Conflict-Driven Clause Learning

IA085: Satisfiability and Automated Reasoning

Martin Jonáš

FI MUNI, Spring 2024

Last Time

- propositional resolution
- Davis-Putnam algorithm
- Davis-Putnam-Logemann-Loveland algorithm (DPLL)
- practical implementation of DPLL

```
def DPLL(formula Φ):
  InitializeDatastructures()

  if UnitPropagation() == CONFLICT:
      return UNSAT

  while not all variables are assigned:
      (var, polarity) ← PickUnassignedVariable()

      Decide(var, polarity)
      while UnitPropagation() == CONFLICT:
          if decisions == []:
              return UNSAT
          Backtrack()

  return SAT
```

DPLL: Not so clever

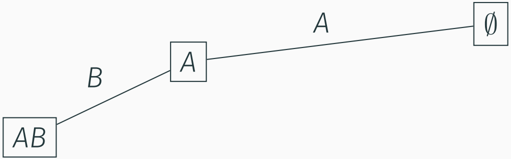
$$\varphi = \{\{\neg A, \neg B, C\}_1, \{B, \neg A, C\}_2, \{A, \neg B, C\}_3, \{\neg C, D, E\}_4, \{\neg C, D, \neg E\}_5, \{\neg C, \neg D, E\}_6, \{\neg C, \neg D, \neg E\}_7\}$$

$$\boxed{\emptyset}$$

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

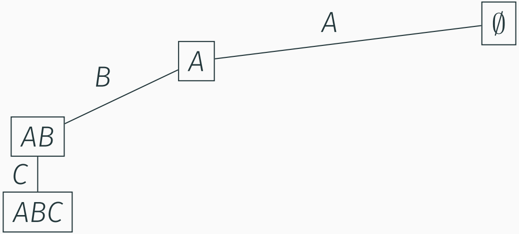
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



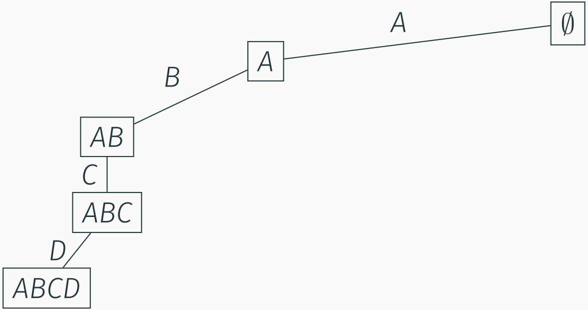
DPLL: Not so clever

$$\begin{aligned}\varphi = \ & \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}\end{aligned}$$



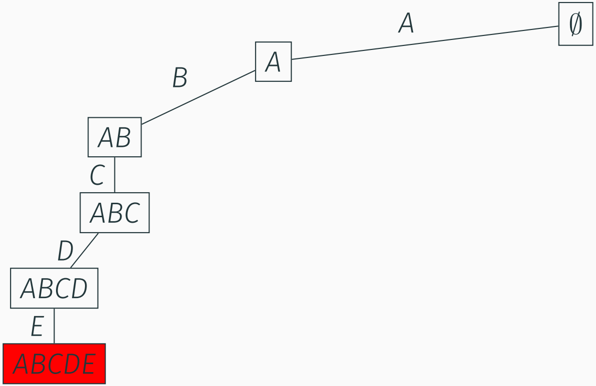
DPLL: Not so clever

$$\begin{aligned}\varphi = \;& \{\{\neg A, \neg B, C\}_1,\; \{B, \neg A, C\}_2,\; \{A, \neg B, C\}_3,\\ & \{\neg C, D, E\}_4,\; \{\neg C, D, \neg E\}_5,\; \{\neg C, \neg D, E\}_6,\; \{\neg C, \neg D, \neg E\}_7\}\end{aligned}$$



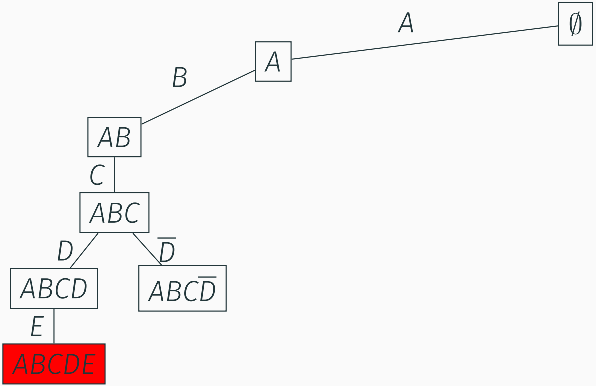
DPLL: Not so clever

$$\begin{aligned}\varphi = \; & \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \color{red}{\{\neg C, \neg D, \neg E\}_7}\}\end{aligned}$$



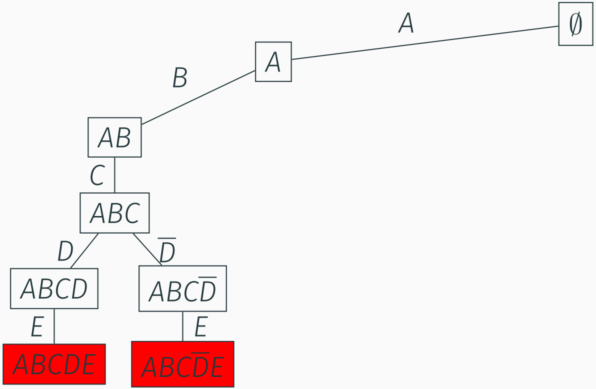
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



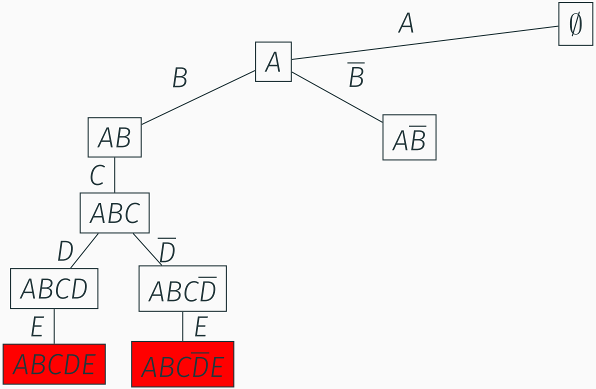
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

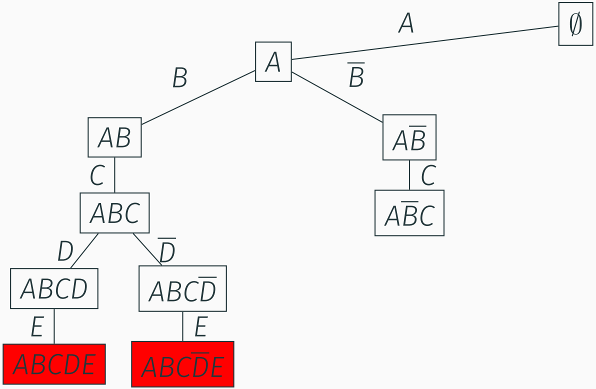


DPLL: Not so clever

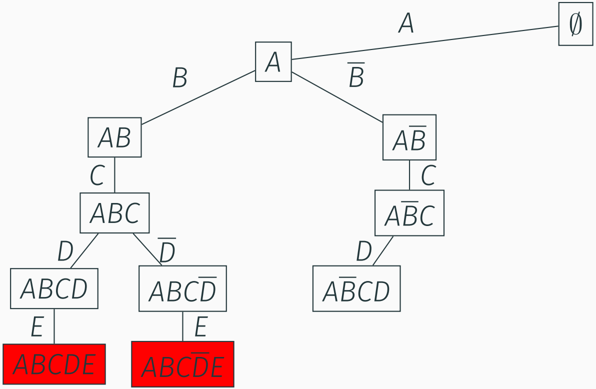
$$\varphi = \{\{\neg A, \neg B, C\}_1, \{B, \neg A, C\}_2, \{A, \neg B, C\}_3, \{\neg C, D, E\}_4, \{\neg C, D, \neg E\}_5, \{\neg C, \neg D, E\}_6, \{\neg C, \neg D, \neg E\}_7\}$$



$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

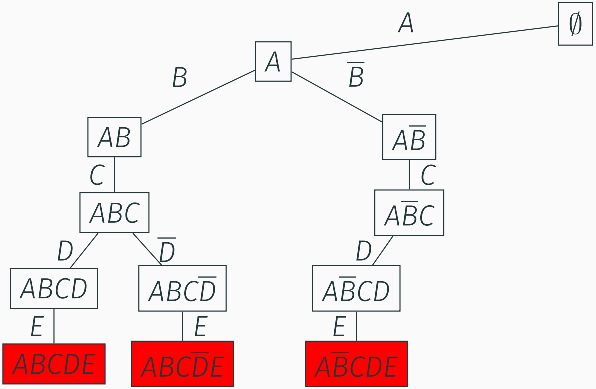


$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

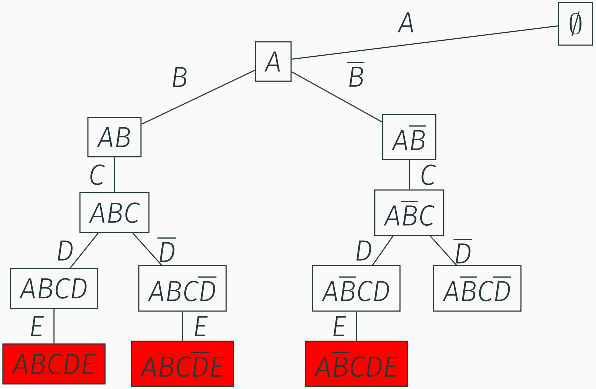


DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

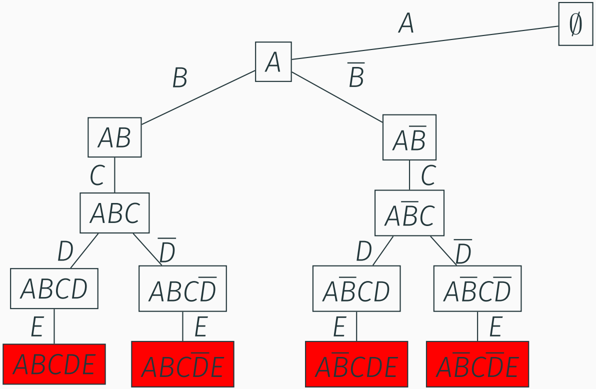


$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5, \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



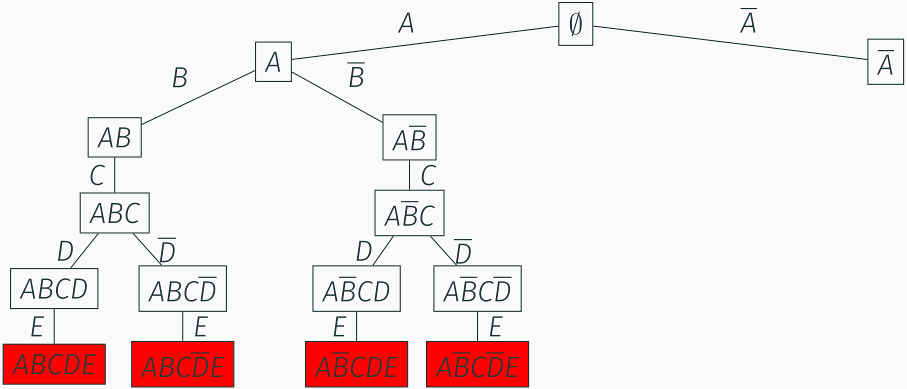
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \color{red}{\{\neg C, D, \neg E\}_5},\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



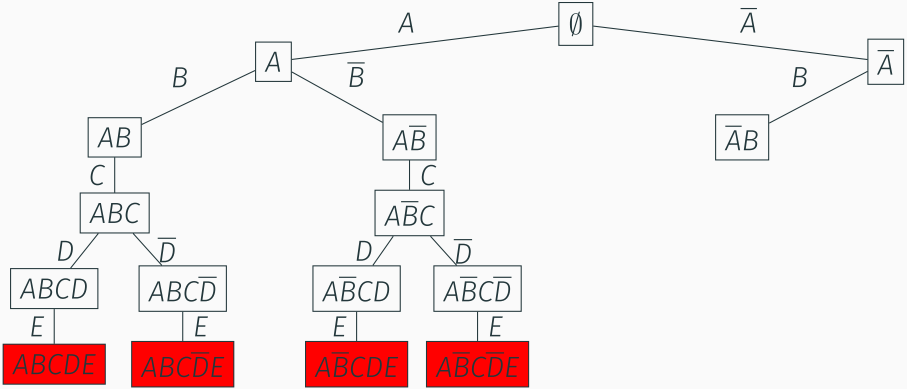
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

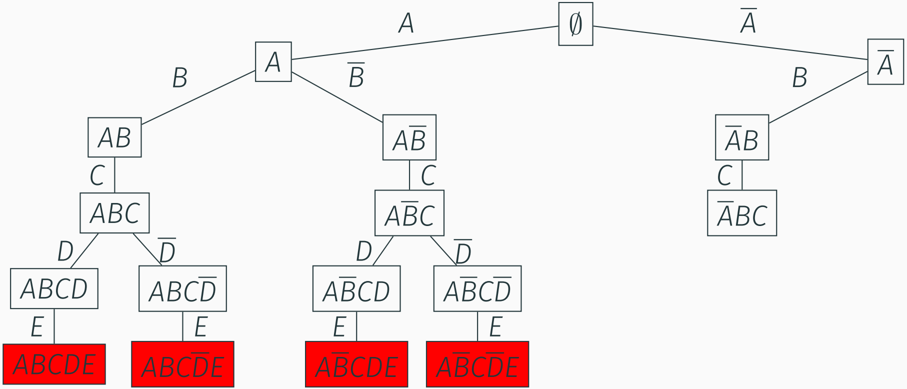


DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

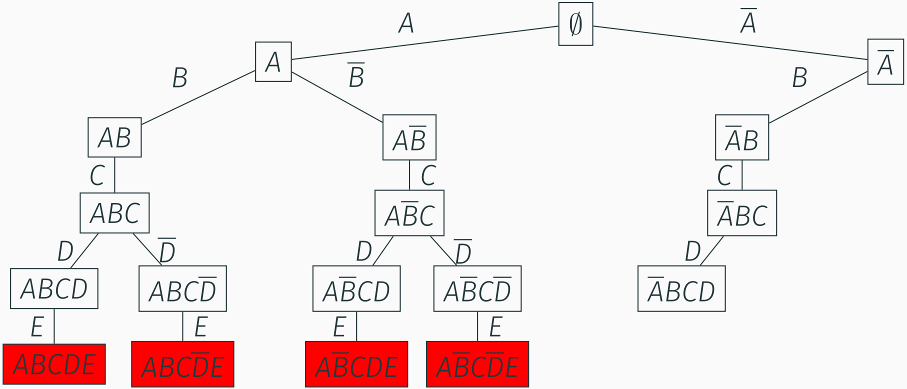


$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

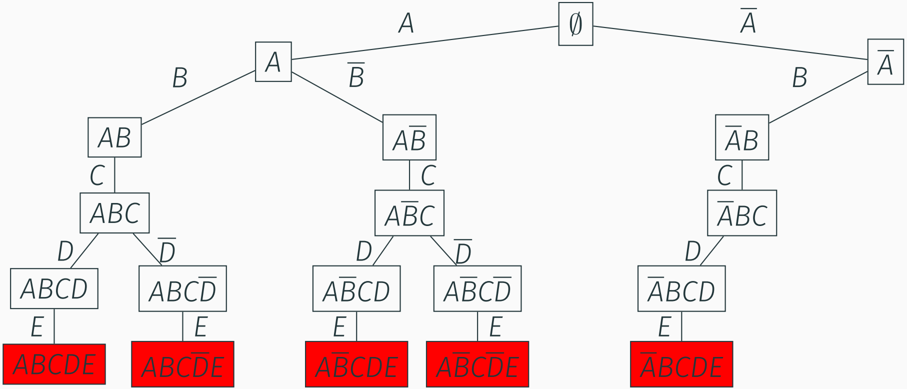


DPLL: Not so clever

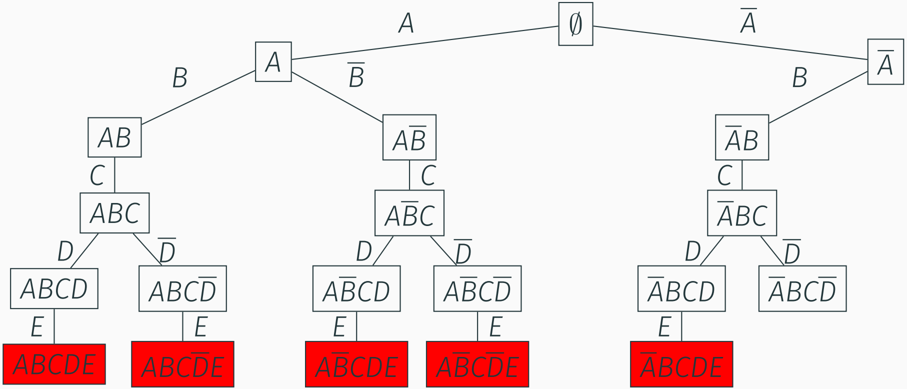
$$\varphi = \{\{\neg A, \neg B, C\}_1, \ \{B, \neg A, C\}_2, \ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4, \ \{\neg C, D, \neg E\}_5, \ \{\neg C, \neg D, E\}_6, \ \{\neg C, \neg D, \neg E\}_7\}$$



$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5, \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

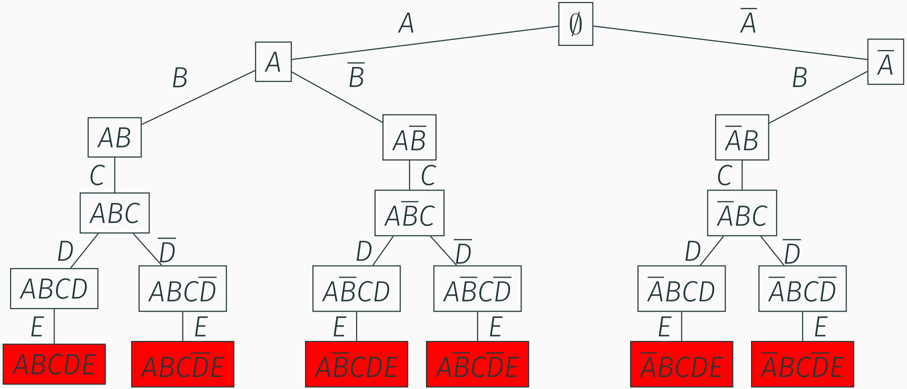


$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



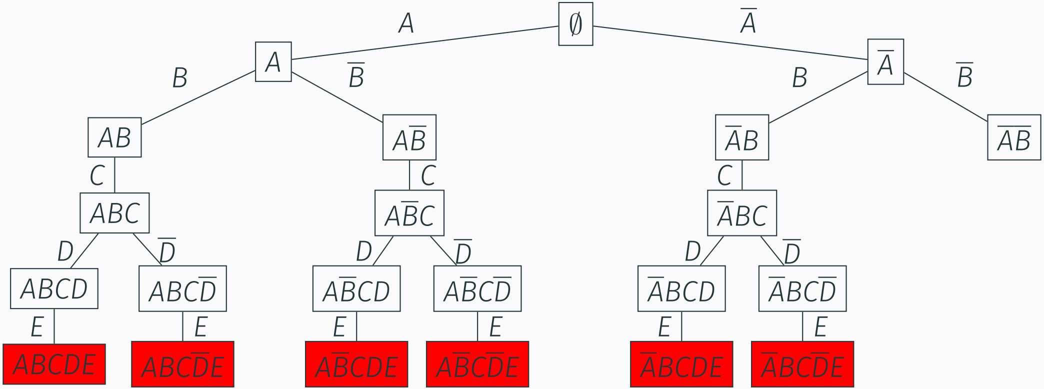
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \color{red}{\{\neg C, D, \neg E\}_5},\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



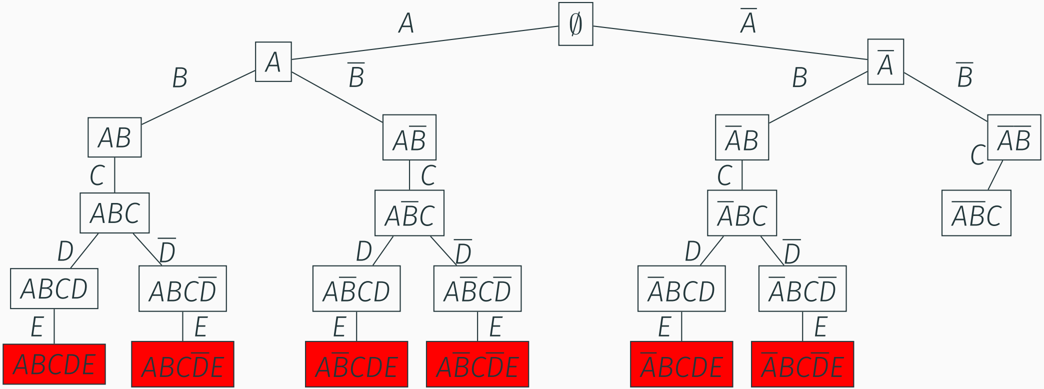
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5, \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



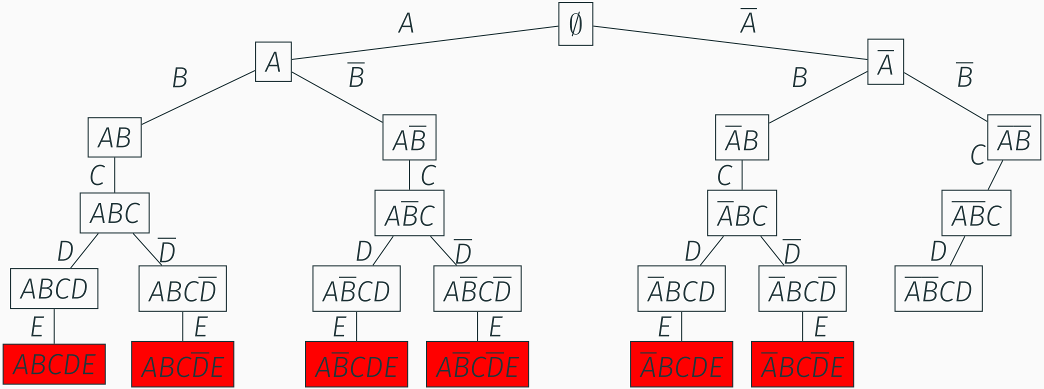
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



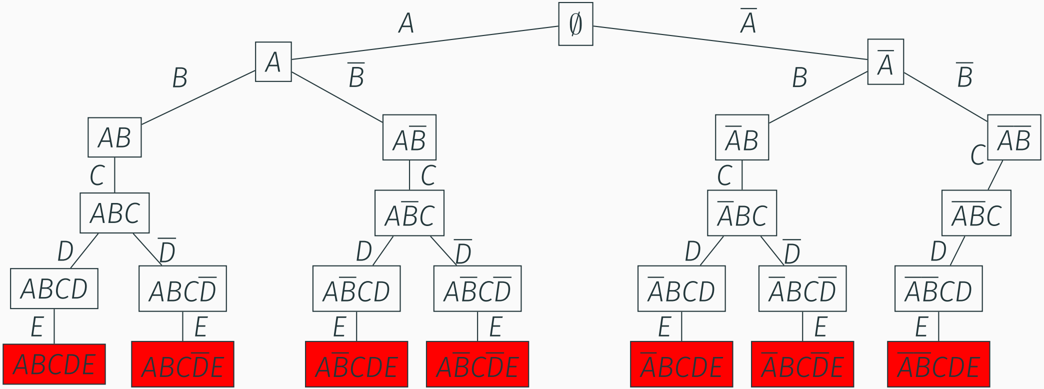
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1, \ \{B, \neg A, C\}_2, \ \{A, \neg B, C\}_3, \\ \{\neg C, D, E\}_4, \ \{\neg C, D, \neg E\}_5, \ \{\neg C, \neg D, E\}_6, \ \{\neg C, \neg D, \neg E\}_7\}$$



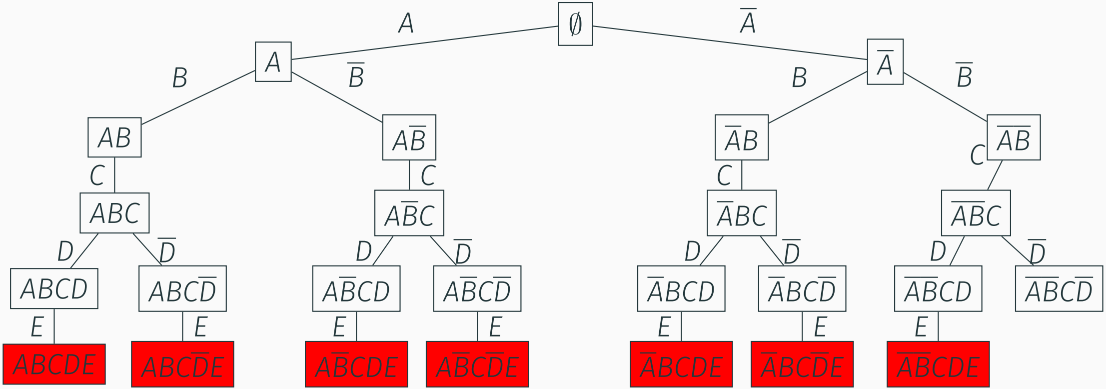
DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ {\color{red}\{\neg C, \neg D, \neg E\}_7}\}$$

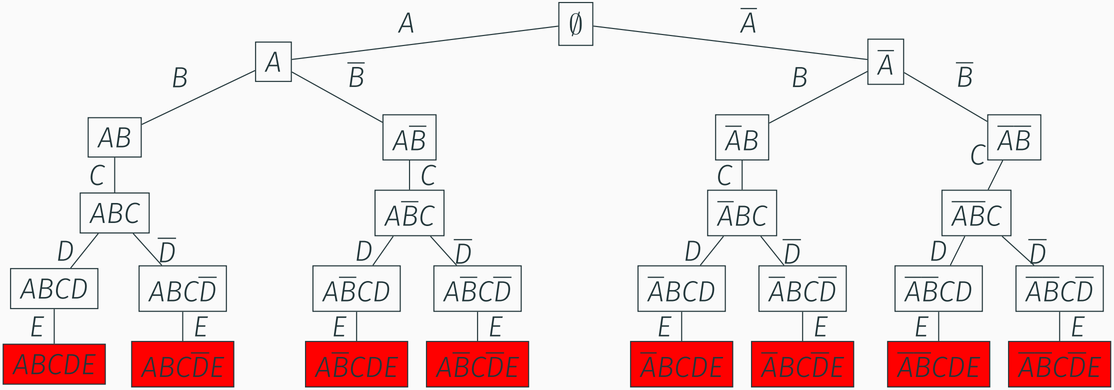


DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

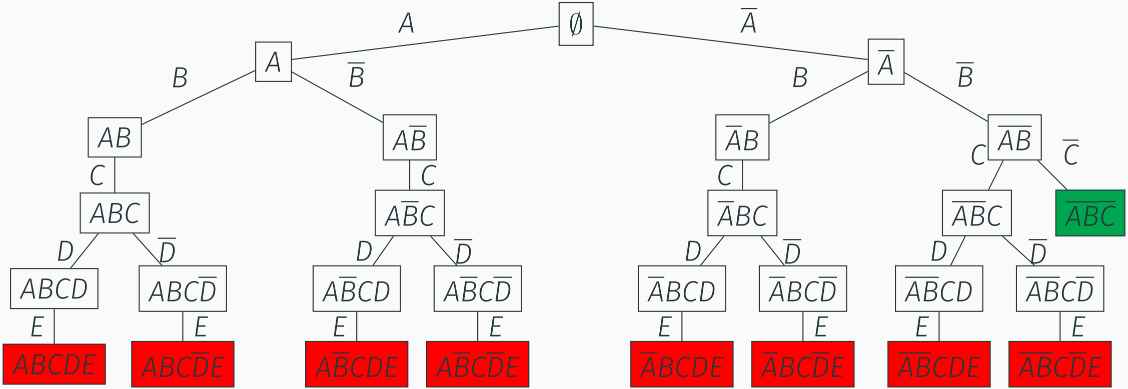


$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \color{red}{\{\neg C, D, \neg E\}_5},\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



DPLL: Not so clever

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



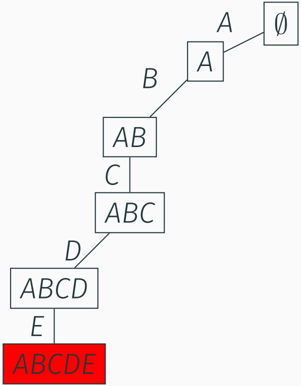
Conflict-Driven Clause Learning (CDCL)

- goal: avoid making similar mistakes multiple times
- after each conflict, perform conflict analysis
- learn a clause that generalizes the reasons for the conflict
- backtrack non-chronologically (backjumping)

Conflict Analysis

Reminder: What do we store

$$\begin{aligned}\varphi = \ & \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}\end{aligned}$$

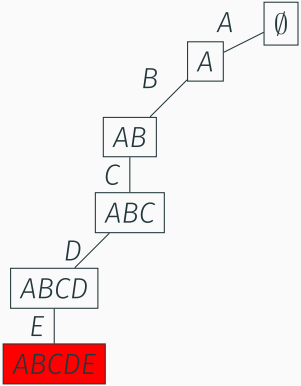


$$\text{trail} = [A, B, C, D, E]$$

$$\text{decisions} = [0, 1, 3]$$

Reminder: What do we store

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



$$\text{trail} = [A, B, C, D, E]$$
$$\text{decisions} = [0, 1, 3]$$

The decisions partition trail into decision levels

- decision literal followed by unit propagations
- level 1: $[A]$, level 2: $[B, C]$, level 3: $[D, E]$

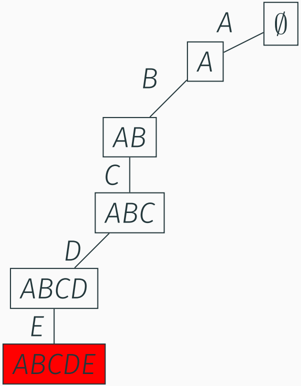
Antecedents

antecendent (reason) clause = clause that caused the unit propagation

Antecedents

antecendent (reason) clause = clause that caused the unit propagation

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$



$$\text{trail} = [A, B, C, D, E]$$
$$\text{decisions} = [0, 1, 3]$$
$$\text{reason}[A] = \texttt{undefined}$$
$$\text{reason}[B] = \texttt{undefined}$$
$$\text{reason}[C] = 1$$
$$\text{reason}[D] = \texttt{undefined}$$
$$\text{reason}[E] = 6$$

Implication graph

Representation of dependencies between currently assigned literals. Not maintained explicitly.

Vertices

- one vertex $l@d$ for each assigned literal
- one special conflict vertex κ
- vertices labeled by their decision levels ($l@d$ is literal l with decision level d)

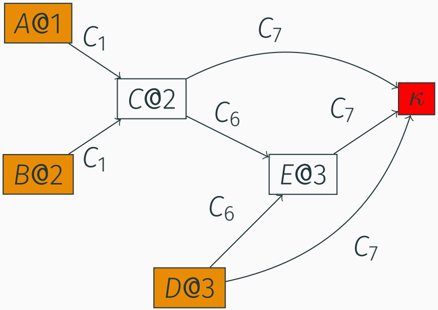
Edges

- edges are labeled by clauses
- $l \xrightarrow{C} \kappa$ if $\neg l$ is in the current conflicting clause C
- $l \xrightarrow{C_{\mathrm{reason}[r]}} r$ if r is unit propagated literal and $\neg l \in \mathrm{C}_{\mathrm{reason[r]}}$ and $\mathrm{value}[\neg l] = \texttt{false}$

Implication graph: example

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7\}$$

$$\begin{aligned}
\text{trail} &= [A, B, C, D, E] \\
\text{decisions} &= [0, 1, 3] \\
\text{reason}[A] &= \texttt{undefined} \\
\text{reason}[B] &= \texttt{undefined} \\
\text{reason}[C] &= 1 \\
\text{reason}[D] &= \texttt{undefined} \\
\text{reason}[E] &= 6
\end{aligned}$$



Clause Learning

Conflict sets

After reaching a conflict, the implication graph encodes several conflict sets = a set of literals that causes the conflict.

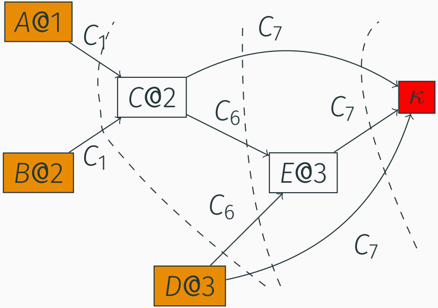
Each conflict set corresponds to a conflict clause that prohibits the conflict.

Example

- Conflict set $\{A, \neg B, C\}$ = any assignment with $\mu(A) = \top$, $\mu(B) = \bot$, $\mu(C) = \top$ causes the conflict.
- Conflict clause $\{\neg A, B, \neg C\}$

Separating cut

- cut = partition of vertices into two disjoint sets
- separating cut = decision vertices in one set, the conflict vertex is in the other
- each separating cut corresponds to a conflict set

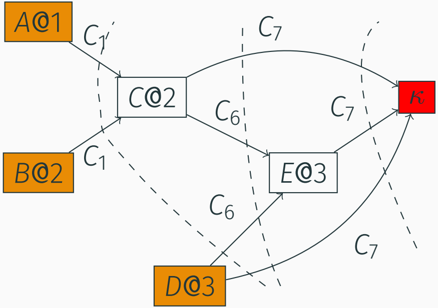


From left to right correspond to conflict sets

- $\{A, B, D\} \rightarrow$ clause $\{\neg A, \neg B, \neg D\}$
- $\{C, D\} \rightarrow$ clause $\{\neg C, \neg D\}$
- $\{C, E, D\} \rightarrow$ clause $\{\neg C, \neg E, \neg D\}$

Separating cut

- cut = partition of vertices into two disjoint sets
- separating cut = decision vertices in one set, the conflict vertex is in the other
- each separating cut corresponds to a conflict set



From left to right correspond to conflict sets

- $\{A, B, D\} \to$ clause $\{\neg A, \neg B, \neg D\}$
- $\{C, D\} \to$ clause $\{\neg C, \neg D\}$
- $\{C, E, D\} \to$ clause $\{\neg C, \neg E, \neg D\}$

Which is the best one?

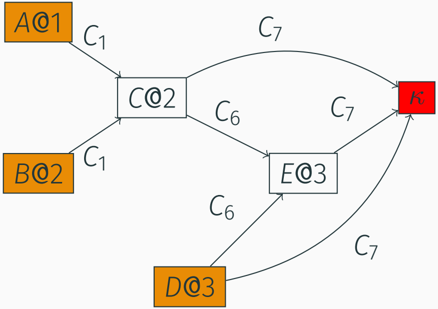
Properties of learnt clauses

The learnt conflict clauses should be

- small: prune the search space as much as possible
- asserting = contain only one literal at the current decision level

Unique Implication Point (UIP)

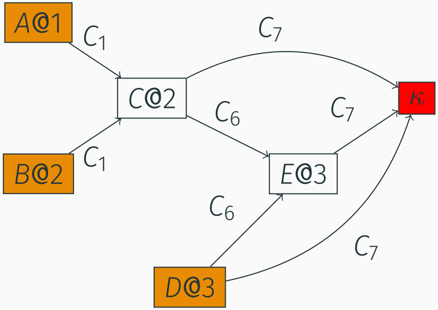
- a vertex $V \neq \kappa$ such that all paths from the current decision vertex to κ go through V
- always exists (why?)
- first UIP = closest to the conflict



Unique implication points

- D (last UIP)
- E (first UIP)

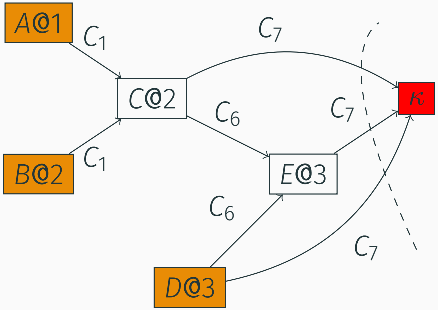
1. start with the conflicting clause
2. resolve with the reason clauses until the clause contains only one literal at the current decision level (asserting first UIP)



$\{\neg C, \neg D, \neg E\}$

Computing the conflict clause

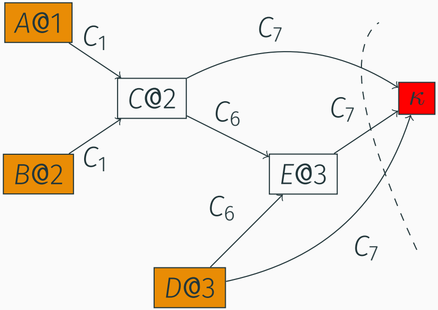
1. start with the conflicting clause
2. resolve with the reason clauses until the clause contains only one literal at the current decision level (asserting first UIP)



$\{\neg C, \neg D, \neg E\}$

Computing the conflict clause

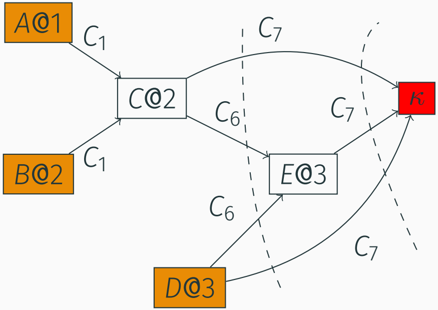
1. start with the conflicting clause
2. resolve with the reason clauses until the clause contains only one literal at the current decision level (asserting first UIP)



$$\frac{\{\neg C, \neg D, E\} \quad \{\neg C, \neg D, \neg E\}}{\{\neg C, \neg D\}}$$

Computing the conflict clause

1. start with the conflicting clause
2. resolve with the reason clauses until the clause contains only one literal at the current decision level (asserting first UIP)



$$\frac{\{\neg C, \neg D, E\} \quad \{\neg C, \neg D, \neg E\}}{\{\neg C, \neg D\}}$$

Computing the conflict clause

It is always safe to add the computed conflict clause C to the formula.

Why?

It is always safe to add the computed conflict clause C to the formula.

Why? It was derived by resolution, so $\Phi \models C$

Computing the conflict clause

```
def ComputeConflictClause(formula Φ):
    res ← current conflict clause
    if res contains only one literal from the latest decision level:
        return res

    for l in reverse(trail):
        if ¬l in C:
            res ← Resolve(var(l), res, reason[l])
        if res contains only one literal from the latest decision level:
            return res
```

For efficient implementation see `https://github.com/niklasso/minisat/blob/master/minisat/core/Solver.cc#L296` (until line 336)

Non-Chronological Backtracking

Backjumping

DPLL

- always changes the value of the last decision variable
- chronological backtracking

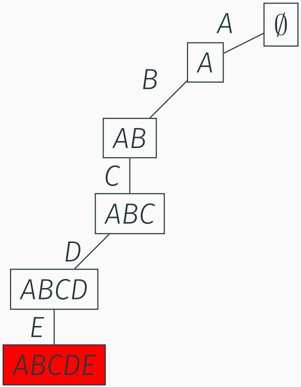
CDCL

1. learn the conflict clause
2. backtrack until the learnt clause becomes unit
3. unit propagate its asserting unit literal

CDCL can undo multiple decision levels and prune large parts of the search space
$\rightarrow$ non-chronological backtracking (or backjumping)

Complete CDCL: Example

$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,\ \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,\ \}$$



$$\text{trail} = [A, B, C, D, E]$$

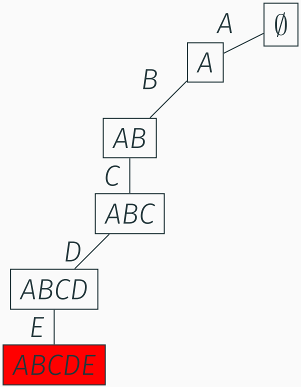
$$\text{decisions} = [0, 1, 3]$$

$$\text{reason}[C] = 1$$

$$\text{reason}[E] = 6$$

Complete CDCL: Example

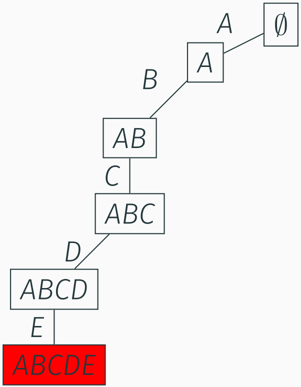
$$\varphi = \{\{\neg A, \neg B, C\}_1, \ \{B, \neg A, C\}_2, \ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4, \ \{\neg C, D, \neg E\}_5, \ \{\neg C, \neg D, E\}_6, \ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8 \qquad \}$$



$$\text{trail} = [A, B, C, D, E]$$
$$\text{decisions} = [0, 1, 3]$$
$$\text{reason}[C] = 1$$
$$\text{reason}[E] = 6$$

Complete CDCL: Example

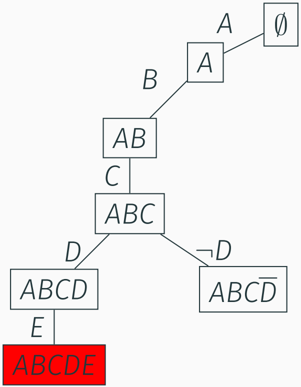
$$\begin{aligned}\varphi = \; & \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7, \\ & \{\neg C, \neg D\}_8 \qquad\qquad\qquad\qquad \}\end{aligned}$$



$$\begin{aligned}\text{trail} &= [A, B, C, D, E] \\ \text{decisions} &= [0, 1, 3] \\ \text{reason}[C] &= 1 \\ \text{reason}[E] &= 6\end{aligned}$$

Complete CDCL: Example

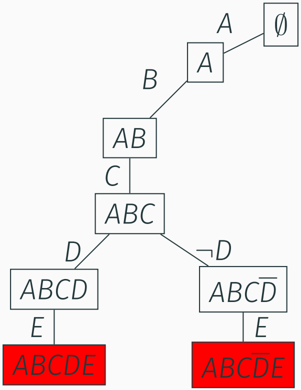
$$
\begin{aligned}
\varphi = \{ & \{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\
& \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7, \\
& \{\neg C, \neg D\}_8 \qquad\qquad\qquad\qquad \}
\end{aligned}
$$



$$
\begin{aligned}
\text{trail} &= [A, B, C, \neg D] \\
\text{decisions} &= [0, 1] \\
\text{reason}[C] &= 1 \\
\text{reason}[\neg D] &= 8
\end{aligned}
$$

Complete CDCL: Example

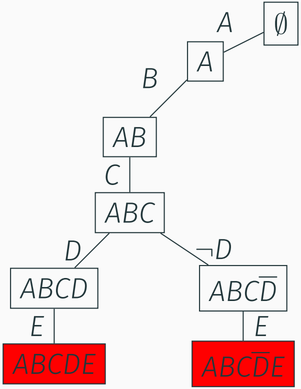
$$\begin{aligned}\varphi = \{ & \{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7, \\ & \{\neg C, \neg D\}_8 \qquad \}\end{aligned}$$



$$\begin{aligned}\text{trail} &= [A, B, C, \neg D, E] \\ \text{decisions} &= [0, 1] \\ \text{reason}[C] &= 1 \\ \text{reason}[\neg D] &= 8 \\ \text{reason}[E] &= 4\end{aligned}$$

Complete CDCL: Example

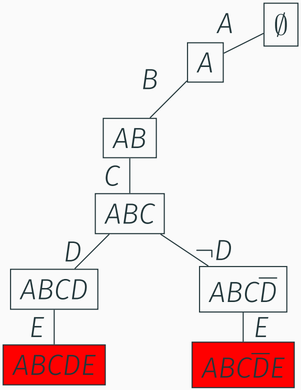
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9 \qquad \}$$



$$\text{trail} = [A, B, C, \neg D, E]$$
$$\text{decisions} = [0, 1]$$
$$\text{reason}[C] = 1$$
$$\text{reason}[\neg D] = 8$$
$$\text{reason}[E] = 4$$

Complete CDCL: Example

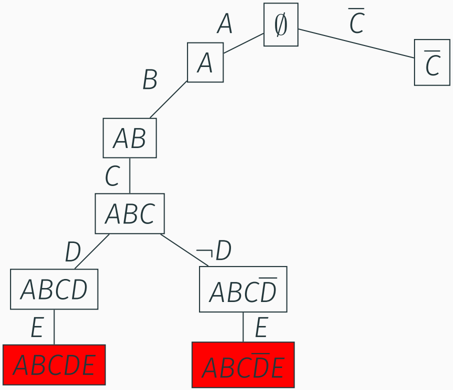
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9 \qquad\qquad \}$$



$$\begin{aligned}
\text{trail} &= [A, B, C, \neg D, E] \\
\text{decisions} &= [0, 1] \\
\text{reason}[C] &= 1 \\
\text{reason}[\neg D] &= 8 \\
\text{reason}[E] &= 4
\end{aligned}$$

Complete CDCL: Example

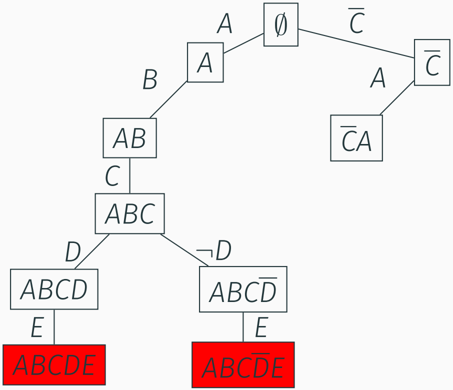
$$\begin{aligned}\varphi = \ & \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3, \\ & \{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7, \\ & \{\neg C, \neg D\}_8, \{\neg C\}_9 \qquad\qquad \}\end{aligned}$$



$$\begin{aligned}\text{trail} &= [\neg C] \\ \text{decisions} &= [] \\ \text{reason}[\neg C] &= 9\end{aligned}$$

Complete CDCL: Example

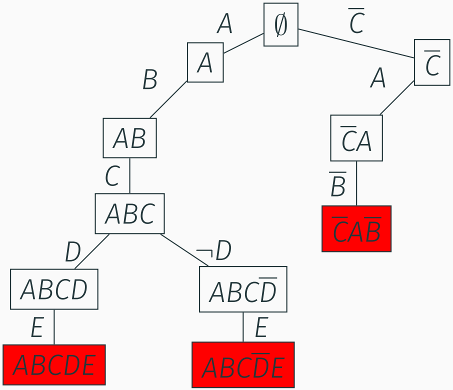
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9 \qquad \}$$



$$\text{trail} = [\neg C, A]$$
$$\text{decisions} = [1]$$
$$\text{reason}[\neg C] = 9$$

Complete CDCL: Example

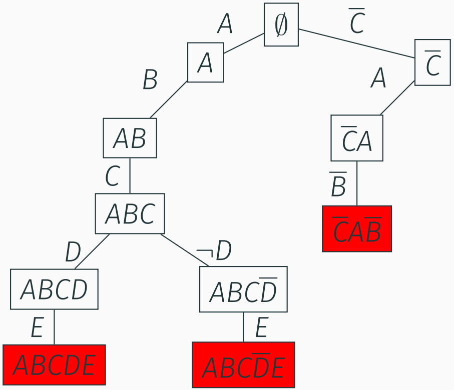
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9 \}$$



$$\text{trail} = [\neg C, A, \neg B]$$
$$\text{decisions} = [1]$$
$$\text{reason}[\neg B] = 1$$
$$\text{reason}[\neg C] = 9$$

Complete CDCL: Example

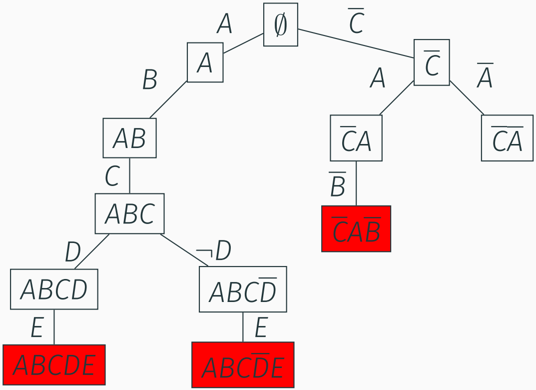
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9, \{\neg A, C\}_{10}\}$$



$$\text{trail} = [\neg C, A, \neg B]$$
$$\text{decisions} = [1]$$
$$\text{reason}[\neg B] = 1$$
$$\text{reason}[\neg C] = 9$$

Complete CDCL: Example

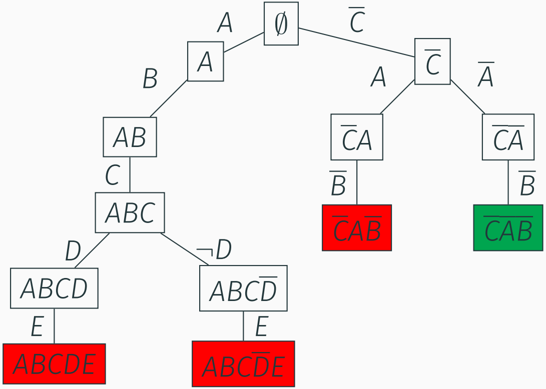
$$\varphi = \{\{\neg A, \neg B, C\}_1,\ \{B, \neg A, C\}_2,\ \{A, \neg B, C\}_3,$$
$$\{\neg C, D, E\}_4,\ \{\neg C, D, \neg E\}_5,\ \{\neg C, \neg D, E\}_6,\ \{\neg C, \neg D, \neg E\}_7,$$
$$\{\neg C, \neg D\}_8, \{\neg C\}_9, \{\neg A, C\}_{10}\}$$



$$\text{trail} = [\neg C, \neg A]$$
$$\text{decisions} = []$$
$$\text{reason}[\neg A] = 10$$
$$\text{reason}[\neg C] = 9$$

Complete CDCL: Example

$$
\begin{aligned}
\varphi = \; & \{\{\neg A, \neg B, C\}_1,\; \{B, \neg A, C\}_2,\; \{A, \neg B, C\}_3, \\
& \{\neg C, D, E\}_4,\; \{\neg C, D, \neg E\}_5,\; \{\neg C, \neg D, E\}_6,\; \{\neg C, \neg D, \neg E\}_7, \\
& \{\neg C, \neg D\}_8, \{\neg C\}_9, \{\neg A, C\}_{10}\}
\end{aligned}
$$



$$
\begin{aligned}
\text{trail} &= [\neg C, \neg A, \neg B] \\
\text{decisions} &= [] \\
\text{reason}[\neg A] &= 10 \\
\text{reason}[\neg B] &= 3 \\
\text{reason}[\neg C] &= 9
\end{aligned}
$$

```
def CDCL(formula Φ):
  InitializeDatastructures()

  if UnitPropagation() == CONFLICT:
      return UNSAT

  while not all variables are assigned:
      (var, polarity) ← PickUnassignedVariable()
      Decide(var, polarity)

      while UnitPropagation() == CONFLICT:
          (learnt, backtrackLevel) ← ConflictAnalysis()
          if (backtrackLevel == 0):
              return UNSAT
          else
              Learn(learnt)
              Backtrack(backtrackLevel, learnt)

  return SAT
```

`ConflictAnalysis()`

- analyzes the current conflict
- returns the learnt clause and the highest decision level that should be backtracked (i.e., the level whose removal makes the learnt clause unit)

`Learn(clause)`

- adds `clause` to the current formula
- initializes the watches etc.

`Backtrack(backtrackLevel, clause)`

- reverts all decisions up to the given level `backtrackLevel` (including)
- unit propagates the clause `clause`

Literal Decision Heuristics

Literal Decision Heuristics

Selecting good decision literals is crucial for performance (an idealistic perfect oracle would assign a model on the first try).

Multiple cheap literal selection heuristics (aka branching heuristics) exist

Can be based on

- current state of the solver (and the formula)
- previous computation

Literal selection often decomposed

1. select the decision variable
2. select its phase/polarity

DPLL decision heuristics

Dynamic Largest Individual Sum (DLIS)

- choose a literal that occurs most often in unsatisfied clauses
- idea: satisfy as many remaining clauses as possible

Jeroslow-Wang

- maximize $\mathit{score}(l) = \sum_{C \in \Phi, l \in C} 2^{-|C|}$
- idea: pick the literal with highest contribution to satisfying φ

MOMS

- pick the literal that occurs most often in minimal size clauses
- idea: try to satisfy the highest number of short clauses

Idea

- variables that occurred in recent conflicts are currently important

Variable State Independent Decaying Sum (VSIDS, ZCHAFF 2001)

- maintain score for each variable
- after each conflict increase score of each variable that occurred during conflict analysis by constant k
- after each 256 conflicts, divide all scores by 2 and sort the variables by score
- always choose the first unassigned variable

Idea

- decrease the scores of older variables more smoothly, not in chunks of 256 conflicts

Exponential VSIDS (EVSIDS, MiniSAT 2003)

- keep the variable sorted all the time (binary heap)
- after each conflict
  - increase score of each variable that occurred during conflict analysis by constant 1 and
  - divide scores of all other variables by a constant (e.g., 1.01)
- always choose the first unassigned variable

Idea

- decrease the scores of older variables more smoothly, not in chunks of 256 conflicts

Exponential VSIDS (EVSIDS, MINISAT 2003)

- keep the variable sorted all the time (binary heap)
- after each conflict
  - increase score of each variable that occurred during conflict analysis by constant 1 and
  - divide scores of all other variables by a constant (e.g., 1.01)
- always choose the first unassigned variable
- often also referred to as VSIDS ☹

Idea

- decrease the scores of older variables more smoothly, not in chunks of 256 conflicts

Exponential VSIDS (EVSIDS, MiniSAT 2003)

- keep the variable sorted all the time (binary heap)
- after each conflict
  - increase score of each variable that occurred during conflict analysis by constant 1 and
  - divide scores of all other variables by a constant (e.g., 1.01)
- always choose the first unassigned variable
- often also referred to as VSIDS ☹

Implementing EVSIDS

Decreasing scores of all variables often is expensive. Increase the constant that is added to the score instead.

- after each conflict, increase the variables that occurred during conflict analysis by constant k and set k to $1.01 \cdot k$

Implementing EVSIDS

Decreasing scores of all variables often is expensive. Increase the constant that is added to the score instead.

- after each conflict, increase the variables that occurred during conflict analysis by constant k and set k to $1.01 \cdot k$

The constant k can become too large ($1.01^{4459} > 2^{64}$ and $1.01^{71333} > 1.797 \cdot 10^{308}$)

- when $k > 10^{100}$ divide all scores and k by 10^{100}

Phase selection

MiniSAT

- in most of the cases assign the variable to false
- with a small probability pick a random phase

Phase saving

- choose the phase that the variable had assigned previously
- idea: the phase was likely important, focus on it unless we have a reason for flipping it

Modern SAT solvers (CaDiCaL)

- cycle between multiple modes
- e.g., phase saving $\rightarrow$ set to opposite $\rightarrow$ set to zero $\rightarrow$ set to random

Restarts

Restarts

If the solver gets ``stuck'' in the region containing no solutions, a restart of the search can help.

Restart

- clear the current assignment
- keep the learnt clauses and the variable scores and other heuristics

Usually, the solver is restarted after a certain number of conflicts.

Geometric sequence

- after $1, 2, 4, 8, 16, 32, \ldots$ conflicts (multiplied by a constant)

Luby sequence

- $1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 4, 8, 1, 1, 2, 4, 8, 16, \ldots$ conflicts (multiplied by a constant)
- used in modern SAT solvers

. . . and many others.

Restarts with phase saving

Restarts + phase saving

- the restart does not escape the current search region
- the variables are set to previous variables, but their order and dependencies are different
- explore the same region, but via a different path

Complete CDCL-based SAT solver

Complete CDCL-based SAT solver

```
def CDCL(formula Φ):
  InitializeDatastructures()

  if UnitPropagation() == CONFLICT:
      return UNSAT

  conflicts = 0;
  while not all variables are assigned:
      (var, polarity) ← PickUnassignedVariable()
      Decide(var, polarity)

      while UnitPropagation() == CONFLICT:
          ++conflicts;
          if ShouldRestart(conflicts):
              Restart()

          (learnt, backtrackLevel) ← ConflictAnalysis()
          if (backjumpLevel == 0):
              return UNSAT
          else:
              Learn(learnt)
              Backtrack(backtrackLevel, learnt)

  return SAT
```

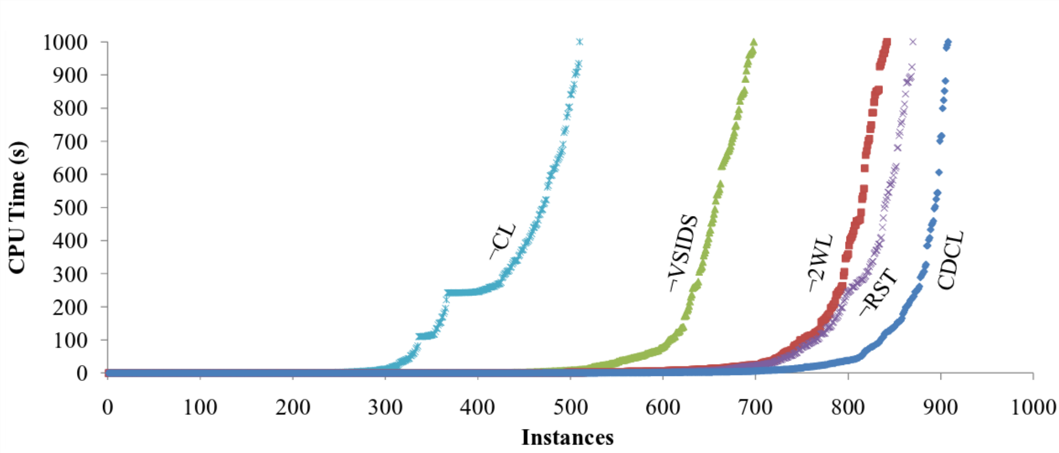
Effect of individual techniques

In 2011, Marques-Silva et. al evaluated effect of all mentioned features on the solver MiniSAT.

Tested configurations

- full MiniSAT (CDCL)
- without clause learning ($\neg$CL)
- without VSIDS ($\neg$VSIDS)
- without 2-watched literal scheme ($\neg$2WL)
- without restarts ($\neg$RST)

Effect of individual techniques



Next time

Additional features of SAT solvers

- solving under assumptions
- proof generation
- unsatisfiable core generation
- interpolation