CDCL SAT Solvers & SAT-Based Problem Solving

Joao Marques-Silva[1,2] & Mikolas Janota[2]

[1]University College Dublin, Ireland

[2]IST/INESC-ID, Lisbon, Portugal

SAT/SMT Summer School 2013

Aalto University, Espoo, Finland

The Success of SAT

- Well-known NP-complete decision problem [C71]
- In practice, SAT is a success story of Computer Science
  - Hundreds (even more?) of practical applications

Part I

CDCL SAT Solvers

Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

What Next in CDCL Solvers?

Outline

Preliminaries

- Variables: $w, x, y, z, a, b, c, \ldots$
- Literals: $w, \bar{x}, \bar{y}, a, \ldots$, but also $\neg w, \neg y, \ldots$
- Clauses: disjunction of literals or set of literals
- Formula: conjunction of clauses or set of clauses
- Model (satisfying assignment): partial/total mapping from variables to $\{0, 1\}$
- Formula can be SAT/UNSAT

Preliminaries

- Variables: $w, x, y, z, a, b, c, \ldots$
- Literals: $w, \bar{x}, \bar{y}, a, \ldots$, but also $\neg w, \neg y, \ldots$
- Clauses: disjunction of literals or set of literals
- Formula: conjunction of clauses or set of clauses
- Model (satisfying assignment): partial/total mapping from variables to $\{0, 1\}$
- Formula can be SAT/UNSAT
- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\bar{w} \vee a) \wedge (\bar{x} \vee b) \wedge (\bar{y} \vee \bar{z} \vee c) \wedge (\bar{b} \vee \bar{c} \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$
    - $\{r, s, \bar{x}, y, \bar{w}, z, \bar{a}, b, c, d\}$

Resolution

- Resolution rule: [DP60,R65]

$$\frac{(\alpha \vee x) \qquad\qquad (\beta \vee \bar{x})}{(\alpha \vee \beta)}$$

  - Complete proof system for propositional logic

Resolution

- Resolution rule: [DP60,R65]

$$\frac{(\alpha \vee x) \qquad\qquad (\beta \vee \bar{x})}{(\alpha \vee \beta)}$$

  - Complete proof system for propositional logic

$$\begin{tikzcd}
(x \vee a) \arrow[d, dash] & (\bar{x} \vee a) \arrow[dl, dash] & & (\bar{y} \vee \bar{a}) \arrow[dr, dash] & (y \vee \bar{a}) \arrow[d, dash] \\
(a) \arrow[drr, dash] & & & & (\bar{a}) \arrow[dll, dash] \\
& & \perp & &
\end{tikzcd}$$

  - Extensively used with (CDCL) SAT solvers

Resolution

- Resolution rule: [DP60,R65]

$$\frac{(\alpha \vee x) \qquad\qquad (\beta \vee \bar{x})}{(\alpha \vee \beta)}$$

  - Complete proof system for propositional logic

$$\begin{tikzcd}
(x \vee a) \arrow[d, dash] & & (\bar{x} \vee a) \arrow[dll, dash] & & (\bar{y} \vee \bar{a}) \arrow[drr, dash] & & (y \vee \bar{a}) \arrow[d, dash] \\
(a) \arrow[drrr, dash] & & & & & & (\bar{a}) \arrow[dlll, dash] \\
& & & \perp & & &
\end{tikzcd}$$

  - Extensively used with (CDCL) SAT solvers

- Self-subsuming resolution (with $\alpha' \subseteq \alpha$): [e.g. SP04,EB05]

$$\frac{(\alpha \vee x) \qquad\qquad (\alpha' \vee \bar{x})}{(\alpha)}$$

  - (α) subsumes $(\alpha \vee x)$

Unit Propagation

$$
\begin{aligned}
\mathcal{F} \quad = \quad & (r) \wedge (\bar{r} \vee s) \wedge \\
& (\bar{w} \vee a) \wedge (\bar{x} \vee \bar{a} \vee b) \\
& (\bar{y} \vee \bar{z} \vee c) \wedge (\bar{b} \vee \bar{c} \vee d)
\end{aligned}
$$

Unit Propagation

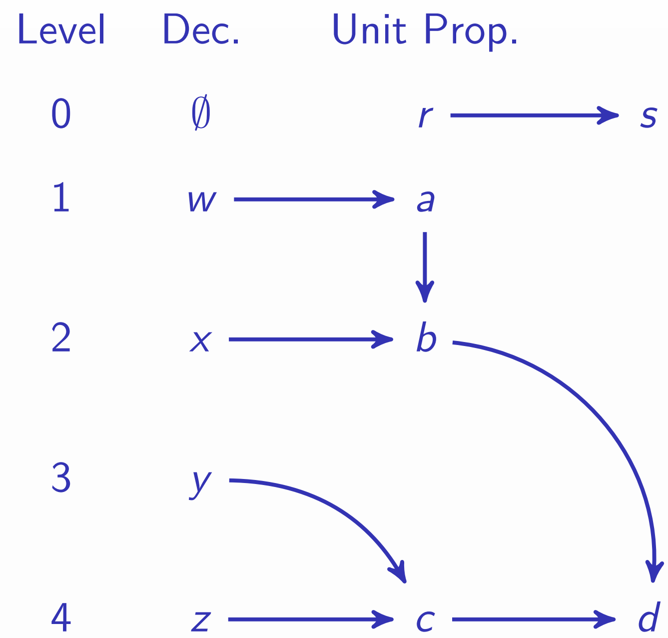
$$\begin{aligned}\mathcal{F} &= (r) \wedge (\bar{r} \vee s) \wedge \\ &\quad (\bar{w} \vee a) \wedge (\bar{x} \vee \bar{a} \vee b) \\ &\quad (\bar{y} \vee \bar{z} \vee c) \wedge (\bar{b} \vee \bar{c} \vee d)\end{aligned}$$

- Decisions / Variable Branchings:
  $w = 1, x = 1, y = 1, z = 1$

Unit Propagation

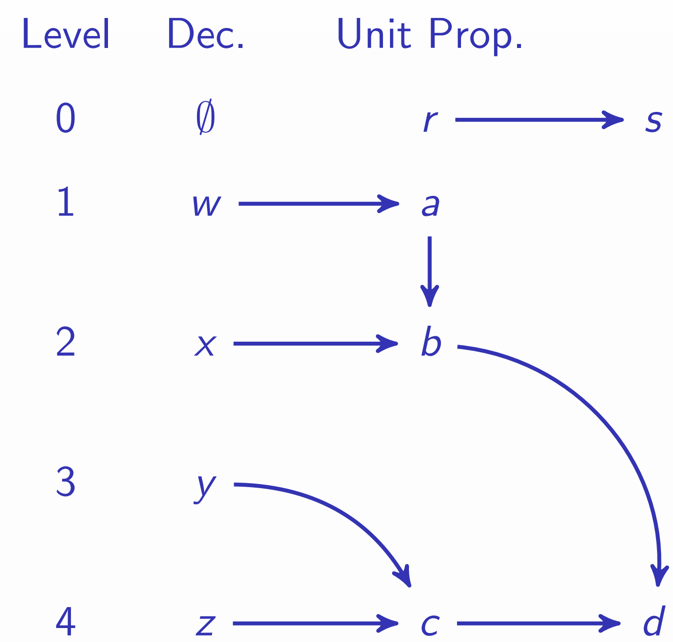
$$\mathcal{F} = (r) \wedge (\bar{r} \vee s) \wedge (\bar{w} \vee a) \wedge (\bar{x} \vee \bar{a} \vee b) \wedge (\bar{y} \vee \bar{z} \vee c) \wedge (\bar{b} \vee \bar{c} \vee d)$$

- Decisions / Variable Branchings:
  $w = 1, x = 1, y = 1, z = 1$



Unit Propagation

$$\mathcal{F} = (r) \wedge (\bar{r} \vee s) \wedge \\ (\bar{w} \vee a) \wedge (\bar{x} \vee \bar{a} \vee b) \\ (\bar{y} \vee \bar{z} \vee c) \wedge (\bar{b} \vee \bar{c} \vee d)$$

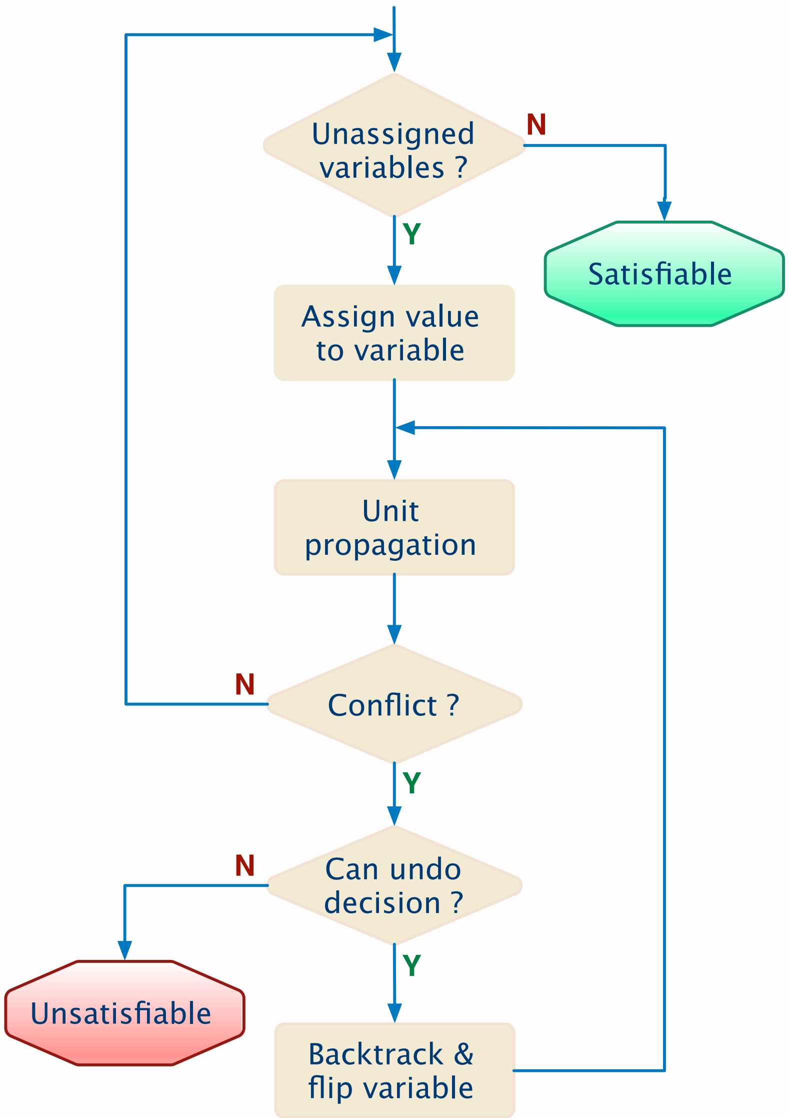


- Decisions / Variable Branchings:
  $w = 1, x = 1, y = 1, z = 1$

- Additional definitions:
  - Antecedent (or reason) of an implied assignment
    - $(\bar{b} \vee \bar{c} \vee d)$ for d
  - Associate assignment with decision levels
    - $w = 1 @ 1,\ x = 1 @ 2,\ y = 1 @ 3,\ z = 1 @ 4$
    - $r = 1 @ 0,\ d = 1 @ 4,\ \ldots$

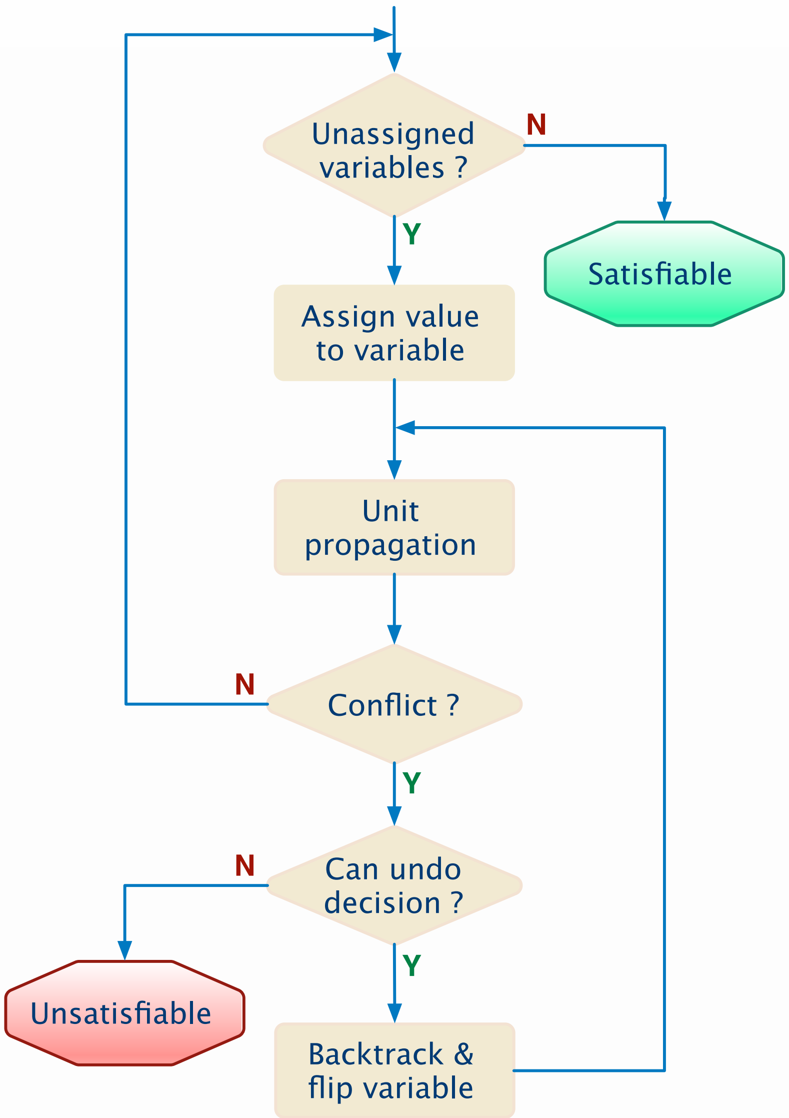
Outline

The DPLL Algorithm



- Optional: pure literal rule

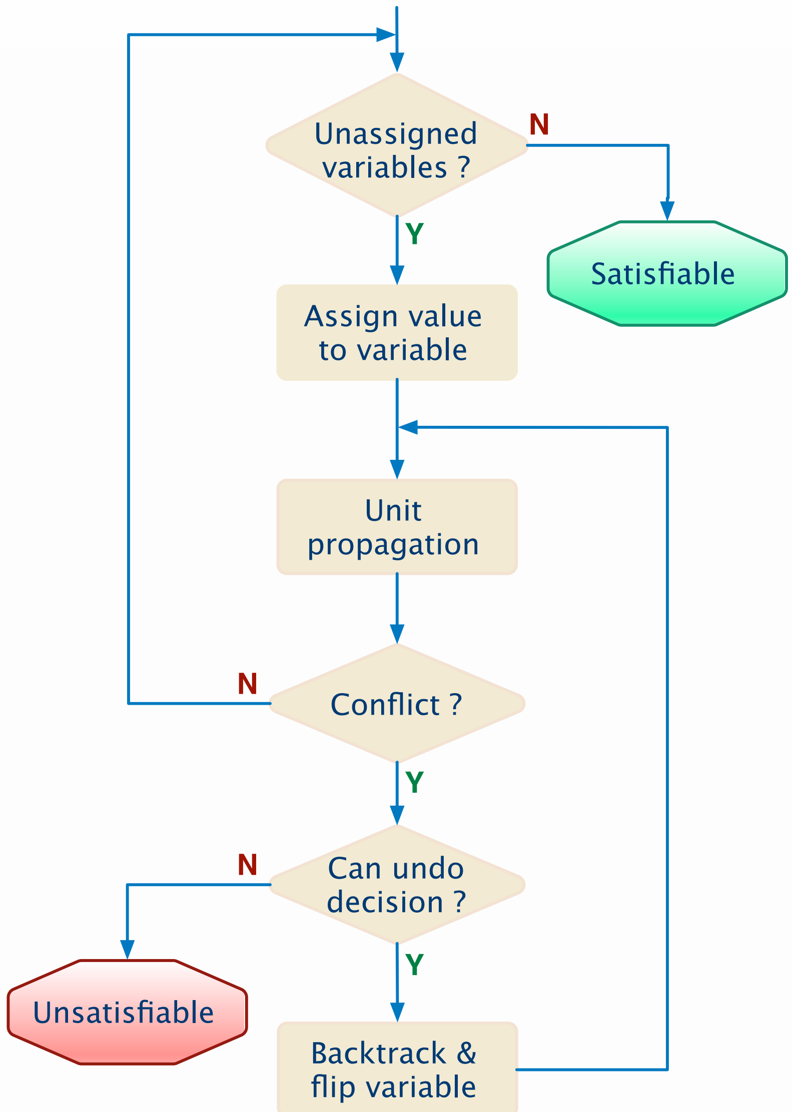
The DPLL Algorithm



$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

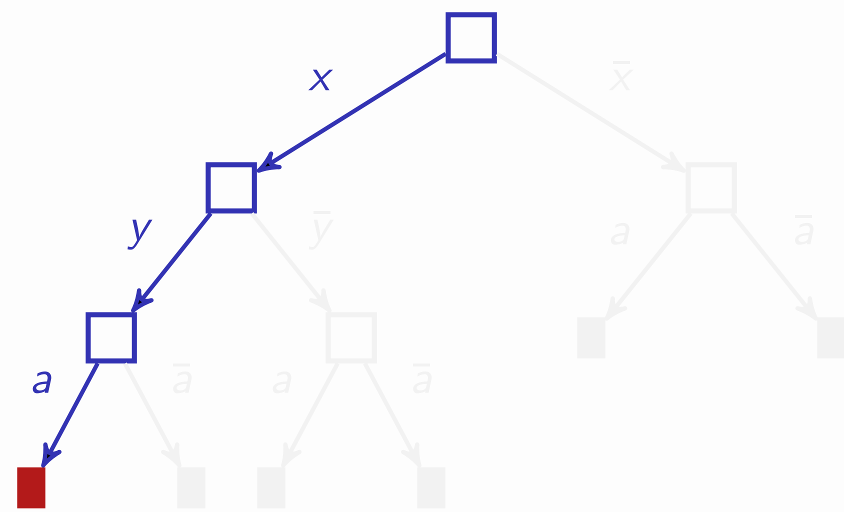
- Optional: pure literal rule

The DPLL Algorithm



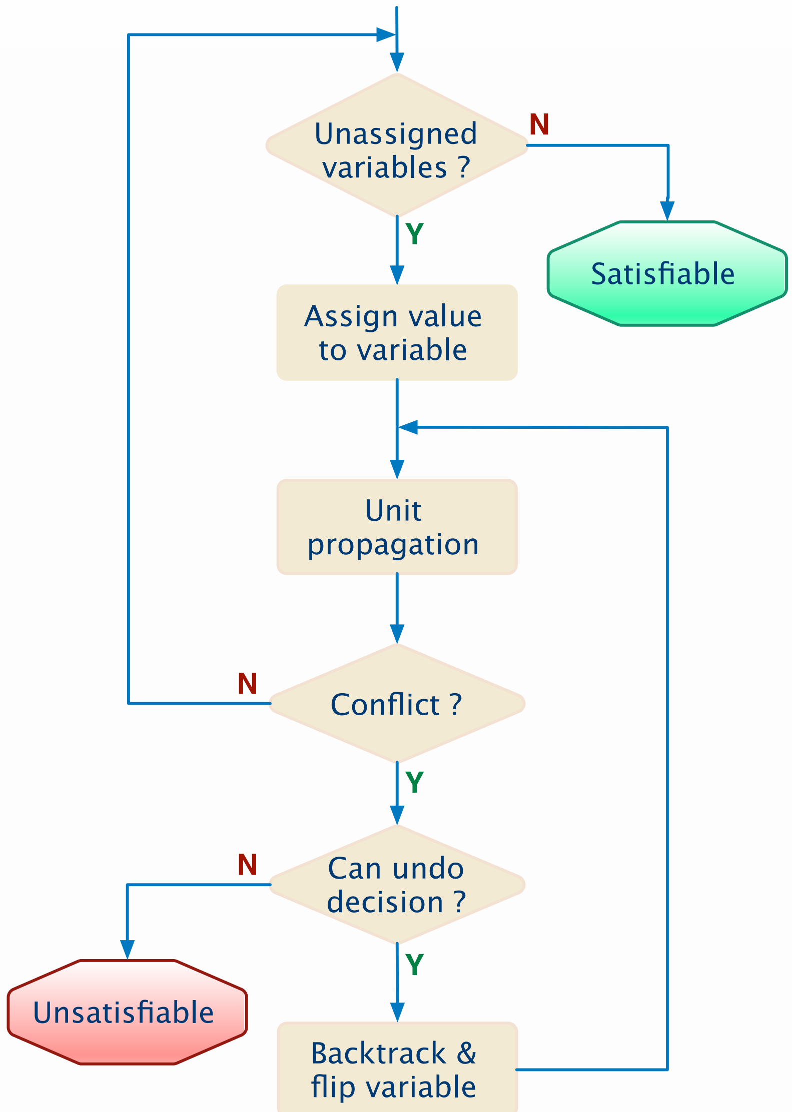
$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	x	
2	y	
3	a	$\longrightarrow b \longrightarrow \perp$



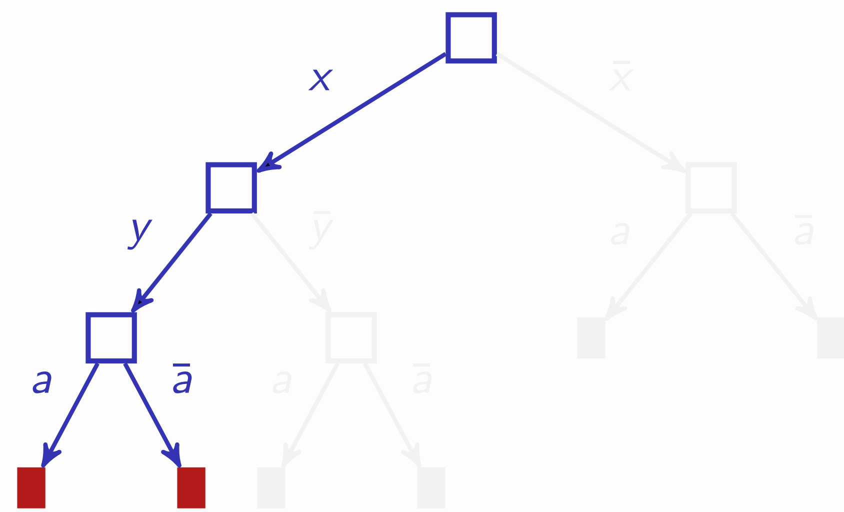
- Optional: pure literal rule

The DPLL Algorithm



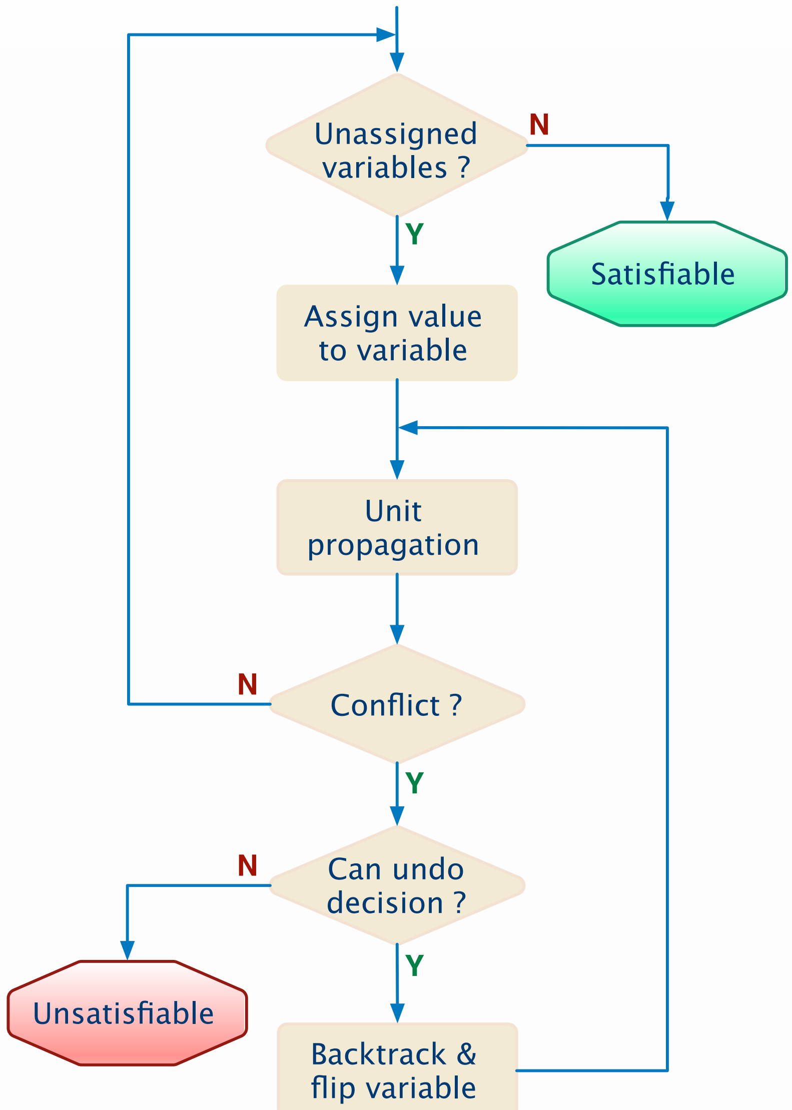
$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	x	
2	y	
3	$\bar{a}$	$\rightarrow \bar{b} \rightarrow \perp$



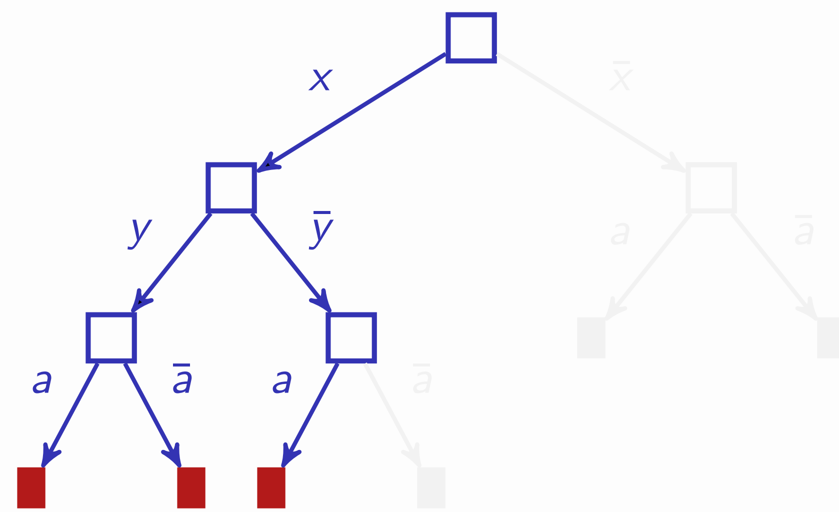
- Optional: pure literal rule

The DPLL Algorithm



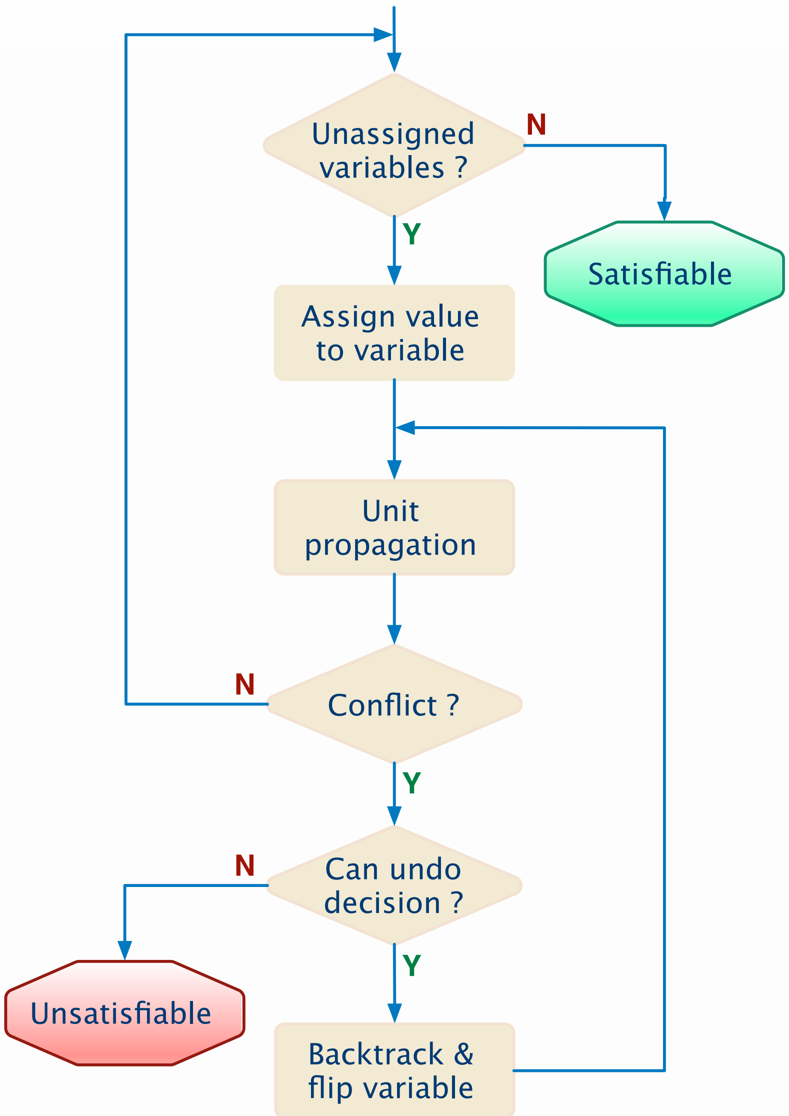
$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	x	
2	$\bar{y}$	
3	a	$\longrightarrow b \longrightarrow \perp$



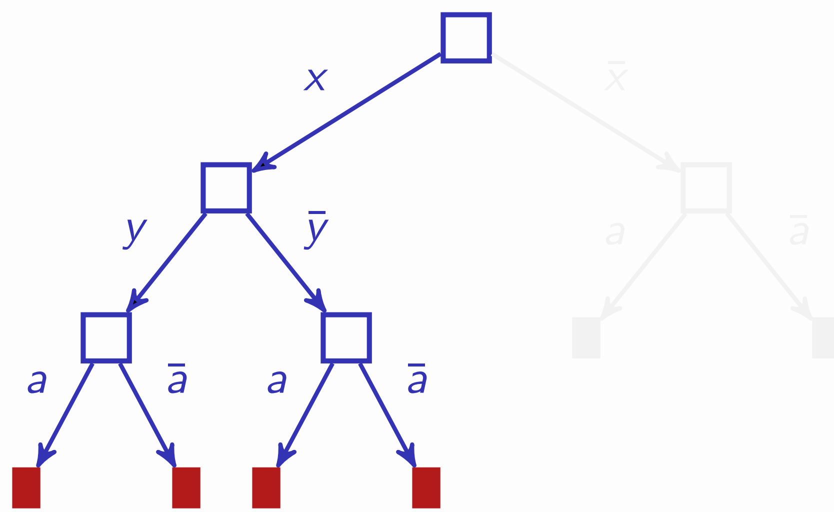
- Optional: pure literal rule

The DPLL Algorithm



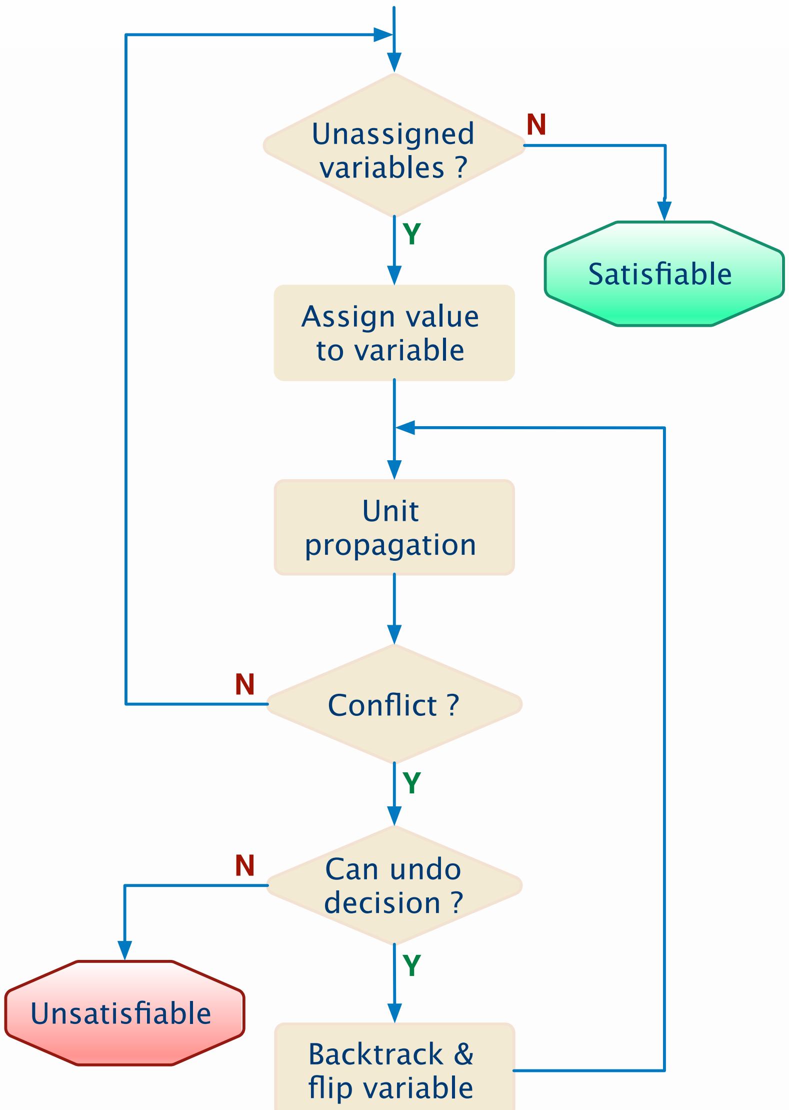
$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	x	
2	$\bar{y}$	
3	$\bar{a}$	$\longrightarrow \bar{b} \longrightarrow \perp$



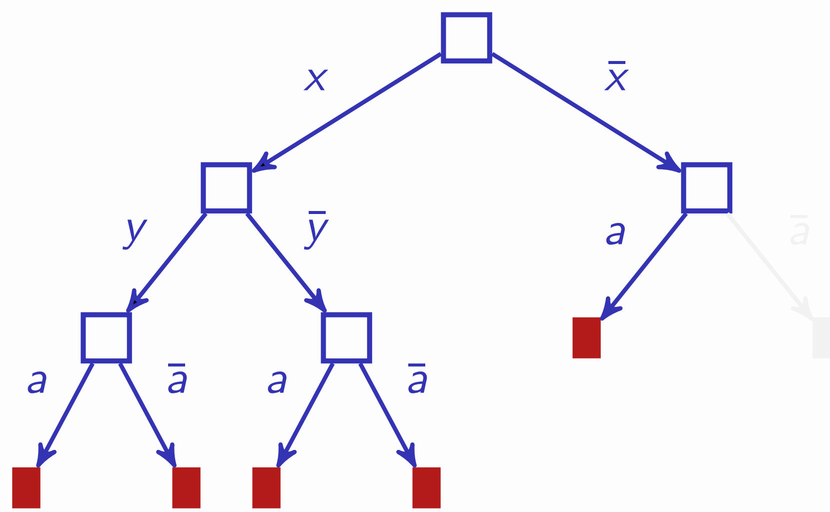
- Optional: pure literal rule

The DPLL Algorithm



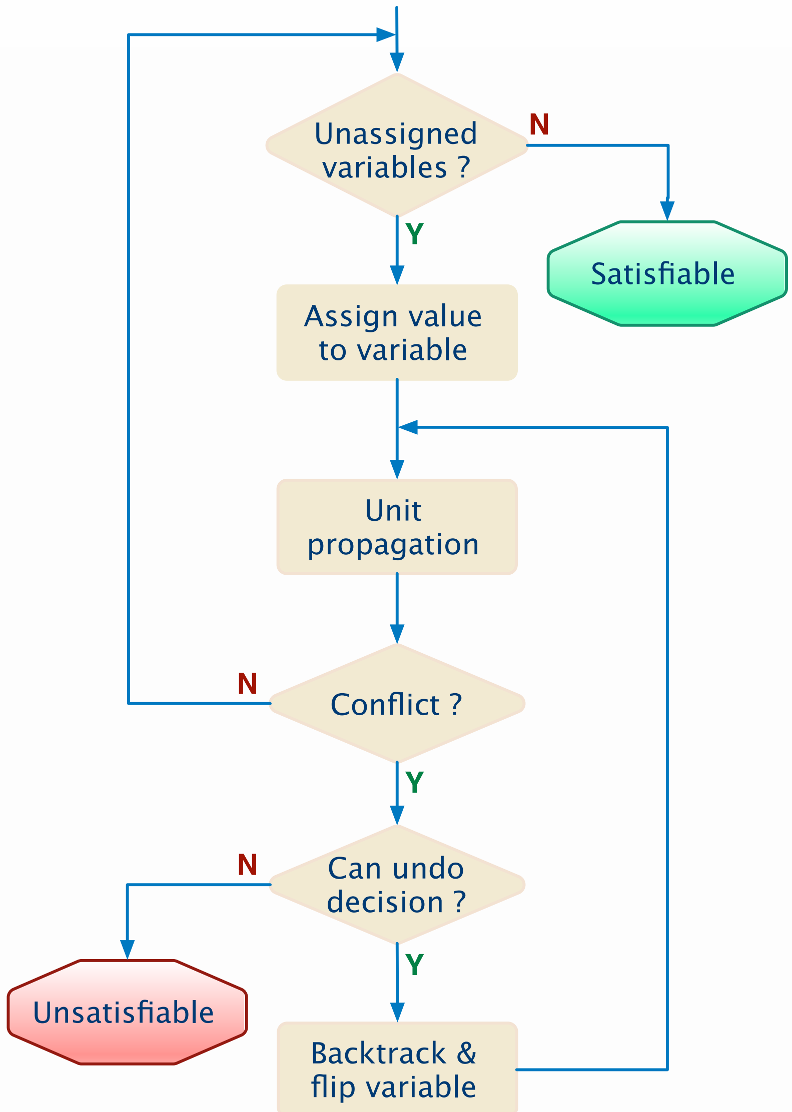
$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	$\bar{x}$	$\longrightarrow y$
2	a	$\longrightarrow b \longrightarrow \perp$



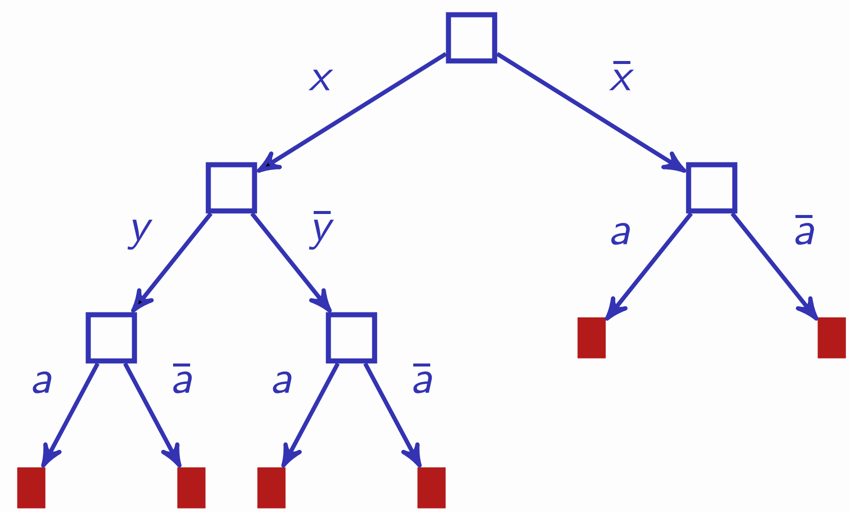
- Optional: pure literal rule

The DPLL Algorithm



$$\mathcal{F} = (x \vee y) \wedge (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

Level	Dec.	Unit Prop.
0	$\emptyset$	
1	$\bar{x}$	$\longrightarrow y$
2	$\bar{a}$	$\longrightarrow \bar{b} \longrightarrow \bot$



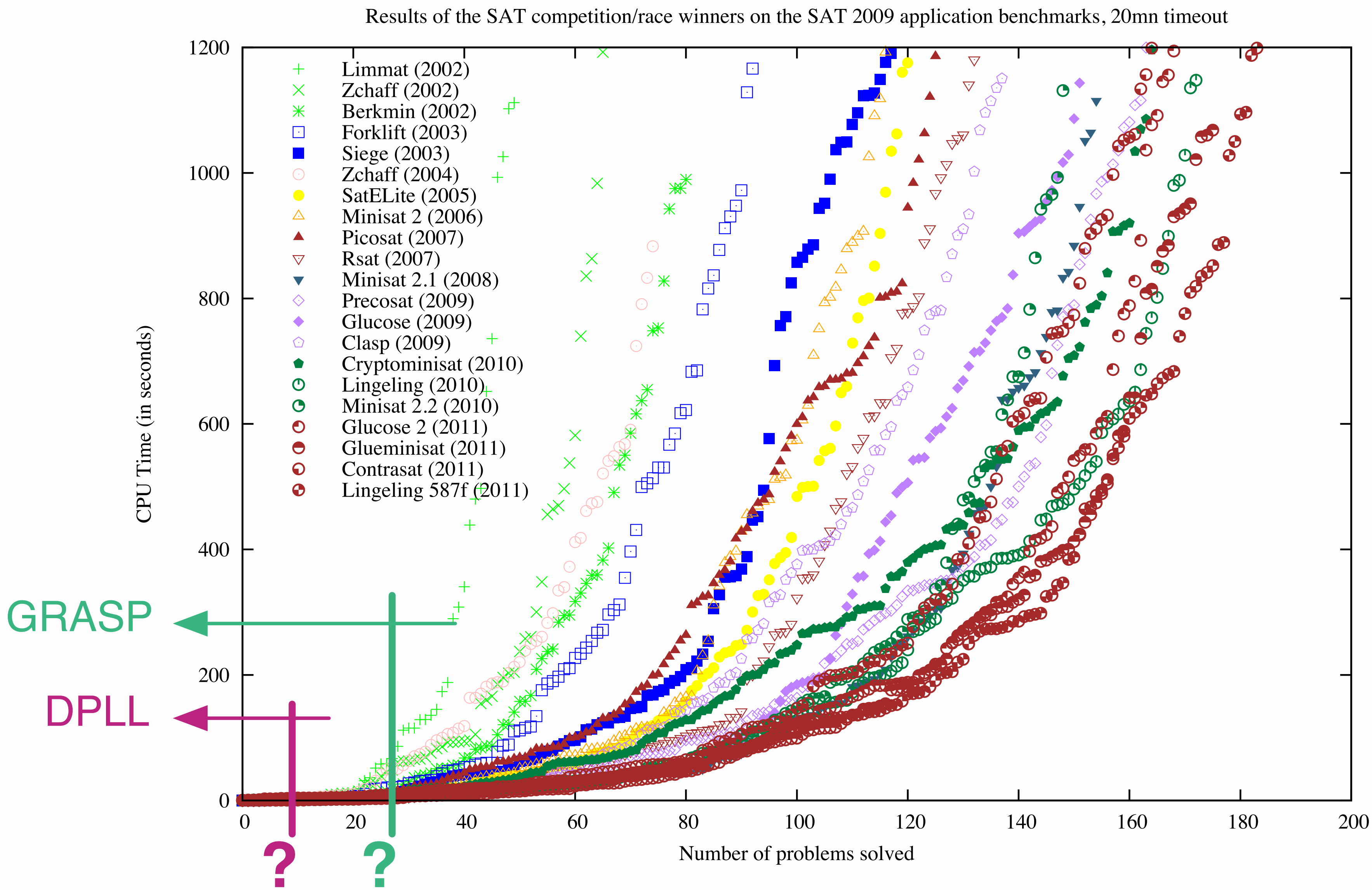
- Optional: pure literal rule

Outline

What is a CDCL SAT Solver?

- Extend DPLL SAT solver with: [DP60,DLL62]
  - Clause learning & non-chronological backtracking [MSS96,BS97,Z97]
    - Exploit UIPs [MSS96,SSS12]
    - Minimize learned clauses [SB09,VG09]
    - Opportunistically delete clauses [MSS96,MSS99,GN02]
  - Search restarts [GSK98,BMS00,H07,B08]
  - Lazy data structures
    - Watched literals [MMZZM01]
  - Conflict-guided branching
    - Lightweight branching heuristics [MMZZM01]
    - Phase saving [PD07]
  - ...

How Significant are CDCL SAT Solvers?

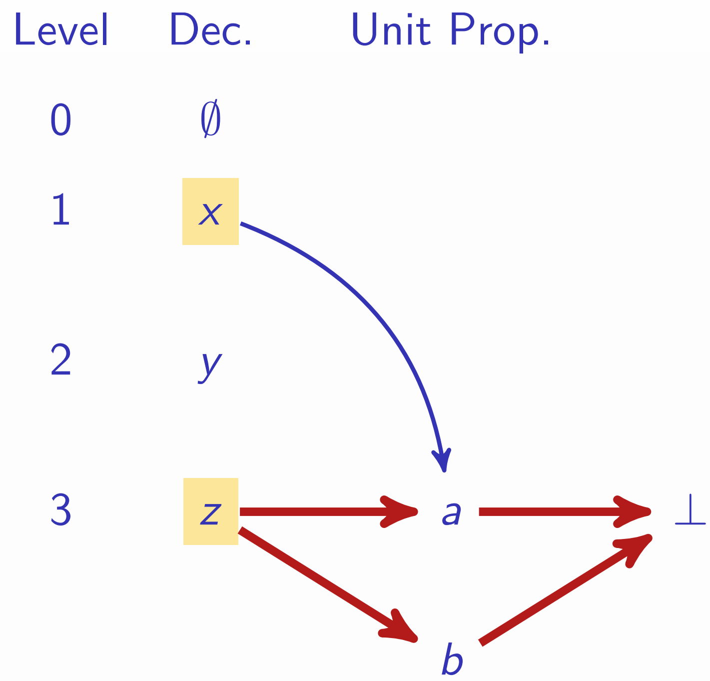


Outline

Clause Learning

$$
\begin{tikzcd}
\text{Level} & \text{Dec.} & \text{Unit Prop.} & \\
0 & \emptyset & & \\
1 & x \arrow[ddr, bend left] & & \\
2 & y & & \\
3 & z \arrow[r] \arrow[dr] & a \arrow[r] & \perp \\
 & & b \arrow[ur] &
\end{tikzcd}
$$

Clause Learning



$(\bar{a} \vee \bar{b})$ $(\bar{z} \vee b)$ $(\bar{x} \vee \bar{z} \vee a)$

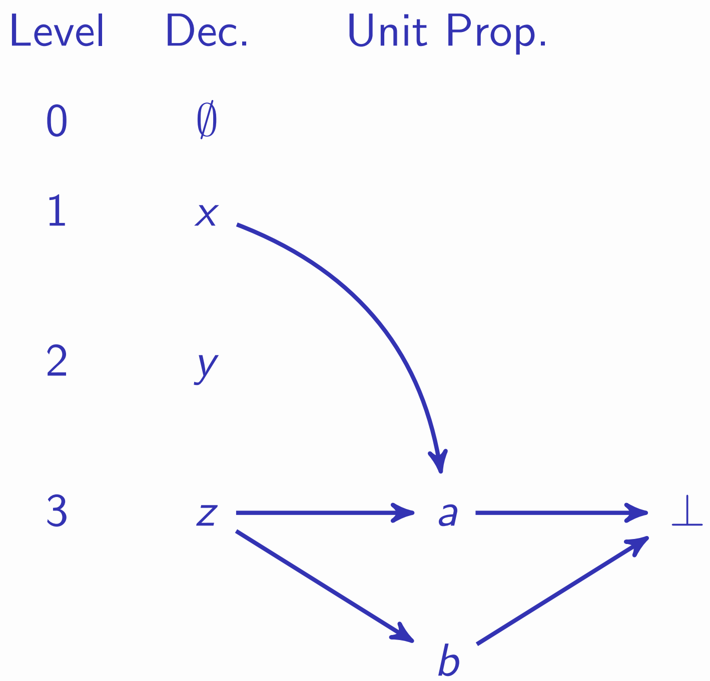
ab -> false = !(ab) + false = !a + !b

z -> b = !z + b

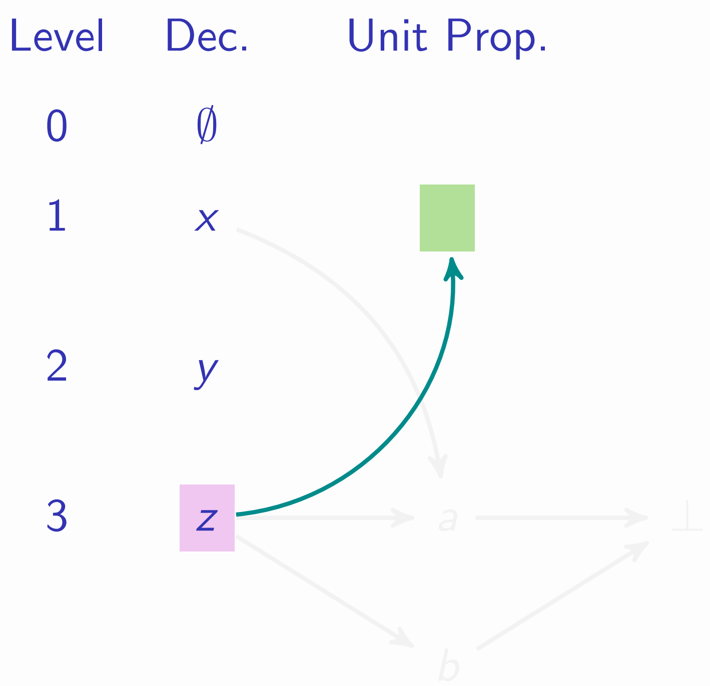
xz -> a. =. !(xz) + a =. !x + !z + a

- Analyze conflict
  - Reasons: x and z
    - Decision variable & literals assigned at lower decision levels
  - Create **new** clause: $(\bar{x} \vee \bar{z})$
- Can relate clause learning with resolution

Clause Learning – After Bracktracking

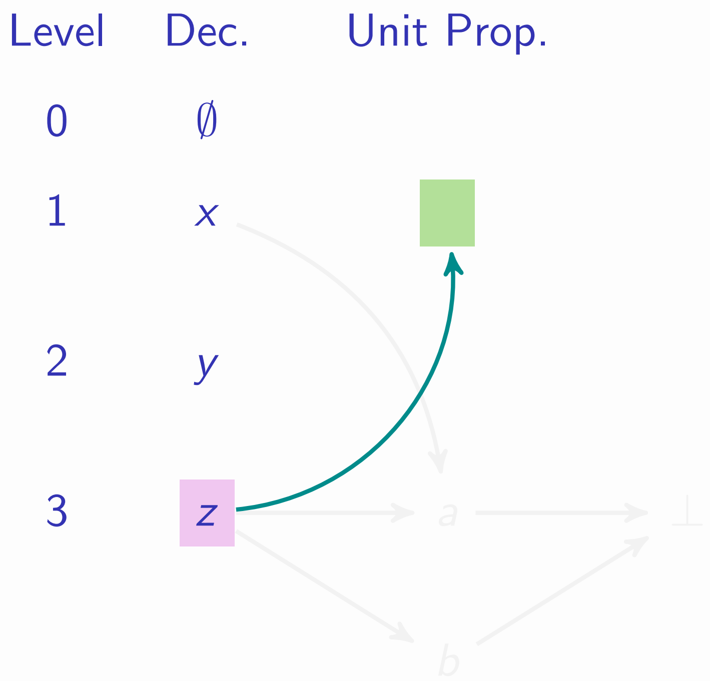


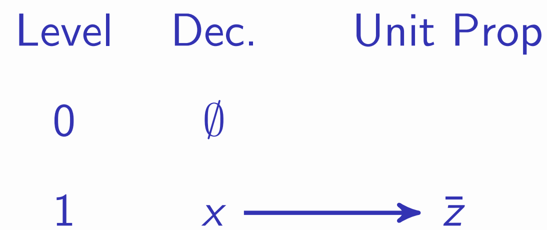
Clause Learning – After Bracktracking



- Clause $(\bar{x} \vee \bar{z})$ is asserting at decision level 1

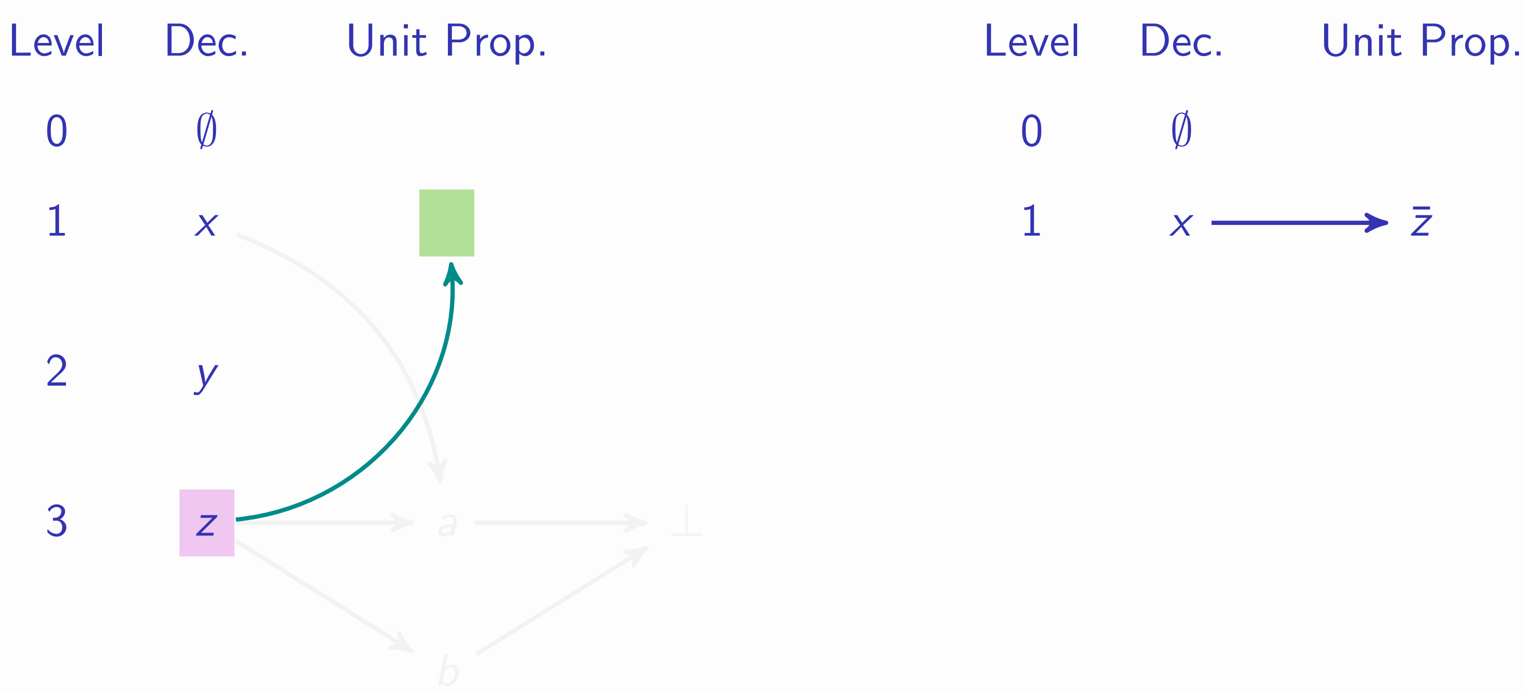
Clause Learning – After Bracktracking





- Clause $(\bar{x} \vee \bar{z})$ is asserting at decision level 1

Clause Learning – After Bracktracking



- Clause $(\bar{x} \vee \bar{z})$ is asserting at decision level 1
- Learned clauses are always asserting [MSS96,MSS99]
- Backtracking differs from plain DPLL:
  - Always bactrack after a conflict [MMZZM01]