To Link or not to Link?

Robin Coutelier

TU Wien, Vienna, Austria
robin.coutelier@tuwien.ac.at

March 13th 2025

Acknowledgements

We thank the reviewers at POS 2024 for their valuable feedback. As a reviewer pointed out, a similar approach was presented in [Bie08] and [MMZ+01]. This work is therefore not novel, but rather an independent re-discovery of an existing method.

The author acknowledges support from the ERC Consolidator Grant ARTIST 101002685; the TU Wien Doctoral College TrustACPS; the FWF SpyCoDe SFB projects F8504; the WWTF Grant ForSmart 10.47379/ICT22007.

European Research Council
Established by the European Commission

Outline

Introduction

- SAT solvers are the core of many automated reasoning tasks.
- They are used in formal verification, planning, scheduling, and many other fields.
- Mordern SAT solvers can deal with millions of variables and clauses.
- CDCL solvers are the most popular solvers.
- Solvers are built with complex data structures and algorithms.

CDCL [SS99]

```
procedure CDCL(F)
    while ⊤ do
        C ← BCP()                          ▷ Boolean Constraint Propagation
        if C = ⊤ then
            if all variables assigned then
                return SAT                 ▷ All variables are assigned
            Decide()
            continue
        if No decisions then
            return UNSAT
        Analyze(C)
        Backtrack()
```

CDCL [SS99]

```
procedure CDCL(F)
    while ⊤ do
        C ← BCP()                                  ▷ Boolean Constraint Propagation
        if C = ⊤ then
            if all variables assigned then
                return SAT                         ▷ All variables are assigned
            Decide()
            continue
        if No decisions then
            return UNSAT
        Analyze(C)
        Backtrack()
```

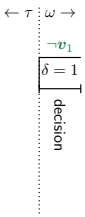
Notation and Jargon

- $\pi = \tau \cdot \omega$ is the current assignment (also called trail).
- A literal can be unassigned, satisfied or falsified (by π).
- A literal can be unassigned, propagated, or not propagated.
- A literal can be a decision or an implied literal.
- The level of a literal is the number of decisions made before it was assigned.

- A clause is unit (or propagating) if all but one of its literals are falsified (by π).
- A clause is falsified (or conflicting) if all its literals are falsified (by π).

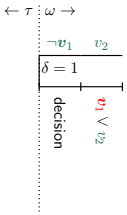
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



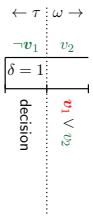
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



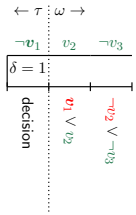
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



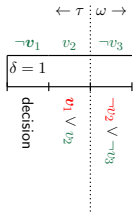
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



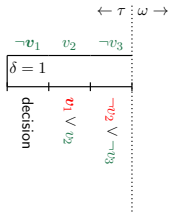
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



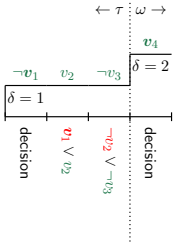
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



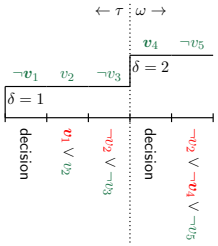
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



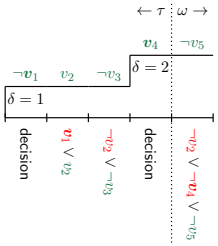
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



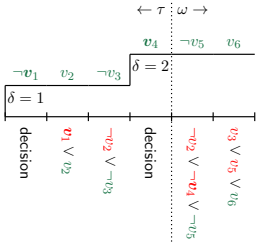
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



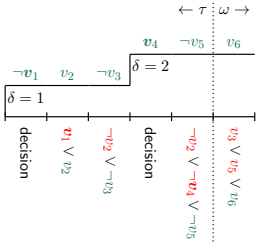
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



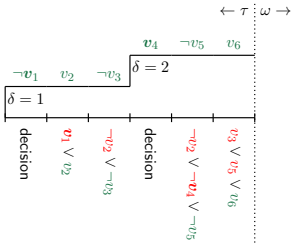
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



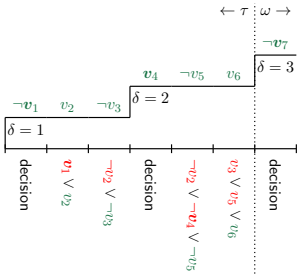
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



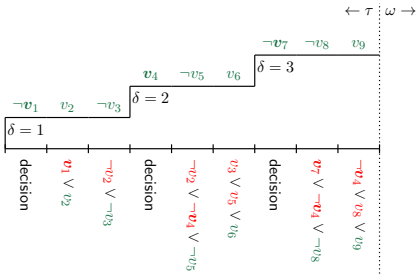
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



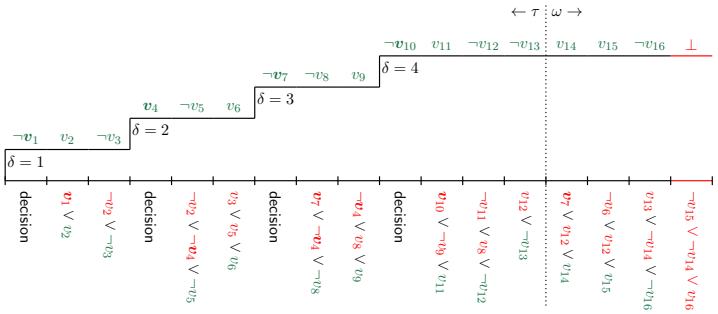
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



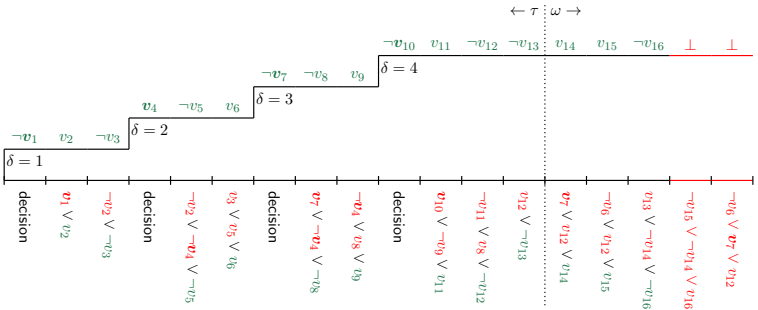
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16}
\end{aligned}
$$



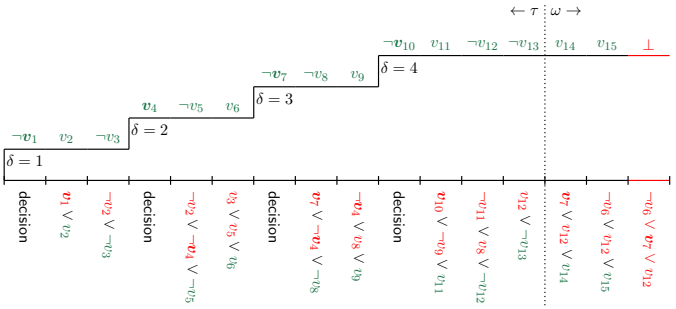
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



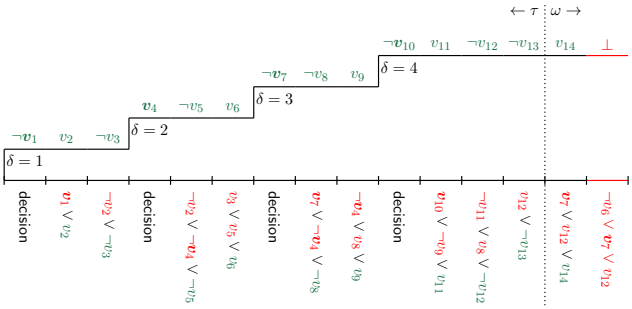
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



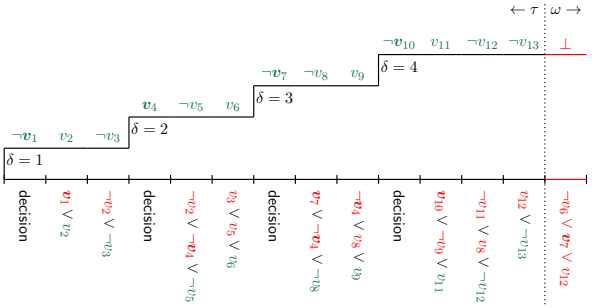
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



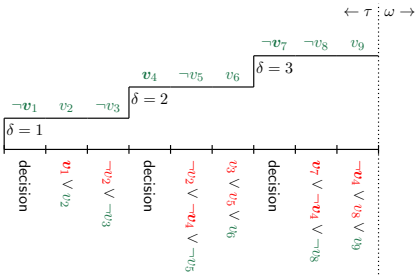
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



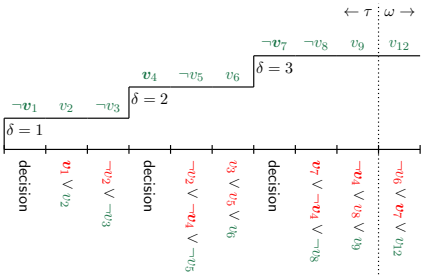
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



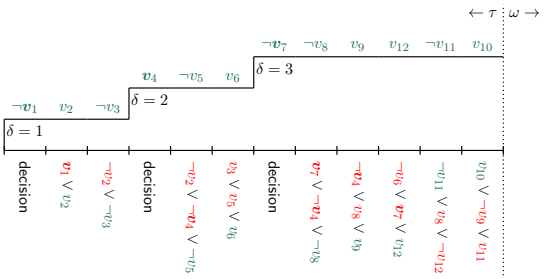
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



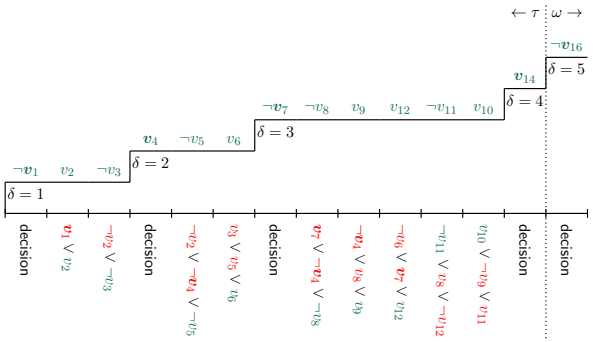
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



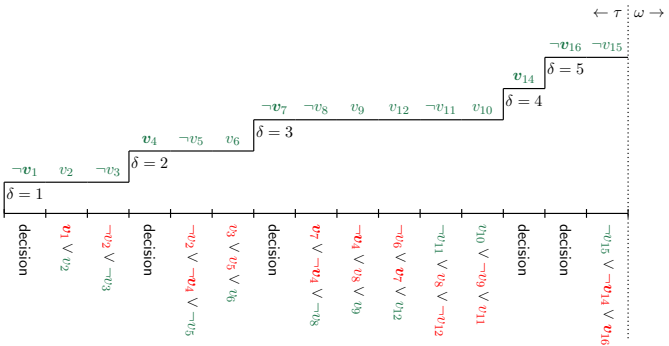
CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



CDCL at Play

$$
\begin{aligned}
C_1 &= v_1 \vee v_2 \\
C_2 &= \neg v_2 \vee \neg v_3 \\
C_3 &= \neg v_2 \vee \neg v_4 \vee \neg v_5 \\
C_4 &= v_3 \vee v_5 \vee v_6 \\
C_5 &= v_7 \vee \neg v_4 \vee \neg v_8 \\
C_6 &= \neg v_4 \vee v_8 \vee v_9 \\
C_7 &= v_{10} \vee \neg v_9 \vee v_{11} \\
C_8 &= \neg v_{11} \vee v_8 \vee \neg v_{12} \\
C_9 &= v_{12} \vee \neg v_{13} \\
C_{10} &= v_7 \vee v_{12} \vee v_{14} \\
C_{11} &= \neg v_6 \vee v_{12} \vee v_{15} \\
C_{12} &= v_{13} \vee \neg v_{14} \vee \neg v_{16} \\
C_{13} &= \neg v_{15} \vee \neg v_{14} \vee v_{16} \\
C_{14} &= \neg v_6 \vee v_7 \vee v_{12}
\end{aligned}
$$



Boolean Constraint Propagation

What is BCP?

- BCP is the process of detecting unit clauses and propagating their assignments.
- BCP also detects conflicts.
- BCP is the most time-consuming part of the SAT solver.

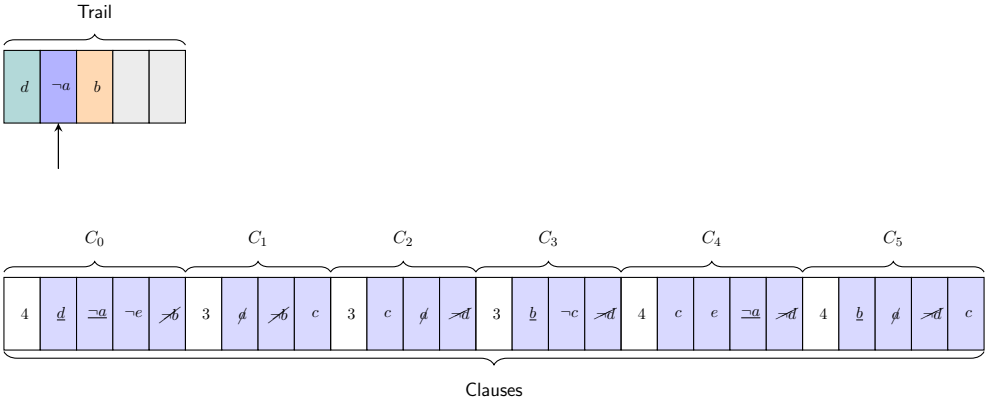
Boolean Constraint Propagation

What is BCP?

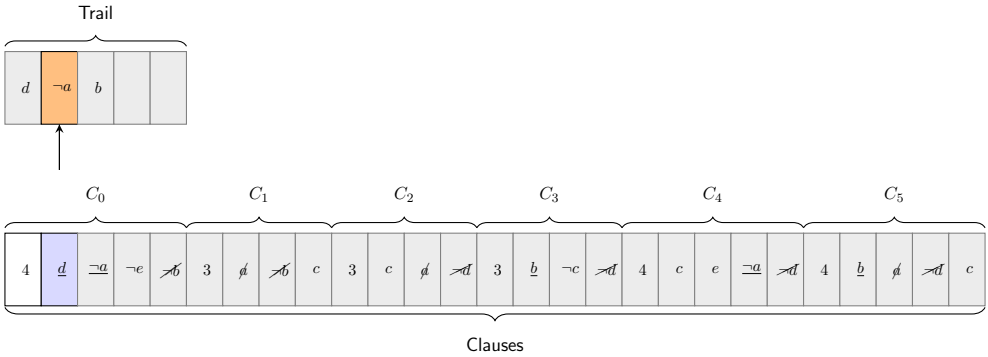
- BCP is the process of detecting unit clauses and propagating their assignments.
- BCP also detects conflicts.
- BCP is the most time-consuming part of the SAT solver.

Once a literal has been propagated, we do not want to look at it again.

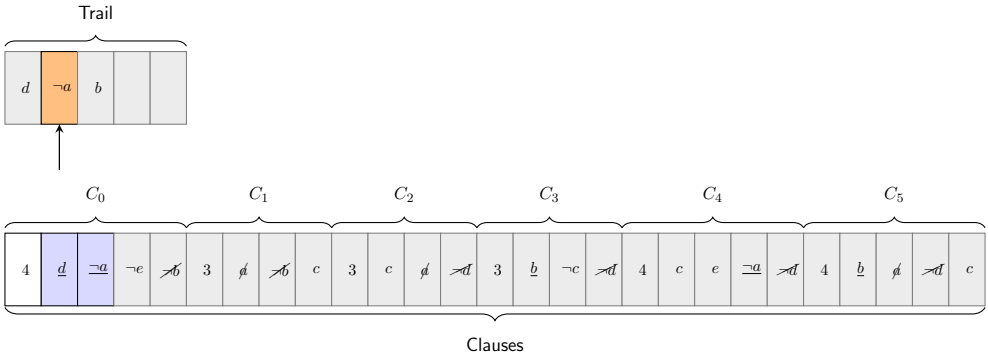
Naive Solution



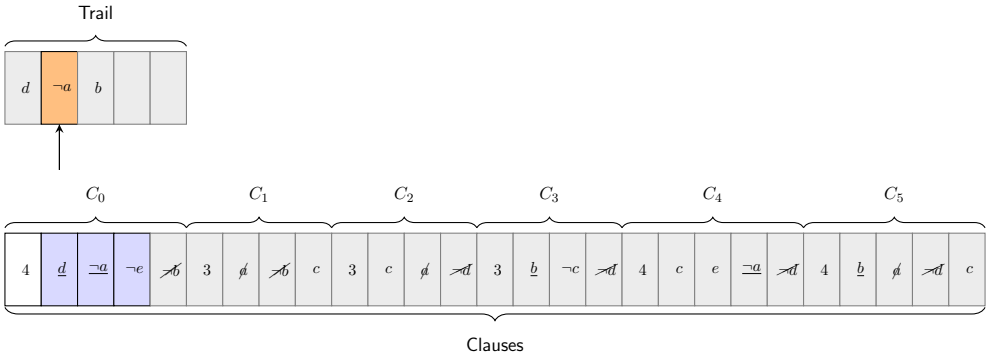
Naive Solution



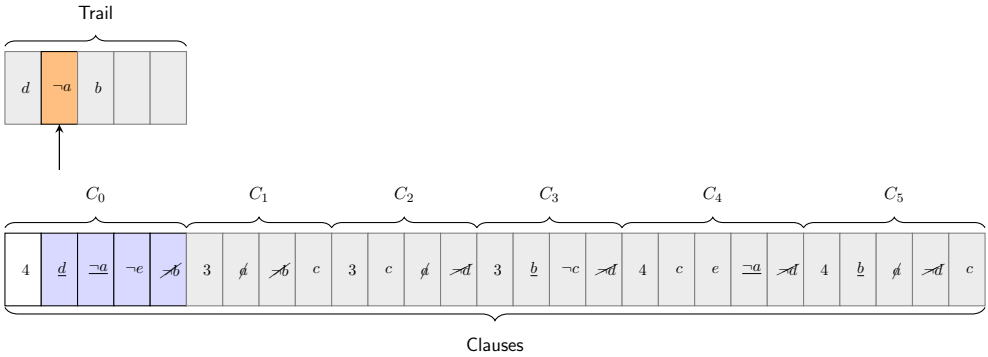
Naive Solution



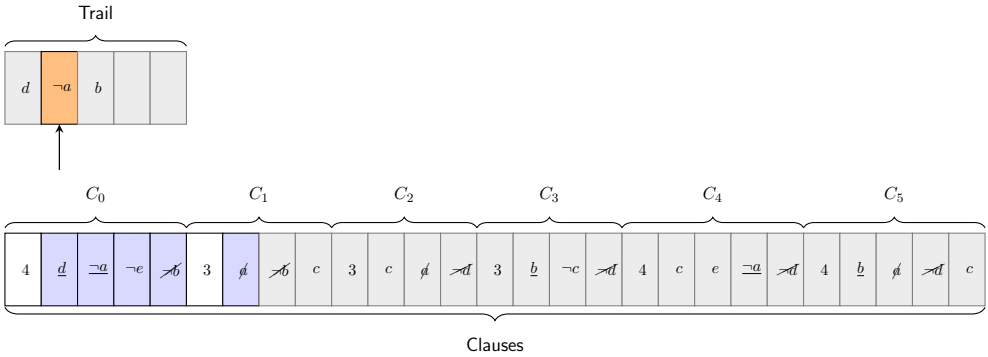
Naive Solution



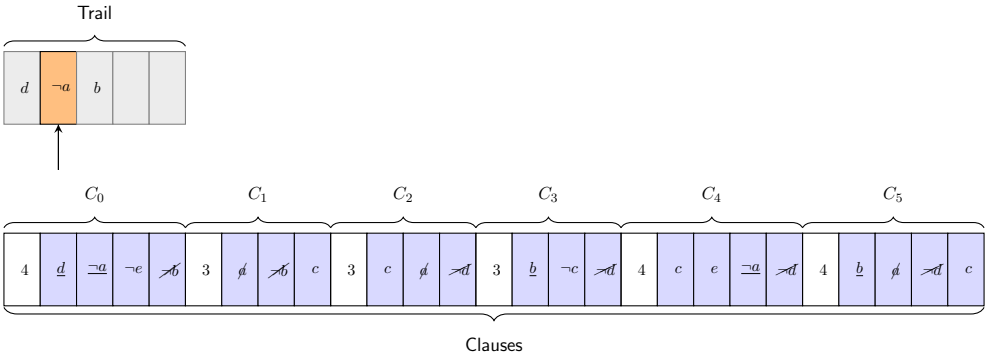
Naive Solution



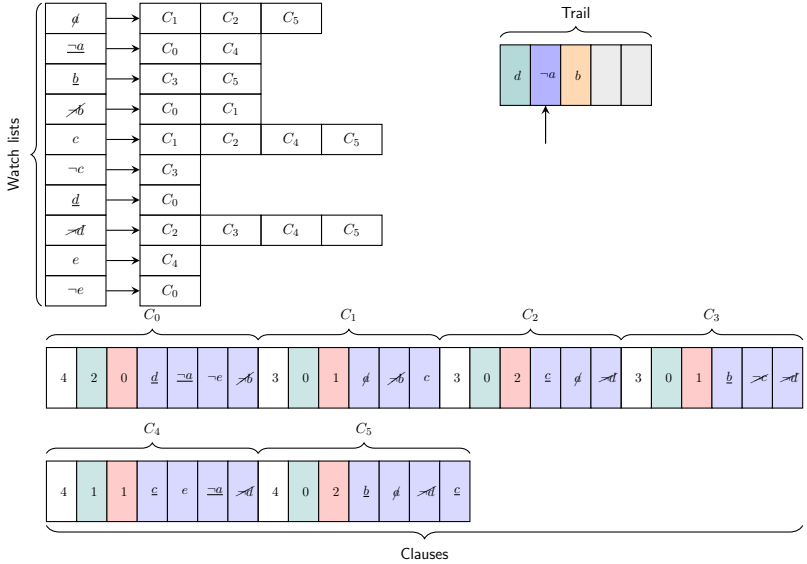
Naive Solution



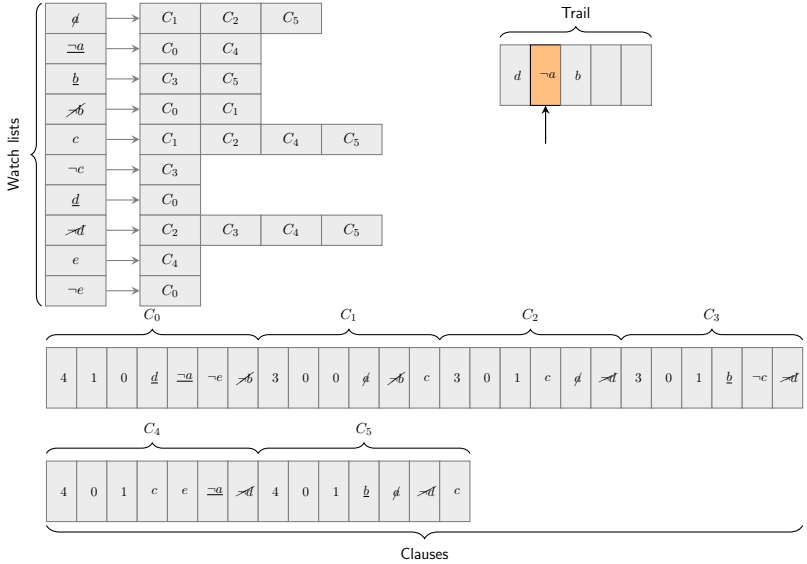
Naive Solution



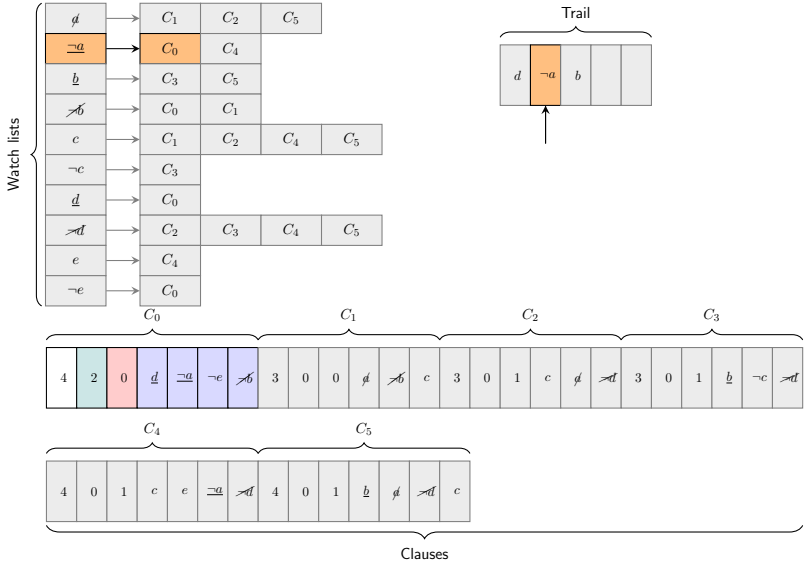
Literal Indexing



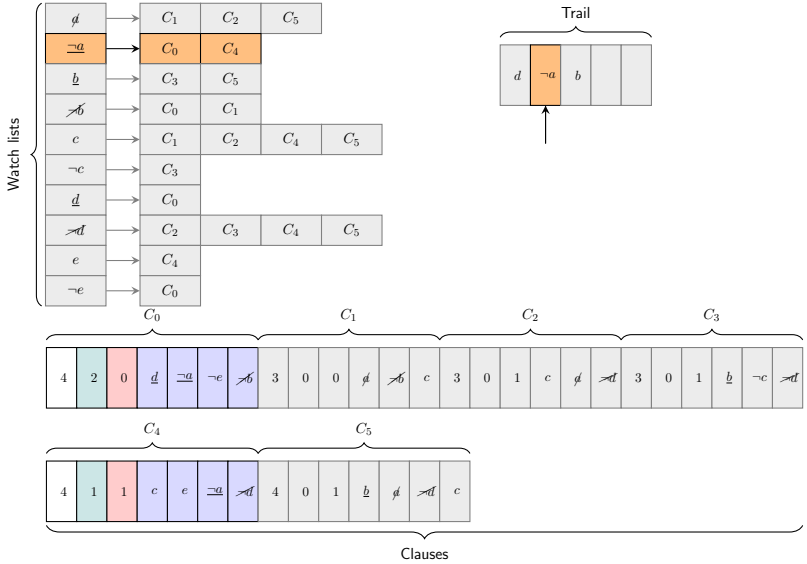
Literal Indexing



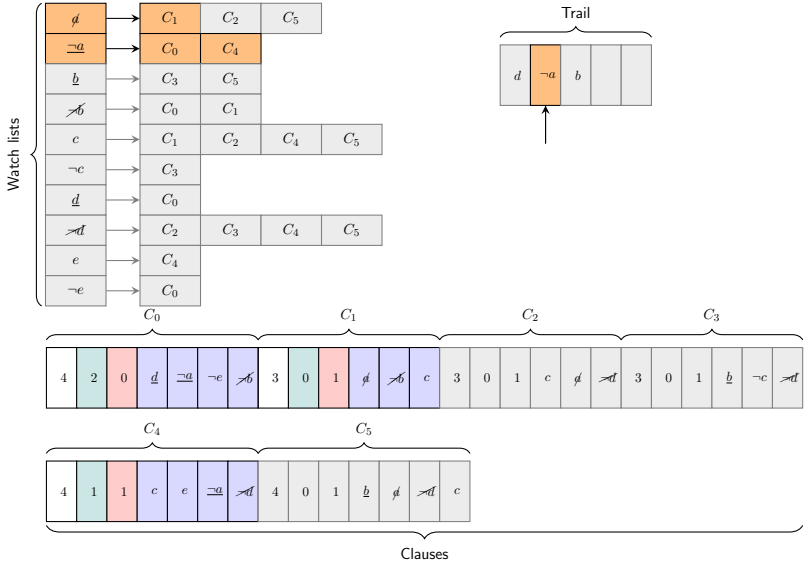
Literal Indexing



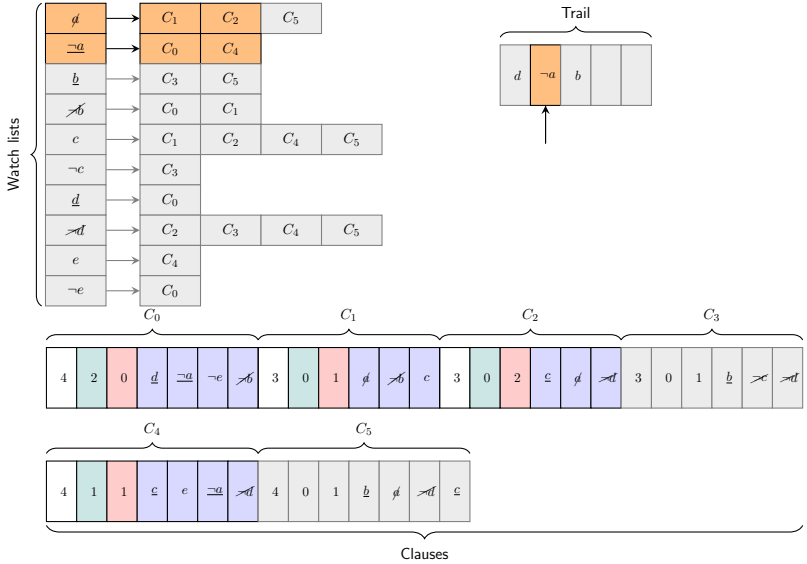
Literal Indexing



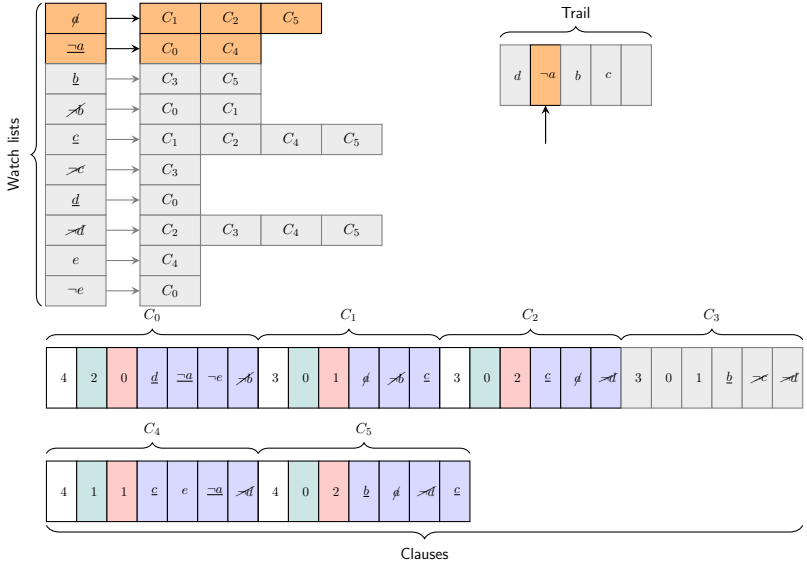
Literal Indexing



Literal Indexing



Literal Indexing



Two Watched Literals Scheme [MMZ+01]

What do we need to detect?

- When a clause becomes unit.
- When a clause is falsified.

Two Watched Literals Scheme [MMZ+01]

What do we need to detect?

- When a clause becomes unit.
- When a clause is falsified.

Are all literals important?

- As long as two literals are unassigned, the clause does not do anything.
- If the clause is satisfied, we can ignore it.

Two Watched Literals Scheme [MMZ+01]

What do we need to detect?

- When a clause becomes unit.
- When a clause is falsified.

Are all literals important?

- As long as two literals are unassigned, the clause does not do anything.
- If the clause is satisfied, we can ignore it.

What do we need?

Two literals per clause are enough. We call them the *watched literals*.

Two Watched Literals Scheme [MMZ+01]

Invariant (Watched literals)

Consider the trail $\pi = \tau \cdot \omega$. For each clause $C \in F$ watched by the two distinct watched literals c_1, c_2, we have $\neg c_1 \in \tau \Rightarrow c_2 \in \pi$.

In other words, if one of the watched literals is falsified and propagated, the other watched literal must be satisfied.

Two Watched Literals Scheme [MMZ+01]

Invariant (Watched literals)

Consider the trail $\pi = \tau \cdot \omega$. For each clause $C \in F$ watched by the two distinct watched literals c_1, c_2, we have $\neg c_1 \in \tau \Rightarrow c_2 \in \pi$.

In other words, if one of the watched literals is falsified and propagated, the other watched literal must be satisfied.

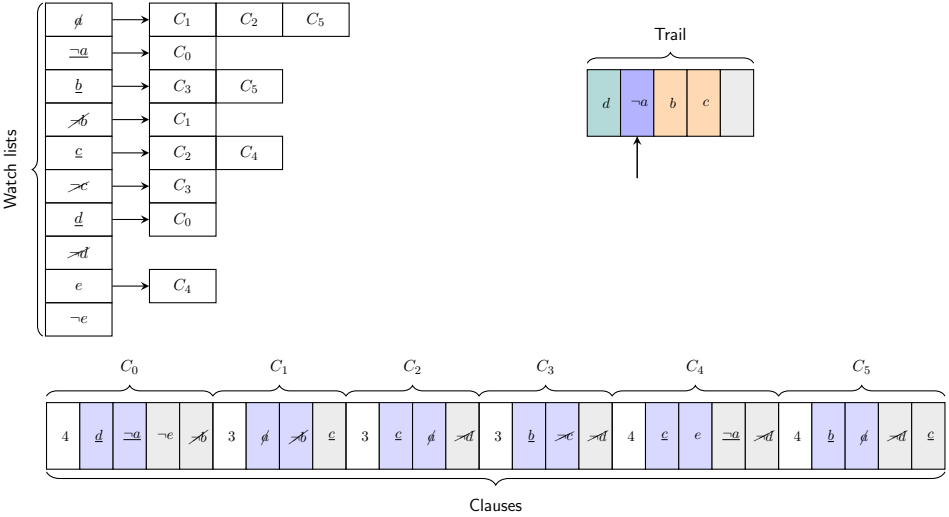
Theorem (Watched literals detect conflicts (simplified))

If the watched literals invariant holds, then no clause is falsified by propagated literals in τ.

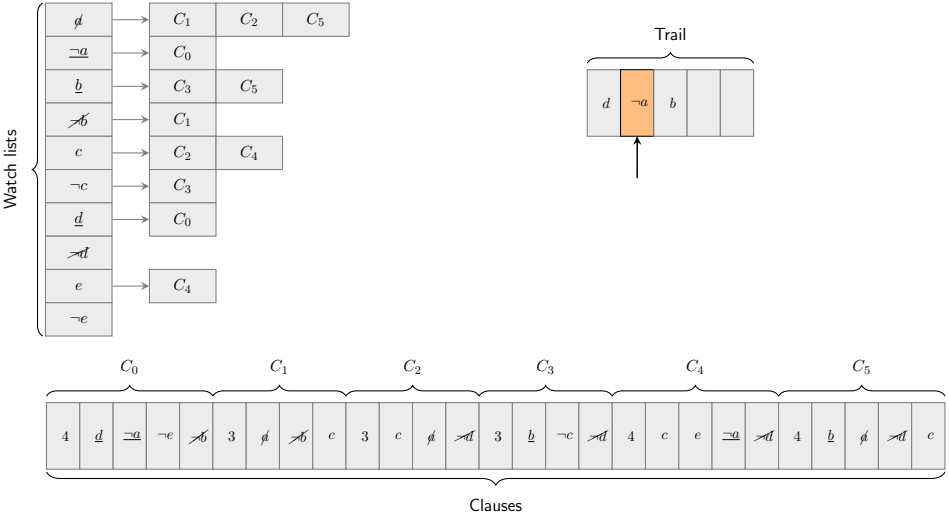
Theorem (Watched literals detect unit clauses (simplified))

If the watched literals invariant holds, then no clause is unit by propagated literals τ and not-satisfied by the partial assignment π.

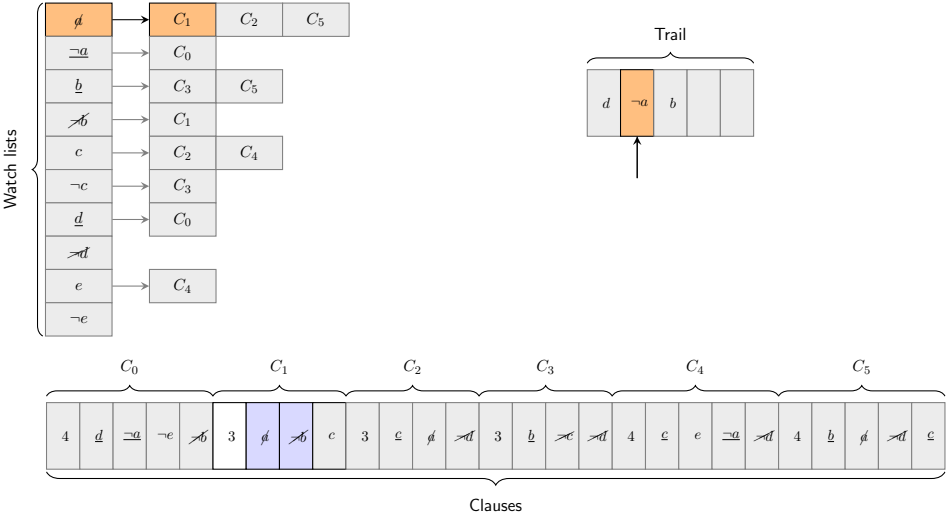
Two Watched Literals Scheme



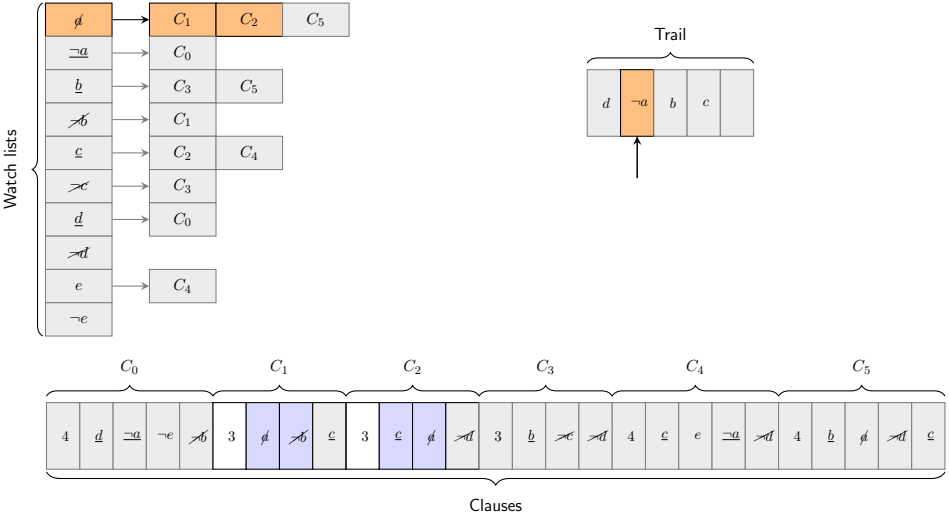
Two Watched Literals Scheme



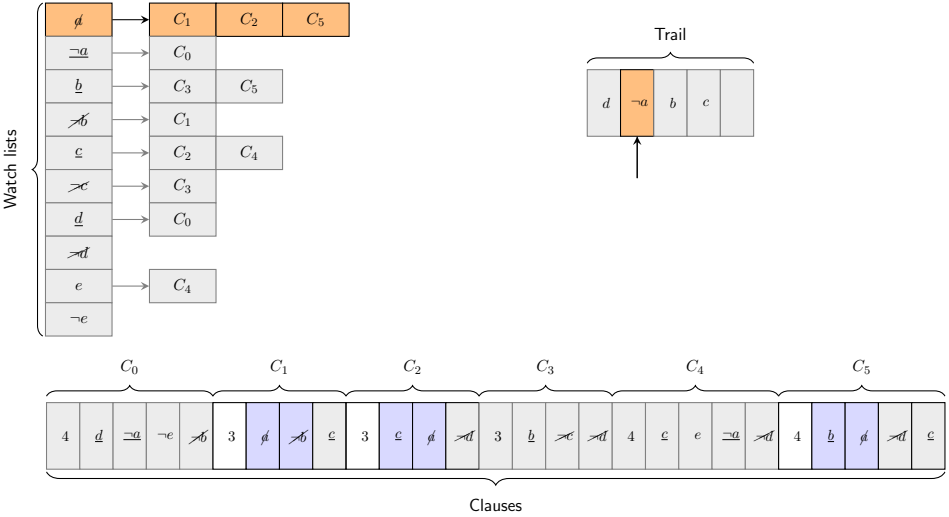
Two Watched Literals Scheme



Two Watched Literals Scheme



Two Watched Literals Scheme



Watched Literals are Great!

- Reduces number of clauses to check.
- We can use the entire assignment, not only the propagated literals.
- Backtracking becomes almost free.

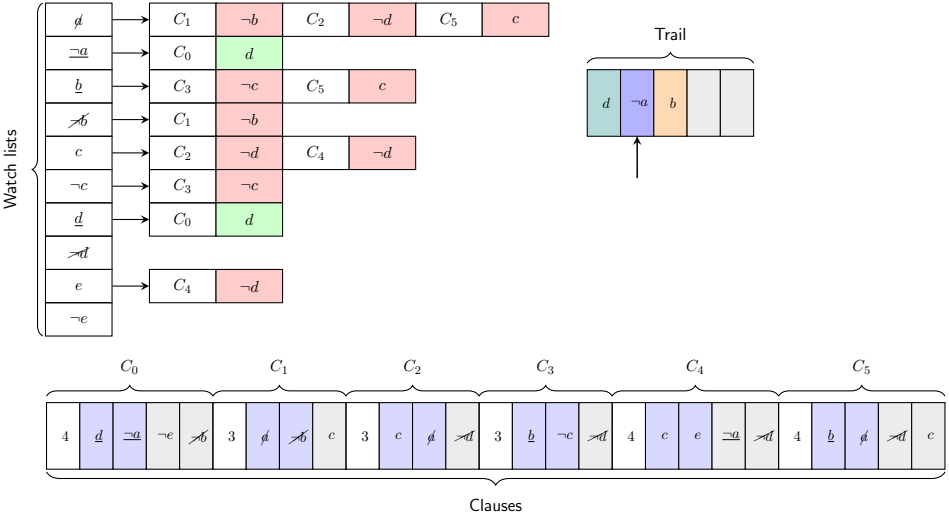
Watched Literals are Great!

- Reduces number of clauses to check.
- We can use the entire assignment, not only the propagated literals.
- Backtracking becomes almost free.

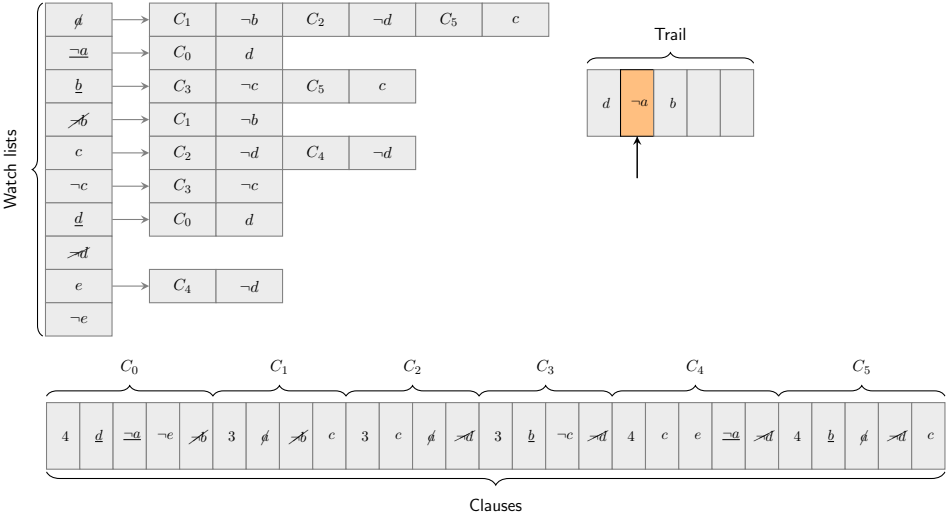
Theorem

Backtracking preserves the Watched literals invariant.

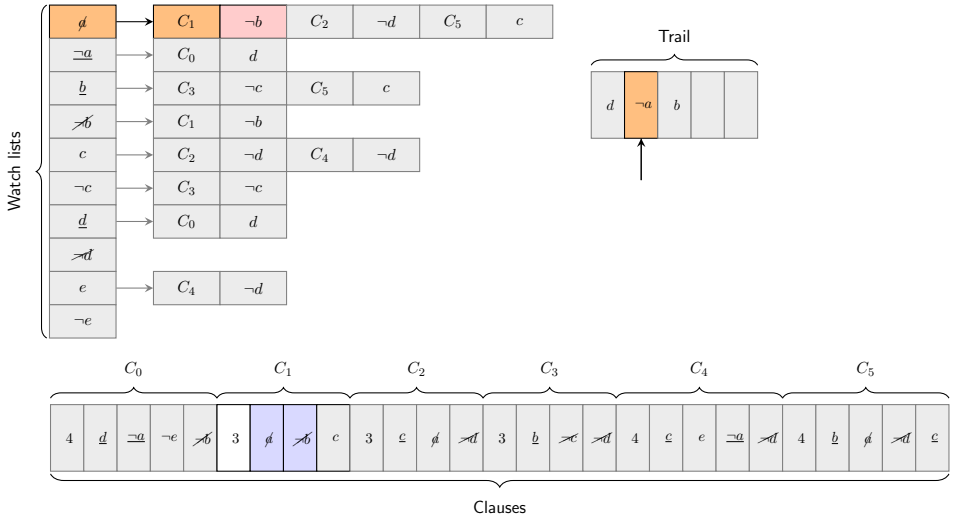
Blockers


Watch lists
$\not{a}$
$\underline{\neg a}$
$\underline{b}$
$\not{b}$
c
$\neg c$
$\underline{d}$
$\not{d}$
e
$\neg e$
C_1 $\neg b$ C_2 $\neg d$ C_5 c
C_0 d
C_3 $\neg c$ C_5 c
C_1 $\neg b$
C_2 $\neg d$ C_4 $\neg d$
C_3 $\neg c$
C_0 d
C_4 $\neg d$
Trail
d $\neg a$ b
C_0 C_1 C_2 C_3 C_4 C_5
4 $\underline{d}$ $\underline{\neg a}$ $\neg e$ $\not{b}$
3 $\not{a}$ $\not{b}$ c
3 c $\not{a}$ $\not{d}$
3 $\underline{b}$ $\neg c$ $\not{d}$
4 c e $\underline{\neg a}$ $\not{d}$
4 $\underline{b}$ $\not{a}$ $\not{d}$ c
Clauses

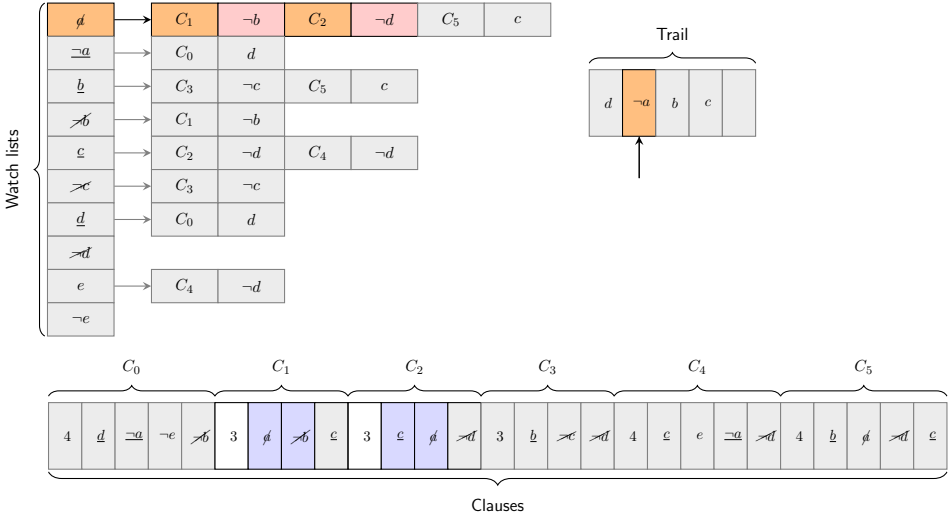
Blockers


Watch lists
$\not{a}$
$\underline{\neg a}$
$\underline{b}$
$\not{b}$
c
$\neg c$
$\underline{d}$
$\not{d}$
e
$\neg e$
C_1 $\neg b$ C_2 $\neg d$ C_5 c
C_0 d
C_3 $\neg c$ C_5 c
C_1 $\neg b$
C_2 $\neg d$ C_4 $\neg d$
C_3 $\neg c$
C_0 d
C_4 $\neg d$
Trail
d $\neg a$ b
C_0 C_1 C_2 C_3 C_4 C_5
4 $\underline{d}$ $\underline{\neg a}$ $\neg e$ $\not{b}$ 3 $\not{a}$ $\not{b}$ c 3 c $\not{a}$ $\not{d}$ 3 $\underline{b}$ $\neg c$ $\not{d}$ 4 c e $\underline{\neg a}$ $\not{d}$ 4 $\underline{b}$ $\not{a}$ $\not{d}$ c
Clauses

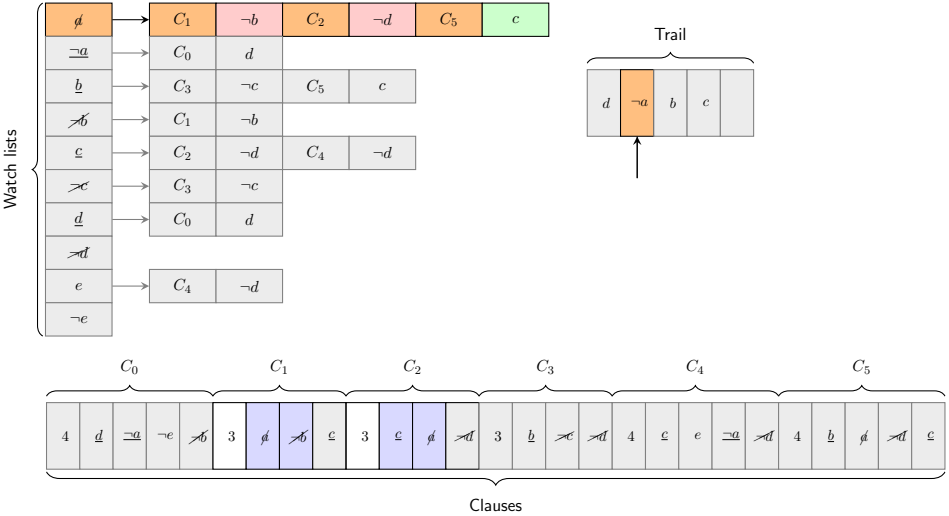
Blockers



Blockers



Blockers



Blocker properties

Theorem

Let C be a clause watched by two literals c_1, c_2 and let b be the blocker literal. In Non-Chronological Backtracking CDCL, if the blocker b is satisfied when $\neg c_1$ is propagated, then b will remain satisfied at least until c_1 is unassigned.

Blocker properties

Theorem

Let C be a clause watched by two literals c_1, c_2 and let b be the blocker literal. In Non-Chronological Backtracking CDCL, if the blocker b is satisfied when $\neg c_1$ is propagated, then b will remain satisfied at least until c_1 is unassigned.

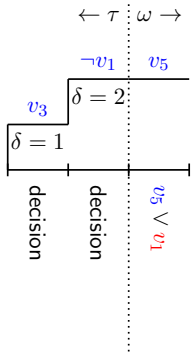
Invariant (Bocked watched literals)

For each clause $C \in F$ watched by the two distinct literals c_1, c_2 and with blocker b, we have

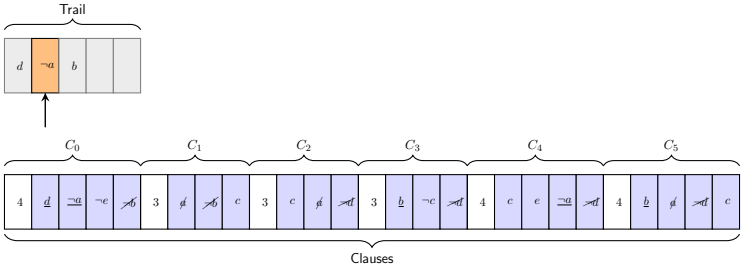
$$\neg c_1 \in \tau \Rightarrow \Big(c_2 \in \pi \vee b \in \pi \Big)$$

Blocker Invariant Illustration

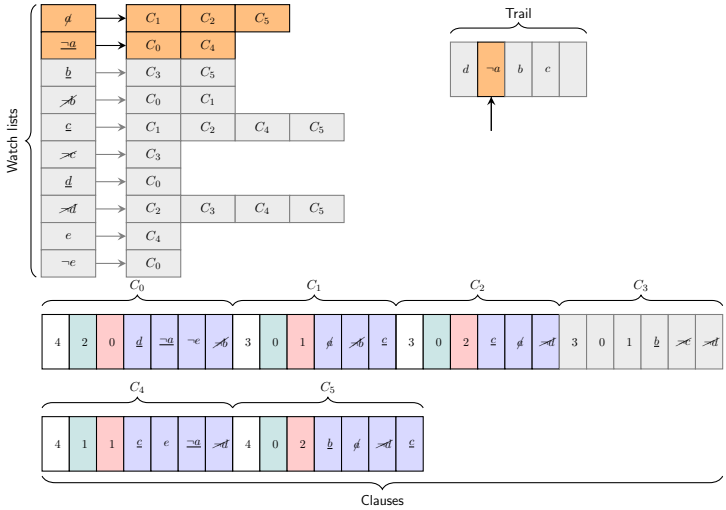
$$
\begin{aligned}
C_1 &= \underline{v_2} \vee \underline{v_1} \vee \boxed{v_3} \\
C_2 &= \neg\underline{v_1} \vee \neg\underline{v_2} \vee v_4 \\
C_3 &= \neg\underline{v_1} \vee \underline{v_5} \\
C_4 &= \underline{v_4} \vee \underline{v_6} \vee v_1 \\
C_5 &= \neg\underline{v_4} \vee \underline{v_6} \vee v_1 \\
C_6 &= \underline{v_5} \vee \underline{v_1}
\end{aligned}
$$



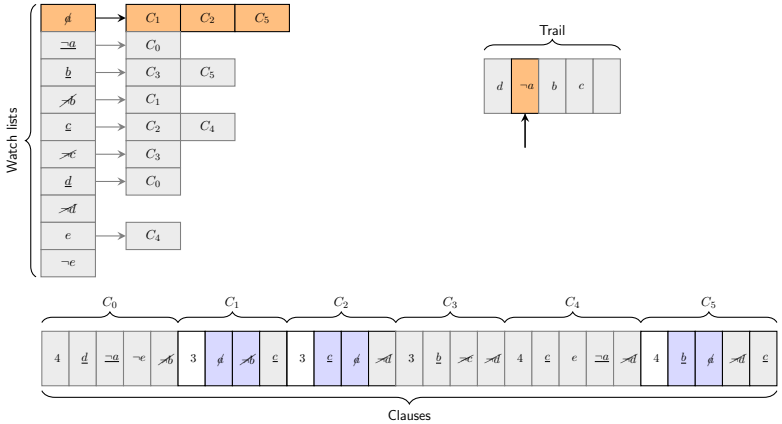
Reminder - Naive Scheme



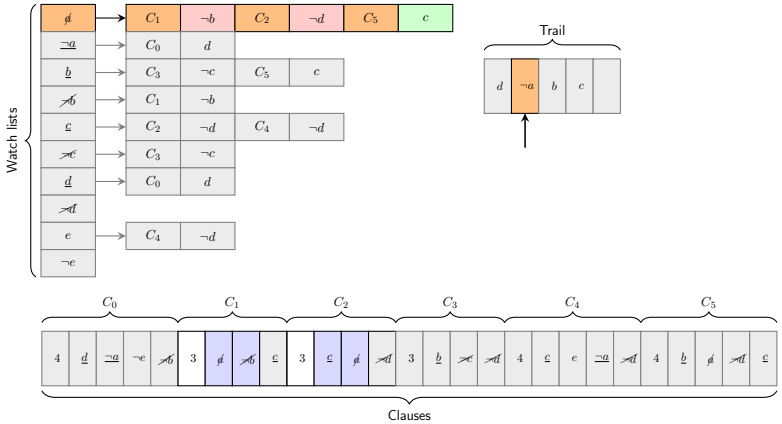
Reminder - Literal Indexing



Reminder - Two Watched Literals



Reminder - Blockers



Does it matter?

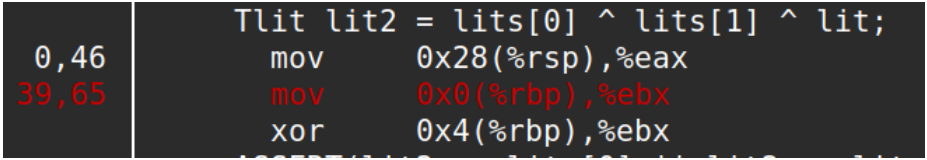


Figure: Time spent dereferencing clause pointers in NapSAT. This was measured using perf for 52 seconds.

Clauses in NapSAT

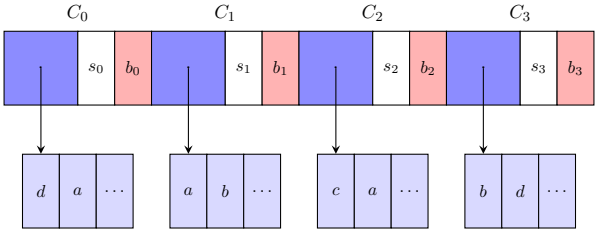


Figure: Representation of clauses in NapSAT. Clauses are stored as a fixed size structure containing a pointer to the literals, the size s of the clause, and a blocker literal b.

Array Representation of Watch Lists in NapSAT

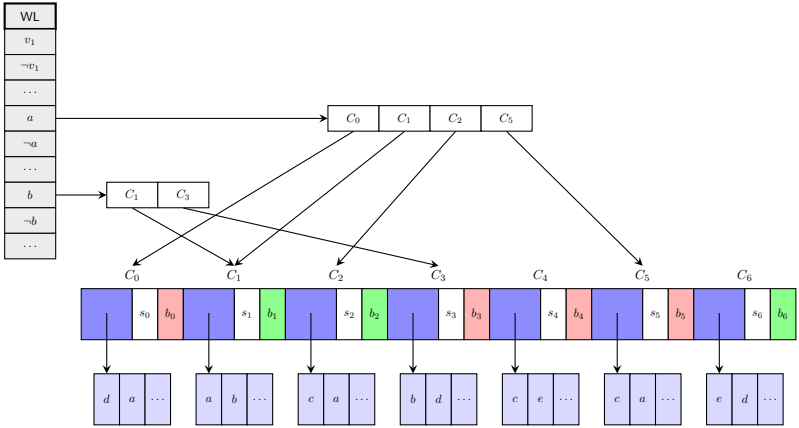


Figure: Array-based representation of watch lists. The watch list of a is $\{C_0, C_1, C_2, C_5\}$ and the watch list of b is $\{C_1, C_3\}$.

Array Representation of Watch Lists in MiniSAT

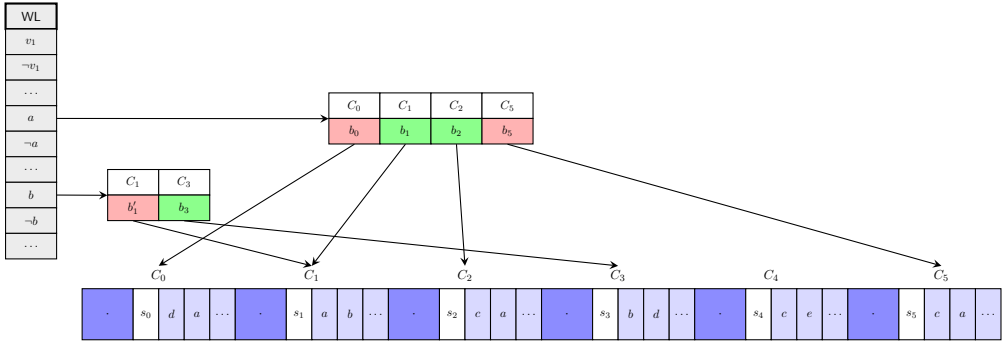


Figure: Array-based representation of watch lists in MiniSAT. The blockers are attached to the watch list.

Can we do better?

We know that each clause is exactly in two watch lists.
Because there are two watched literals per clause, and the clause is in the watch lists of the two literals.

Can we do better?

We know that each clause is exactly in two watch lists.
Because there are two watched literals per clause, and the clause is in the watch lists of the two literals.

Do we need extensible data structures?

We could use a double-tailed singly linked list.

Linked List Representation of Watch Lists

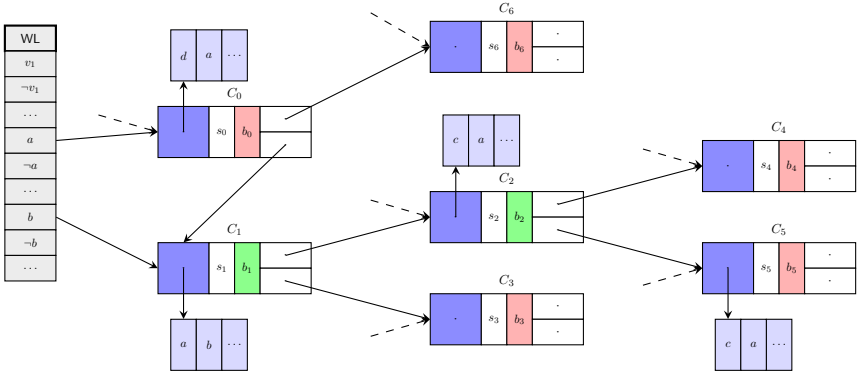


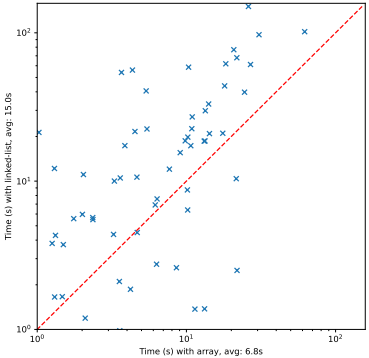
Figure: Representation of watch lists using the linked list data structure. The watch list of a is $\{C_0, C_1, C_2, C_5\}$ and the watch list of b is $\{C_1, C_3\}$.

Why Linked Lists?

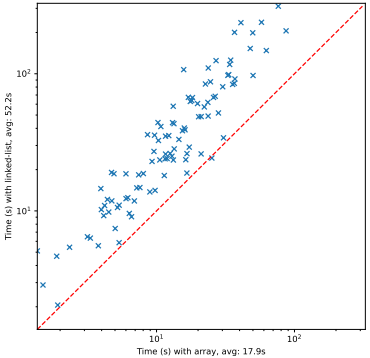
Table: Intuitive comparison of the array and linked list representation of watch lists.

Aspect	**Array**		**Linked list**	
Dereference level	2 levels	$(-)$	1 level	$(+)$
Memory usage	Extensible	$(-)$	Fixed	$(+)$
Insertion	$O(1)$ or $O(n)$	$(-)$	$O(1)$	$(+)$
Bookkeeping overhead	Low	$(+)$	High	$(-)$
Code complexity	Low	$(+)$	Medium	$(-)$

Experiments



(a) SAT instances.
$T(\text{linked list}) = 2.21 \times T(\text{array})$



(b) UNSAT instances.
$T(\text{linked list}) = 2.91 \times T(\text{array})$

Figure: Random 3-SAT instances of the SATLIB with 250 variables.

Going through the Array-based Watch List

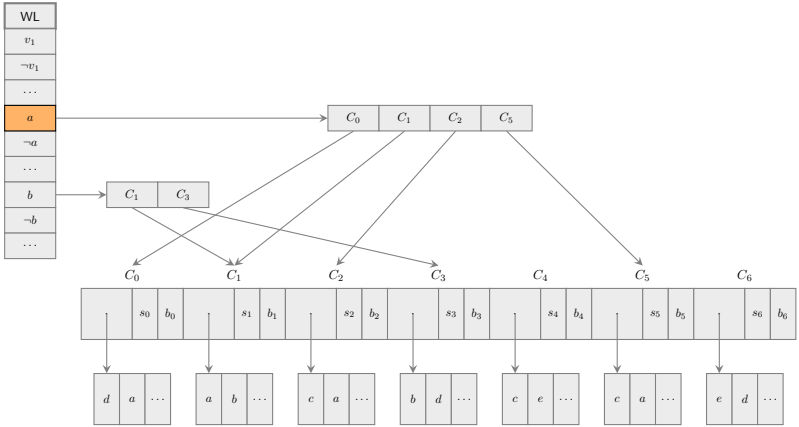


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

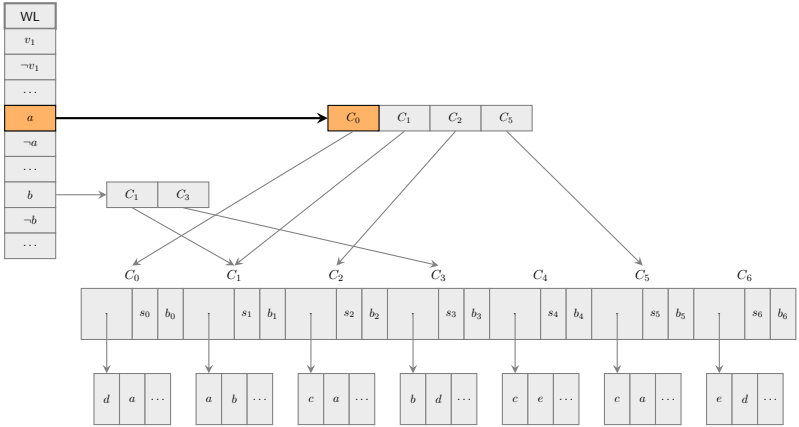


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

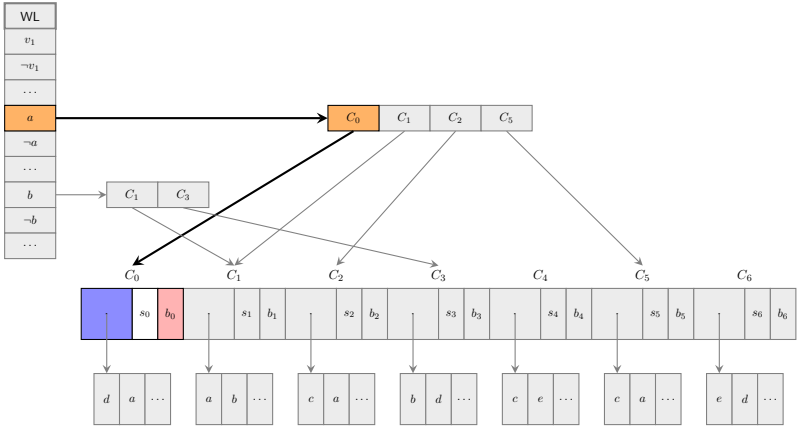


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

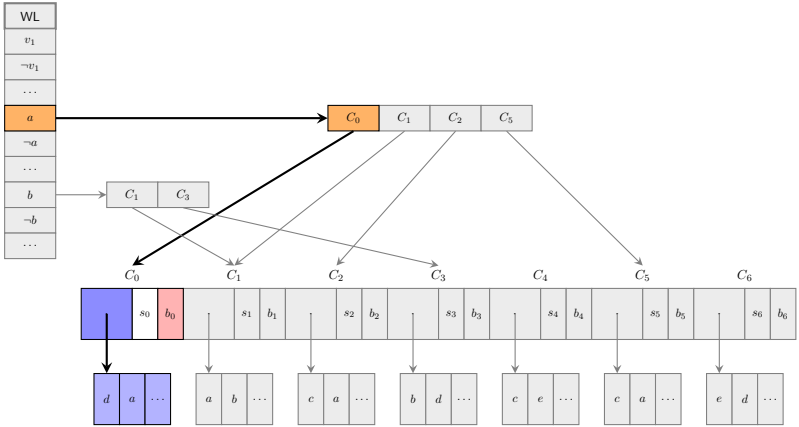


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

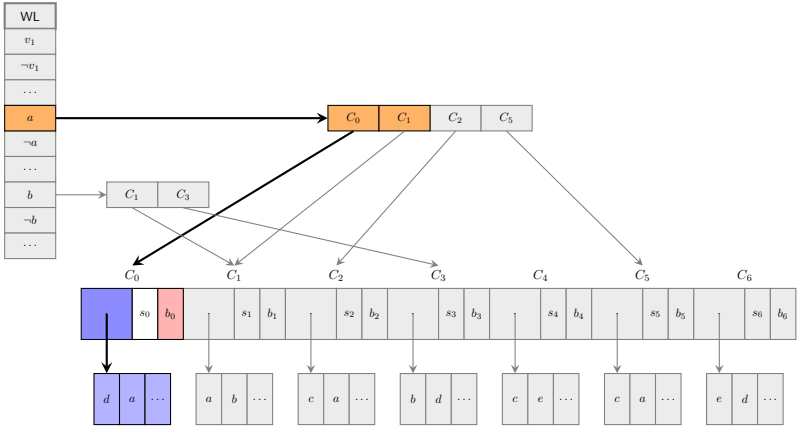


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

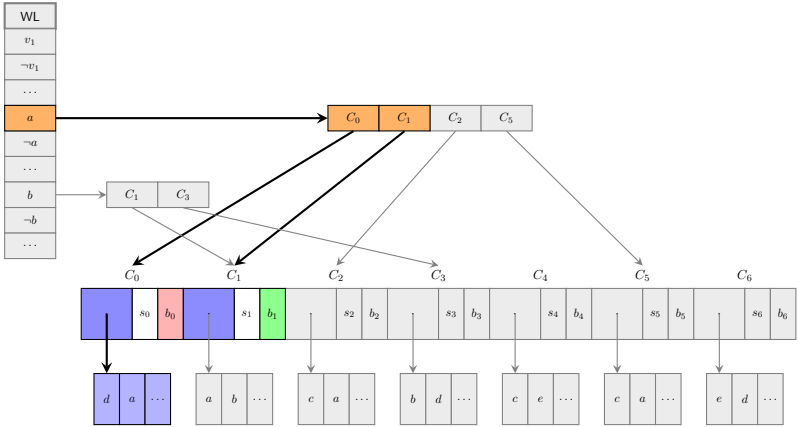


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

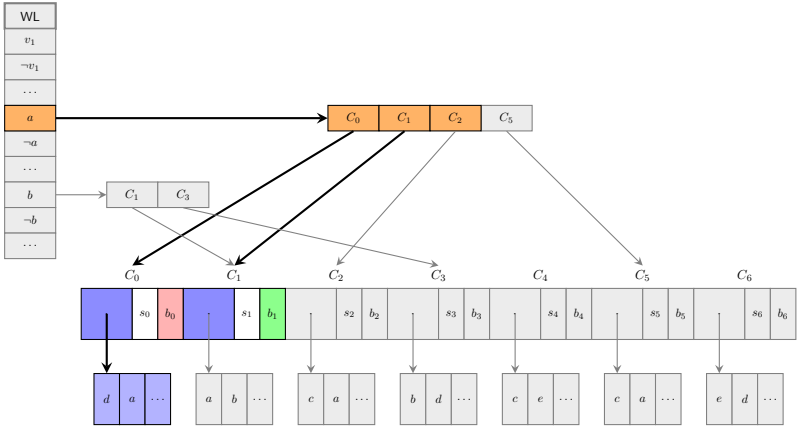


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

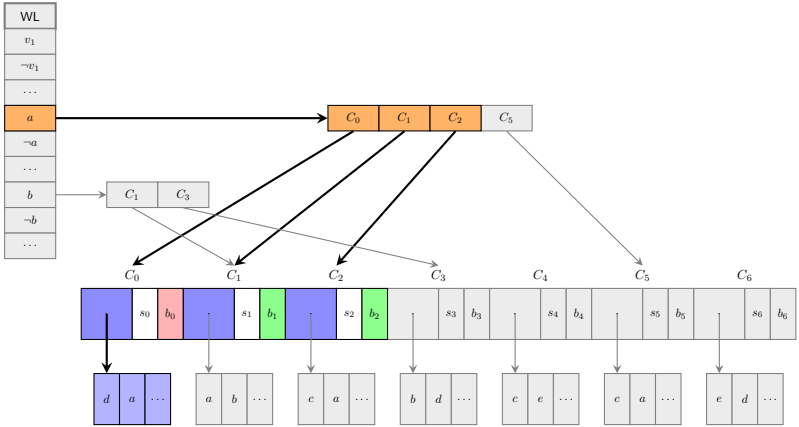


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

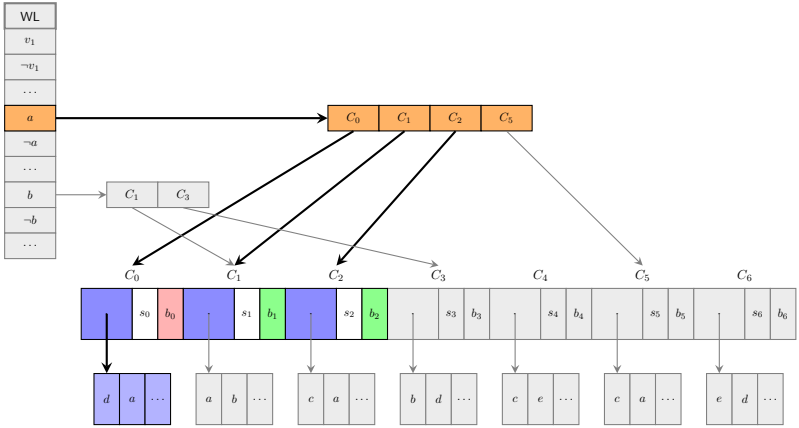


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

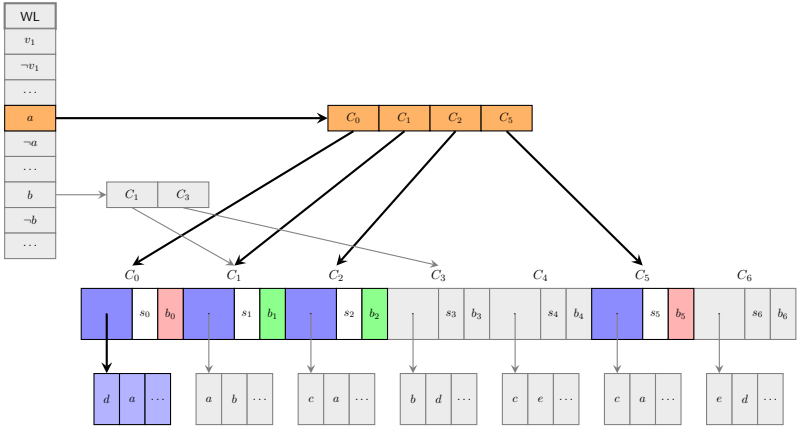


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Array-based Watch List

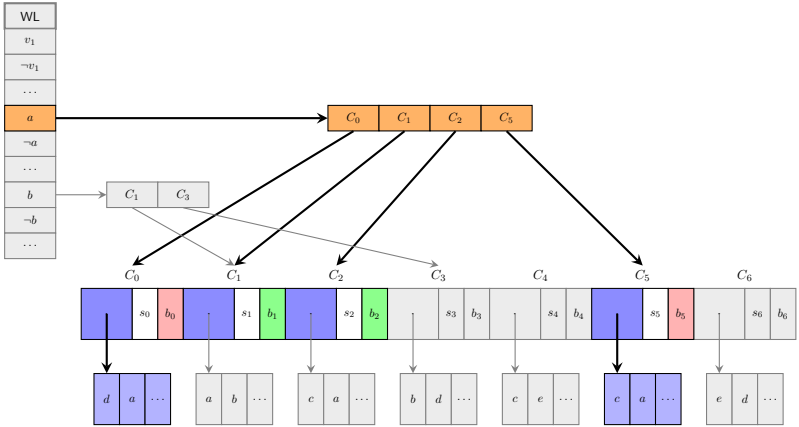


Figure: Iteration over the watch list of a in the array-based representation.

Going through the Linked List-based Watch List

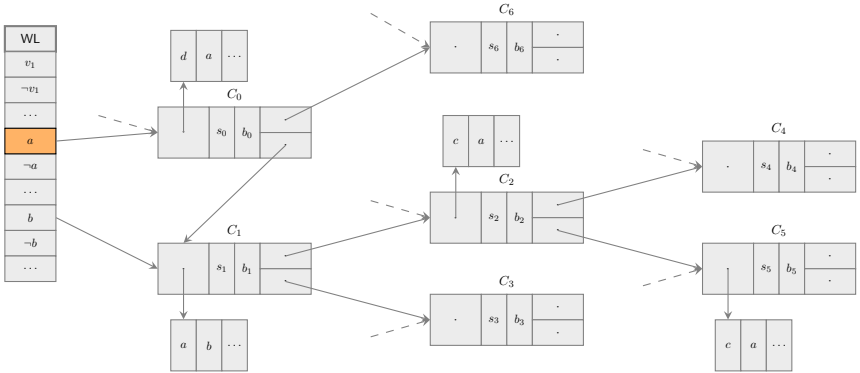


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

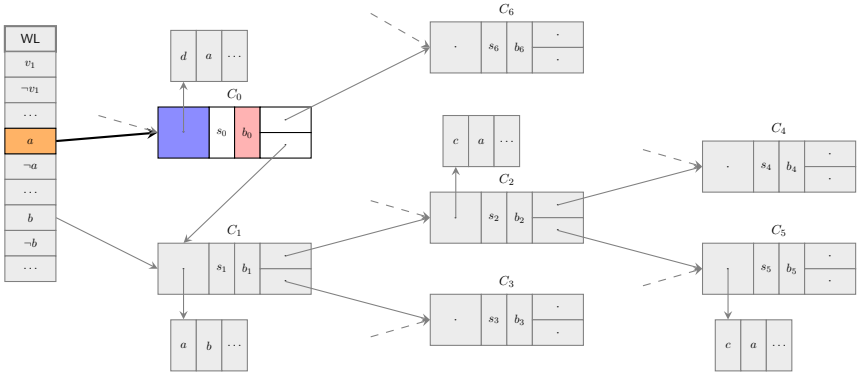


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

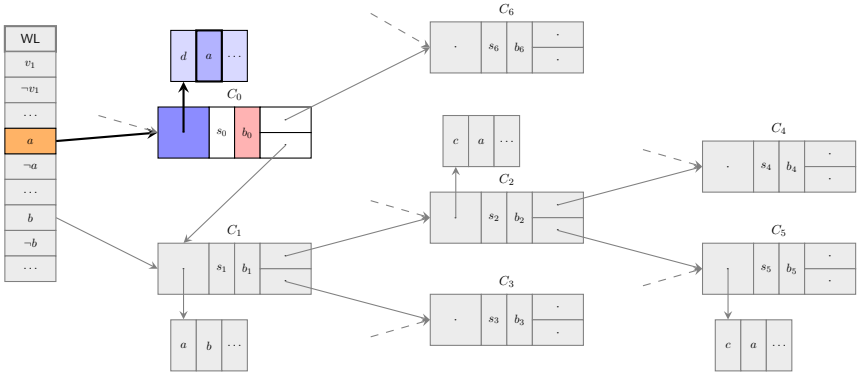


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

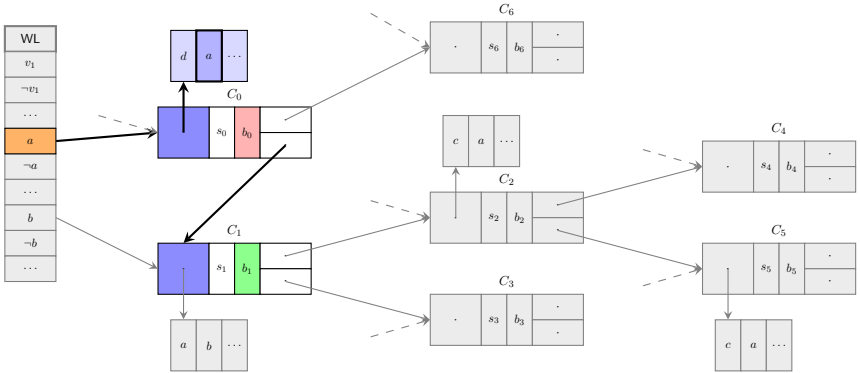


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

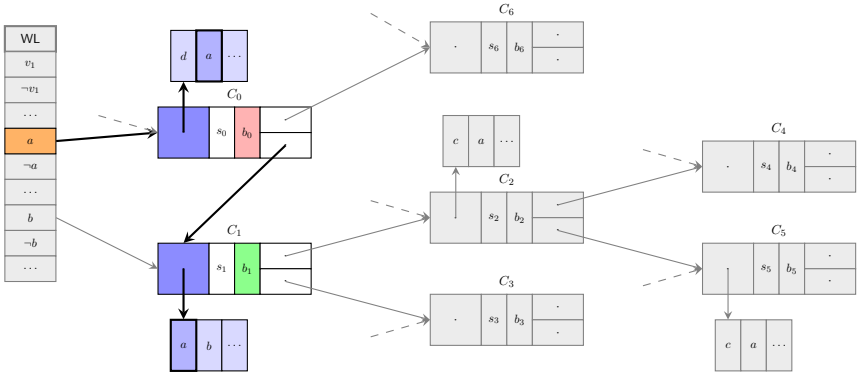


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

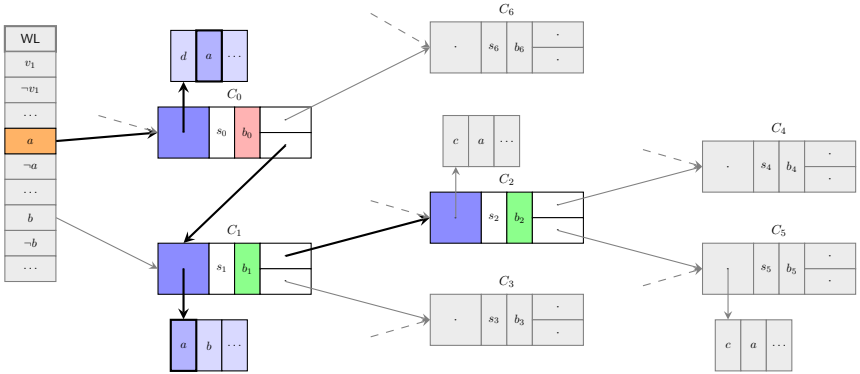


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

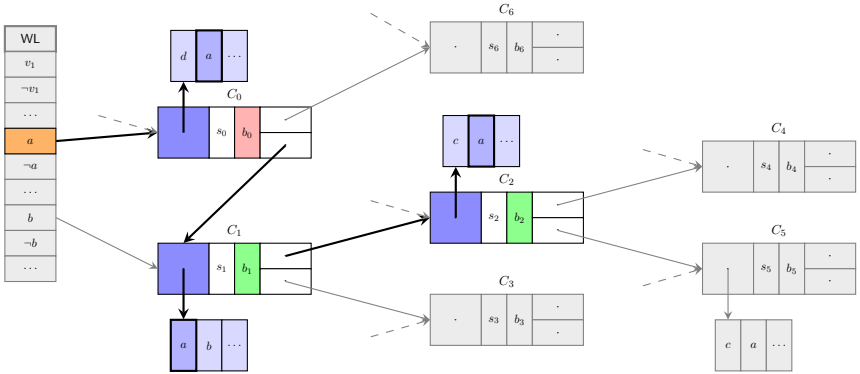


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

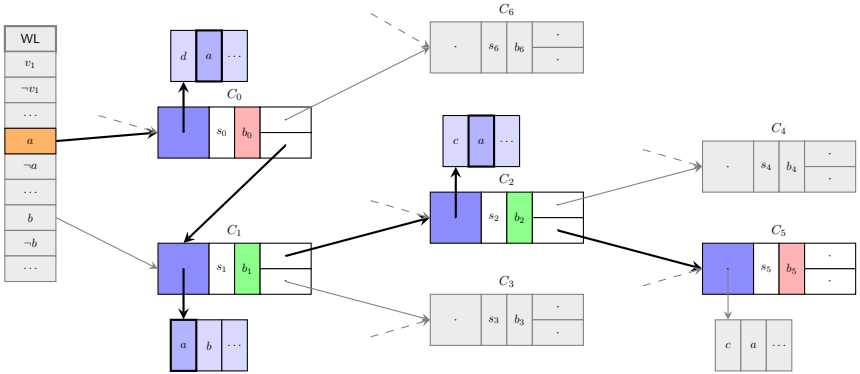


Figure: Iteration over the watch list of a in the linked list-based representation.

Going through the Linked List-based Watch List

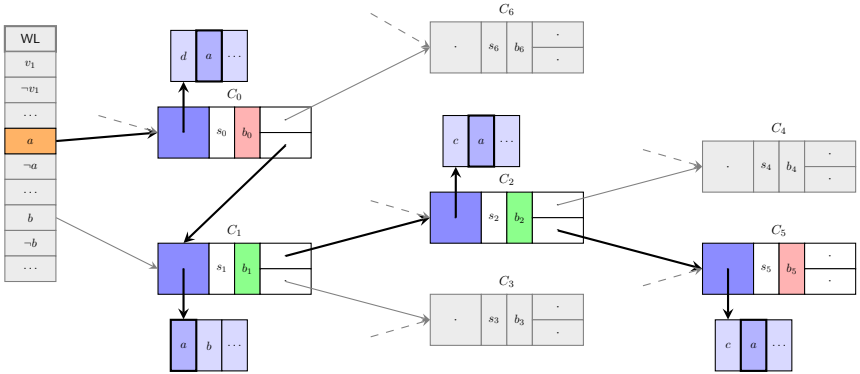


Figure: Iteration over the watch list of a in the linked list-based representation.

(Reminder) Going through the Array-based Watch List

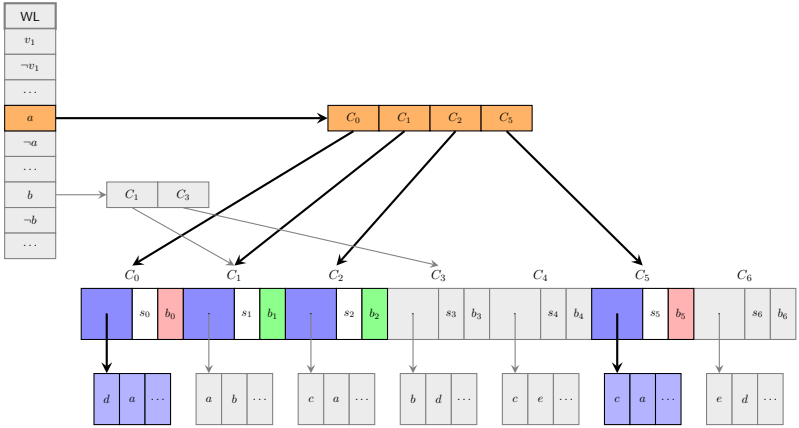


Figure: Iteration over the watch list of a in the array-based representation.

Does it generalize to MiniSAT?

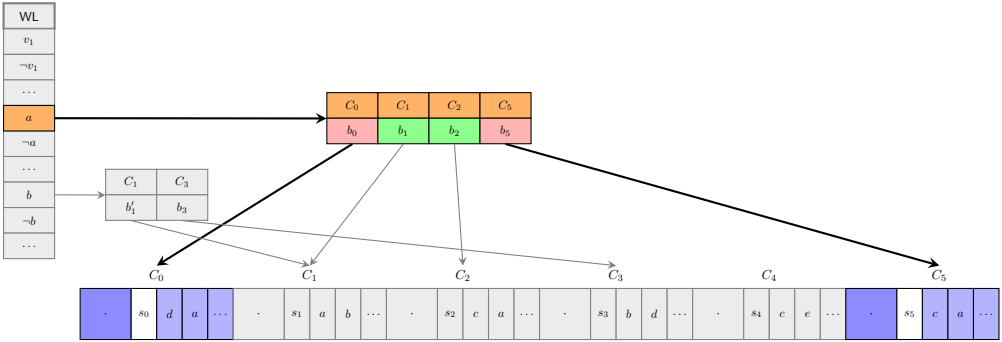


Figure: Iteration over the watch list of a in the array-based representation (MiniSAT).

Is dereferencing so important?

Table: Comparison of the average runtime of the different watch list representations on the uniform random 3-SAT instances of the SATLIB.

	uf200	uuf200	uf225	uuf225	uf250	uuf250
Linked list	0.28 s	0.75 s	1.78 s	5.10 s	15.00 s	52.20 s
Array	0.20 s	0.44 s	1.10 s	2.63 s	6.80 s	17.92 s
Array with dereference	0.17 s	0.46 s	1.16 s	2.92 s	8.52 s	24.52 s

Related Work

Thank you to the reviewers for pointing out the following related work:

- The original implementation of watched lists in [MMZ+01]
- Implementation and detailed analysis of linked lists in PicoSAT [Bie08]

This paper is an independent rediscovery of the same ideas.

Contrast with PicoSAT [Bie08]

Table: (Simplified) Results of the experiments ran with PicoSAT and presented in [Bie08]

Version	Solved	Unsolved	Sum Time (s)	Sum Space (MB)
Linked list	78	22	38240	5793
Array	76	24	40334	6768

Contrast with PicoSAT [Bie08]

Table: (Simplified) Results of the experiments ran with PicoSAT and presented in [Bie08]

Version	Solved	Unsolved	Sum Time (s)	Sum Space (MB)
Linked list	78	22	38240	5793
Array	76	24	40334	6768

Discussion

Why are the results so different?

Conclusion

Summary

- We reimplemented the linked list-based watch list idea on modern hardware.
- We empirically showed that the array-based watch list is faster.
- We discussed the importance of dereferencing pointers.
- We conclude that the linked list-based watch list is not a good idea.

References

Armin Biere.

Picosat essentials.

J. Satisf. Boolean Model. Comput., 4(2-4):75–97, 2008.

Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik.

Chaff: Engineering an efficient SAT solver.

In *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pages 530–535. ACM, 2001.

João P. Marques Silva and Karem A. Sakallah.

GRASP: A search algorithm for propositional satisfiability.

IEEE Trans. Computers, 48(5):506–521, 1999.