ that was assigned last and next to last. Backtrack to a level that is $< \texttt{lev}(l)$ and $\geq \texttt{lev}(l_n)$

Note: $\texttt{lev}(l) = i$ iff l occurs in decision level i of M

From DPLL to CDCL Solvers

Replace **Backtrack** with three rules:

$$\frac{C = \texttt{no} \qquad \{l_1, \cdots l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M}{C := \{l_1, \cdots, l_n\}} \text{ (**Conflict**)}$$

The conflict clause is $\texttt{no}$, and there is a conflicting clause w.r.t. the current partial assignment M. So we set C to the conflicting clause.

$$\frac{C = \{l\} \cup D \quad \{l_1, \cdots, l_n, \neg l\} \in \Delta \quad \neg l_1, \ldots \neg l_n, \neg l \in M \quad \neg l_1, \ldots, \neg l_n \prec_M \neg l}{C := \{l_1, \cdots, l_n\} \cup D} \text{ (**Explain**)}$$

Δ contains a clause $\{l_1, \cdots, l_n, \neg l\}$ such that 1) l is in the conflict clause; 2) $\neg l$ is assigned true; 3) $l_1, \cdots, l_n$ are all assigned false and are assigned before l. We can derive a new conflict clause C by applying resolution.

$$\frac{C = \{l_1, \cdots, l_n, l\} \qquad \texttt{lev}(\neg l_1), \ldots, \texttt{lev}(\neg l_n) \leq \ i < \texttt{lev}(\neg l)}{C := \texttt{no} \quad M := M^{[i]} l} \text{ (**Backjump**)}$$

Compute the backtracking level: find the literals $\neg l, \neg l_n \in C$ that was assigned last and next to last. Backtrack to a level that is $< \texttt{lev}(l)$ and $\geq \texttt{lev}(l_n)$

Maintain invariant: $\Delta \models C$ and $M \models \neg C$ when $C \neq \texttt{no}$

From DPLL to CDCL Solvers

Modify **Fail** to

From DPLL to CDCL Solvers

Modify **Fail** to

$$\frac{C \neq \texttt{no} \qquad \bullet \notin M}{\texttt{Fail}} \ (\textbf{Fail})$$

C contains a conflict clause and there are no decision points to backjump to. So the formula is unsatisfiable.

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \text{ (**Propagate**)}$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**

$$\frac{l \in \texttt{Lits}(\Delta) \quad l, \neg l \notin M}{M := M \bullet l} \ (\textbf{Decide})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
1 2 • 3	Δ	no	**Decide**
1 2 • 3 4	Δ	no	**Propagate**

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
1 2 • 3	Δ	no	**Decide**
1 2 • 3 4	Δ	no	**Propagate**
1 2 • 3 4 • 5	Δ	no	**Decide**

$$\frac{l \in \texttt{Lits}(\Delta) \quad l, \neg l \notin M}{M := M \bullet l}\ (\textbf{Decide})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l}\ (\textbf{Propagate})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
1 2 $\bullet$ 3	Δ	no	**Decide**
1 2 $\bullet$ 3 4	Δ	no	**Propagate**
1 2 $\bullet$ 3 4 $\bullet$ 5	Δ	no	**Decide**
1 2 $\bullet$ 3 4 $\bullet$ 5 $\neg$6	Δ	no	**Propagate**
1 2 $\bullet$ 3 4 $\bullet$ 5 $\neg$6 7	Δ	no	**Propagate**

$$\frac{\{l_1, \cdots, l_n, l\} \in \Delta \quad \neg l_1, \cdots, \neg l_n \in M \quad l, \neg l \notin M}{M := M\ l} \ (\textbf{Propagate})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**

$$\frac{C = \mathtt{no} \qquad \{l_1, \cdots l_n\} \in \Delta \qquad \neg l_1, \ldots, \neg l_n \in M}{C := \{l_1, \cdots, l_n\}} \ (\textbf{Conflict})$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5

$$\frac{C = \{l\} \cup D \quad \{l_1, \cdots, l_n, \neg l\} \in \Delta \quad \neg l_1, \ldots \neg l_n, \neg l \in M \quad \neg l_1, \ldots, \neg l_n \prec_M \neg l}{C := \{l_1, \cdots, l_n\} \cup D} \ (\textbf{Explain})$$

$$C = \{\neg 7\} \cup \{\neg 2, \neg 5, 6\} \quad \{\neg 1, \neg 5, 7\} \in \Delta \quad 1, 5 \prec_M 7$$

$$\Rightarrow C = \{\neg 1, \neg 5\} \cup \{\neg 2, \neg 5, 6\} = \{\neg 1, \neg 2, \neg 5, 6\}$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4

$$\frac{C = \{l\} \cup D \qquad \{l_1, \cdots, l_n, \neg l\} \in \Delta \qquad \neg l_1, \ldots \neg l_n, \neg l \in M \qquad \neg l_1, \ldots, \neg l_n \prec_M \neg l}{C := \{l_1, \cdots, l_n\} \cup D} \text{ (Explain)}$$

$$C = \{6\} \cup \{\neg 1, \neg 2, \neg 5\} \qquad \{\neg 5, \neg 6\} \in \Delta \qquad 5 \prec_M \neg 6$$
$$\Rightarrow C = \{\neg 1, \neg 2, \neg 5\} \cup \{\neg 5\} = \{\neg 1, \neg 2, \neg 5\}$$

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4
$1\ 2\ \neg 5$	Δ	no	**Backjump**

$$\frac{C = \{l_1, \cdots, l_n, l\} \qquad \mathtt{lev}(\neg l_1), \ldots, \mathtt{lev}(\neg l_n) \leq i < \mathtt{lev}(\neg l)}{C := \mathtt{no} \quad M := M^{[i]} l} \text{ (\textbf{Backjump})}$$

$\mathtt{lev}(1) = 0 \quad \mathtt{lev}(2) = 0 \quad \mathtt{lev}(5) = 2$

$\Rightarrow$ backtrack to $M^{[0]}\neg 5$

Note: could backtrack to $M^{[1]}\neg 5$ as well.

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4
$1\ 2\ \neg 5$	Δ	no	**Backjump**
$1\ 2\ \neg 5 \bullet 3$	Δ	no	**Decide**

CDCL Execution Example

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4
$1\ 2\ \neg 5$	Δ	no	**Backjump**
$1\ 2\ \neg 5 \bullet 3$	Δ	no	**Decide**
$1\ 2\ \neg 5 \bullet 3\ 4$	Δ	no	**Propagate** SAT!

From DPLL to CDCL Solvers

Also add

$$\frac{\Delta \models C \qquad C \notin \Delta}{\Delta := \Delta \cup \{C\}} \; (\textbf{Learn})$$

Learn can be applied to any clause stored in C when $C \neq \texttt{no}$.

From DPLL to CDCL Solvers

Also add

$$\frac{\Delta \models C \qquad C \notin \Delta}{\Delta := \Delta \cup \{C\}} \; (\textbf{Learn})$$

Learn can be applied to any clause stored in C when $C \neq \texttt{no}$.

$$\frac{C = \texttt{no} \qquad \Delta = \Delta' \cup \{C\} \qquad \Delta' \models C}{\Delta := \Delta'} \; (\textbf{Forget})$$

Memory can become quickly filled with millions of (conflict) clauses, so it would be nice to be able to delete clauses.

From DPLL to CDCL Solvers

Also add

$$\frac{\Delta \vDash C \qquad C \notin \Delta}{\Delta := \Delta \cup \{C\}} \; (\textbf{Learn})$$

Learn can be applied to any clause stored in C when $C \neq \texttt{no}$.

$$\frac{C = \texttt{no} \qquad \Delta = \Delta' \cup \{C\} \qquad \Delta' \vDash C}{\Delta := \Delta'} \; (\textbf{Forget})$$

Memory can become quickly filled with millions of (conflict) clauses, so it would be nice to be able to delete clauses.

$$\frac{}{M := M^{[0]} \quad C := \texttt{no}} \; (\textbf{Restart})$$

If the solver got stuck in a hopeless branch, it would be nice to be able to restart altogether. The progress is not completely lost due to Learn.

Modeling Modern SAT Solvers

At the core, current CDCL SAT solvers are implementations of the transition system with rules

Propagate, **Decide**,

Conflict, **Explain**, **Backjump**,

Learn, **Forget**, **Restart**

Modeling Modern SAT Solvers

At the core, current CDCL SAT solvers are implementations of the transition system with rules

Propagate, **Decide**,

Conflict, **Explain**, **Backjump**,

Learn, **Forget**, **Restart**

Basic CDCL $\stackrel{\text{def}}{=}$

$\{$ **Propagate**, **Decide**, **Conflict**, **Explain**, **Backjump** $\}$

Modeling Modern SAT Solvers

At the core, current CDCL SAT solvers are implementations of the transition system with rules

Propagate, **Decide**,

Conflict, **Explain**, **Backjump**,

Learn, **Forget**, **Restart**

Basic CDCL $\stackrel{\text{def}}{=}$

{ **Propagate**, **Decide**, **Conflict**, **Explain**, **Backjump** }

CDCL $\stackrel{\text{def}}{=}$ Basic CDCL + { **Learn**, **Forget**, **Restart** }

The Basic CDCL System – Correctness

Note the following terminology:

Irreducible state: state for which no Basic CDCL rules apply

Execution: sequence of transitions allowed by the rules and starting with $M = \emptyset$ and $C = \mathtt{no}$

Exhausted execution: execution ending in an irreducible state

The Basic CDCL System – Correctness

Note the following terminology:

Irreducible state: state for which no Basic CDCL rules apply

Execution: sequence of transitions allowed by the rules and starting with $M = \emptyset$ and $C = \texttt{no}$

Exhausted execution: execution ending in an irreducible state

Proposition(Strong Termination) Every execution in Basic CDCL is finite.

Note: This is not so immediate, because of **Backjump**.

The Basic CDCL System – Correctness

Note the following terminology:

Irreducible state: state for which no Basic CDCL rules apply

Execution: sequence of transitions allowed by the rules and starting with $M = \emptyset$ and $C = \texttt{no}$

Exhausted execution: execution ending in an irreducible state

Proposition(Strong Termination) Every execution in Basic CDCL is finite.

Lemma Every exhausted execution ends with either $C = \texttt{no}$ or $\texttt{Fail}$.

The Basic CDCL System – Correctness

Note the following terminology:

Irreducible state: state for which no Basic CDCL rules apply

Execution: sequence of transitions allowed by the rules and starting with $M = \emptyset$ and $C = \texttt{no}$

Exhausted execution: execution ending in an irreducible state

Proposition (Refutation Soundness) For every exhausted execution starting with $\Delta = \Delta_0$ and ending with $\texttt{Fail}$, the clause set Δ_0 is unsatisfiable.

Proposition (Solution Soundness) For every exhausted execution starting with $\Delta = \Delta_0$ and ending with $C = \texttt{no}$, the clause set Δ_0 is satisfied by M.

The CDCL System – Strategies

To ensure termination, apply 1) at least one Basic CDCL rule between each two **Learn** applications; 2) **Restart** less and less often.

The CDCL System – Strategies

To ensure termination, apply 1) at least one Basic CDCL rule between each two **Learn** applications; 2) **Restart** less and less often. A common basic strategy applies the rules with the following priorities:

1. If $n > 0$ conflicts have been found so far, increase n and apply **Restart**
2. If a clause is falsified by M, apply **Conflict**
3. Apply **Explain** repeatedly
4. Apply **Learn**
5. Apply **Backjump**
6. Apply **Propagate** to completion
7. Apply **Decide**

The CDCL System – Strategies

To ensure termination, apply 1) at least one Basic CDCL rule between each two **Learn** applications; 2) **Restart** less and less often. A common basic strategy applies the rules with the following priorities:

1. If $n > 0$ conflicts have been found so far, increase n and apply **Restart**
2. If a clause is falsified by M, apply **Conflict**
3. Apply **Explain** repeatedly
4. Apply **Learn**
5. Apply **Backjump**
6. Apply **Propagate** to completion
7. Apply **Decide**

Step 3-5 is called **conflict analysis** and there are some heuristic choices in this process.

- When to stop applying **Explain** to a conflict?
- Which level to **Backjump** to?

Conflict Analysis: Implication Graph

The goal of clause learning is to blocks partial assignments that lead to the current conflict.

A common strategy is to learn an **asserting clause**, a conflict clause that is unit after backtracking.

One way to illustrate different conflict analysis strategy is through **implication graphs**.

Conflict Analysis: Implication Graph

The goal of clause learning is to blocks partial assignments that lead to the current conflict.

A common strategy is to learn an **asserting clause**, a conflict clause that is unit after backtracking.

One way to illustrate different conflict analysis strategy is through **implication graphs**.

An implication graph is a labeled directed acyclic graph G(V, E), where:

- $v \in V$ are literals of the current partial assignment. Each node is labeled with:
  - the literal that it represents
  - the decision level at which it entered the partial assignment

Conflict Analysis: Implication Graph

The goal of clause learning is to blocks partial assignments that lead to the current conflict.

A common strategy is to learn an **asserting clause**, a conflict clause that is unit after backtracking.

One way to illustrate different conflict analysis strategy is through **implication graphs**.

An implication graph is a labeled directed acyclic graph G(V, E), where:

- $v \in V$ are literals of the current partial assignment. Each node is labeled with:
  - the literal that it represents
  - the decision level at which it entered the partial assignment
- $e \in E$ are directed labeled edges:
  - $E = \{(v_i, v_j) | v_i, v_j \in V, \neg v_i \in \texttt{Antecedent}(v_j)\}$
  - each edge (v_i, v_j) is labeled with $\texttt{Antecedent}(v_j)$.

Conflict Analysis: Implication Graph

The goal of clause learning is to blocks partial assignments that lead to the current conflict.

A common strategy is to learn an **asserting clause**, a conflict clause that is unit after backtracking.

One way to illustrate different conflict analysis strategy is through **implication graphs**.

An implication graph is a labeled directed acyclic graph G(V, E), where:

- $v \in V$ are literals of the current partial assignment. Each node is labeled with:
  - the literal that it represents
  - the decision level at which it entered the partial assignment
- $e \in E$ are directed labeled edges:
  - $E = \{(v_i, v_j) | v_i, v_j \in V, \neg v_i \in \texttt{Antecedent}(v_j)\}$
  - each edge (v_i, v_j) is labeled with $\texttt{Antecedent}(v_j)$.
- G can also contain a single conflict node labeled with $\mathcal{K}$ and incoming edges $\{(v, \mathcal{K}) | \neg v \in c\}$ labeled with c for some conflicting clause c.

  In this case, G is called a **conflict graph**.

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	`no`	

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	`no`	
1	Δ	`no`	**Propagate**

$\boxed{1@0}$

Revisiting CDCL Execution Example with Implication Graph

$$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$$

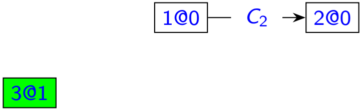
M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**

$$\boxed{1@0} \xrightarrow{C_2} \boxed{2@0}$$

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

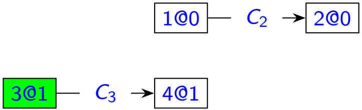
M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**



Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

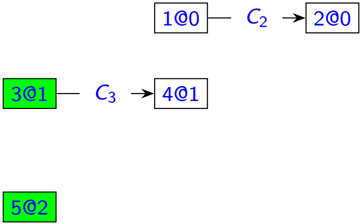
M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**



Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
1 2 $\bullet$ 3	Δ	no	**Decide**
1 2 $\bullet$ 3 4	Δ	no	**Propagate**
1 2 $\bullet$ 3 4 $\bullet$ 5	Δ	no	**Decide**



Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**

$\boxed{1@0} \xrightarrow{C_2} \boxed{2@0}$

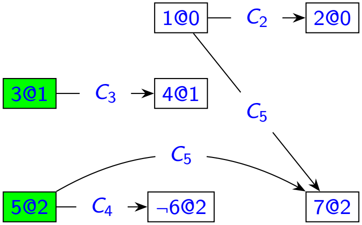
$\boxed{3@1} \xrightarrow{C_3} \boxed{4@1}$

$\boxed{5@2} \xrightarrow{C_4} \boxed{\neg 6@2}$

Revisiting CDCL Execution Example with Implication Graph

$$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$$

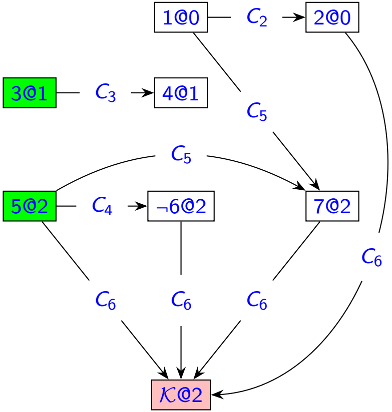
M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**



Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

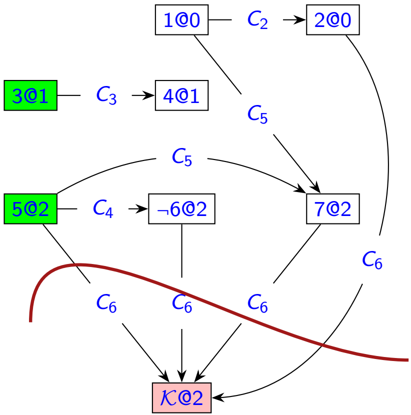
M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
1 2	Δ	no	**Propagate**
1 2 • 3	Δ	no	**Decide**
1 2 • 3 4	Δ	no	**Propagate**
1 2 • 3 4 • 5	Δ	no	**Decide**
1 2 • 3 4 • 5 ¬6	Δ	no	**Propagate**
1 2 • 3 4 • 5 ¬6 7	Δ	no	**Propagate**
1 2 • 3 4 • 5 ¬6 7	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**



Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**

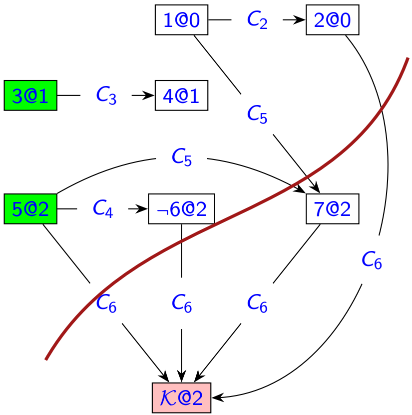


Any **separating cut** that breaks all paths from root nodes to conflict node, with roots on the reason side and conflict node on the conflict side, defines a valid conflict clause.

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5

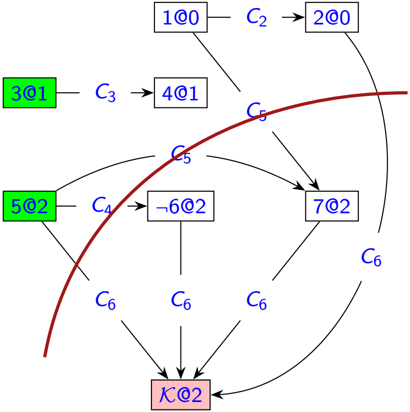


Explain can be viewed as picking a literal l in the conflict clause C, and replace C with the l-resolvant of C and $\texttt{Antecedent}(\neg l)$. In this case, we pick $l := \neg 7$.

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4

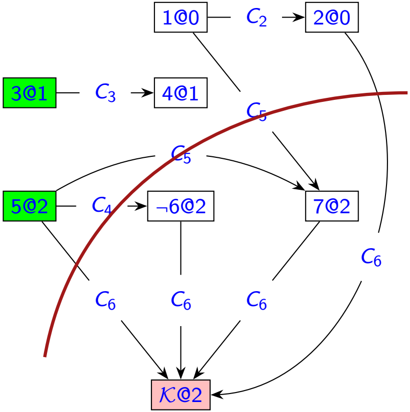


Explain can be viewed as picking a literal l in the conflict clause C, and replace C with the l-resolvant of C and $\texttt{Antecedent}(\neg l)$. In this case, we pick $l := 6$.

Revisiting CDCL Execution Example with Implication Graph

$\Delta := \{C_1 : \{1\}, C_2 : \{\neg 1, 2\}, C_3 : \{\neg 3, 4\}, C_4 : \{\neg 5, \neg 6\}, C_5 : \{\neg 1, \neg 5, 7\}, C_6 : \{\neg 2, \neg 5, 6, \neg 7\}\}$

M	Δ	C	rule
	Δ	no	
1	Δ	no	**Propagate**
$1\ 2$	Δ	no	**Propagate**
$1\ 2 \bullet 3$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5$	Δ	no	**Decide**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	no	**Propagate**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 2, \neg 5, 6, \neg 7\}$	**Conflict**
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5, 6\}$	**Explain** w. C_5
$1\ 2 \bullet 3\ 4 \bullet 5\ \neg 6\ 7$	Δ	$\{\neg 1, \neg 2, \neg 5\}$	**Explain** w. C_4
$1\ 2\ \neg 5$	Δ	no	**Backjump**



A **Unique Implication Point (UIP)** is any node other than $\mathcal{K}$ that is on all paths from the current decision node to $\mathcal{K}$.
A **first UIP** is a UIP that is closest to the conflict node.
In this case, 5@2 is the only UIP and thus also the first UIP.

Learning the First UIP

Empirical studies show that it is a good strategy to

- learn a conflict clause C such that the first UIP is the only literal at the current decision level;
- backjump to the second lowest decision level among literals in C.

To compute such conflict clause, keep applying the **Explain** rule on the last assigned literal in C, until the first UIP is the only literal at the current decision level.

Learning the First UIP

Empirical studies show that it is a good strategy to

- learn a conflict clause C such that the first UIP is the only literal at the current decision level;
- backjump to the second lowest decision level among literals in C.

To compute such conflict clause, keep applying the **Explain** rule on the last assigned literal in C, until the first UIP is the only literal at the current decision level.

The resulting conflict clause is an asserting clause.

Learning the First UIP

Empirical studies show that it is a good strategy to

- learn a conflict clause C such that the first UIP is the only literal at the current decision level;
- backjump to the second lowest decision level among literals in C.

To compute such conflict clause, keep applying the **Explain** rule on the last assigned literal in C, until the first UIP is the only literal at the current decision level.

The resulting conflict clause is an asserting clause.

Possible explanations for the results of the empirical studies:

- The strategy has a low computational cost, compared with stategies that choose UIPs further away from the conflict.
- It backtracks to the lowest decision level.

Non-chronological Backtracking is not Necessarily Better

See "Chronological Backtracking" by Nadel and Ryvchin, SAT 2018.