Presentation of →

Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T)

ROBERT NIEUWENHUIS AND ALBERT OLIVERAS

Technical University of Catalonia, Barcelona, Spain

AND

CESARE TINELLI

The University of Iowa, Iowa City, Iowa

Abstract. We first introduce *Abstract DPLL*, a rule-based formulation of the Davis–Putnam–Logemann–Loveland (DPLL) procedure for propositional satisfiability. This abstract framework allows one to cleanly express practical DPLL algorithms and to formally reason about them in a simple way. Its properties, such as soundness, completeness or termination, immediately carry over to the modern DPLL implementations with features such as backjumping or clause learning.

We then extend the framework to Satisfiability Modulo background Theories (SMT) and use it to model several variants of the so-called *lazy approach* for SMT. In particular, we use it to introduce a few variants of a new, efficient and modular approach for SMT based on a general DPLL(X) engine, whose parameter X can be instantiated with a specialized solver $Solver_T$ for a given theory T, thus producing a DPLL(T) system. We describe the high-level design of DPLL(X) and its cooperation with $Solver_T$, discuss the role of *theory propagation*, and describe different DPLL(T) strategies for some theories arising in industrial applications.

Our extensive experimental evidence, summarized in this article, shows that DPLL(T) systems can significantly outperform the other state-of-the-art tools, frequently even in orders of magnitude, and have better scaling properties.

Categories and Subject Descriptors: B.6.3 [**Logic Design**]: Design Aids—*Verification*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Computational logic*; *verification*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*Deduction (e.g., natural, rule-based)*

General Terms: Theory, Verification

Additional Key Words and Phrases: SAT solvers, Satisfiability Modulo Theories

by

Ramūnas Gutkovas

SMT Reading Group

Uppsala

2012-10-31

Background for Abstract DPLL

Modern **SAT** (~2006) solvers based on **DPLL** use many **extensions** and combinations thereof

Original **DPLL** [Davis et al. '62]

Problem: these extensions lack formal treatment

Background for Abstract DPLL

Modern **SAT** (~2006) solvers based on **DPLL** use many **extensions** and combinations thereof

Original **DPLL** [Davis et al. '62]

+ lemma learning

+ restarts

+ backjumping

+ background theories

Problem: these extensions lack formal treatment

Abstract DPLL

Solution: Abstract DPLL

[NOT'06]

" ... a **uniform**, **declarative framework** for describing **DPLL-**based solvers, both for **propositional** satisfiability and for satisfiability **modulo theories**.

(emphasis is mine)

Abstract DPLL

Notions and Classical DPLL

Formal Preliminaries (1/3)
Propositional Case

$p \in P$ **set** of propositions (**atoms**)
(in examples this set will be identified with natural numbers)

l is a **literal** whenever l is p or $\neg p$

$\neg l = \texttt{if } l = p \texttt{ then } \neg p \texttt{ else } p$ **negation** op. on **literal**

$C = l_1 \vee \cdots \vee l_n$ **clause** is a **set** of **literals**

$F = C_1, \ldots, C_m$ **formula** is a **set** of **clauses**

F is in **CNF**

Formal Preliminaries (2/3)
Propositional Case

p atom
l literal
$C = l_1 \vee \cdots \vee l_n$
$F = C_1, \ldots, C_m$

$M = l_1 l_2 \ldots l_n$ **assignment** is a **consistent sequence** of **literals**

M is **consistent** if there is no p s.t. $\neg p \in M$ and $p \in M$

l is **defined in** M if $l \in M$ or $\neg l \in M$

$M \models C$ if there is $l \in C$ s.t. $l \in M$

conflict $M \models \neg C$ if for every $l \in C$ implies $\neg l \in M$

$M \models F$ if for every $C \in F$ implies $M \models C$

$\left\{ \begin{array}{l} M \text{ is a } \textbf{model} \text{ of } F \\ F \text{ is } \textbf{satisfiable} \end{array} \right\}$ whenever for some M $\quad M \models F$

$F \models F'$ if for every $M \models F$ implies $M \models F'$

Abstract DPLL System (

Transition System

p atom

l literal

$C = l_1 \vee \cdots \vee l_n$ clause

$F = C_1, \ldots, C_m$ CNF

$M = l_1 l_2 \ldots l_n$ model

Transition system $\langle \Gamma, \Longrightarrow \rangle$ where $\Longrightarrow \in \Gamma \times \Gamma$

State $S \in \Gamma$ in case of ADPLL is one of

1. $FailState$
2. $M \,||\, F$

Transition $S \Longrightarrow S'$

Final state S wrt $\Longrightarrow$ whenever $S \not\Longrightarrow$

Reflexive-transitive closure $\Longrightarrow^*$ of $\Longrightarrow$

Decision literal l^d in $M = N l^d N'$

Classical DPLL system $\Longrightarrow_{Cl}$

p atom
l literal
$C = l_1 \vee \cdots \vee l_n$ clause
$F = C_1, \ldots, C_m$ CNF
$M = l_1 l_2 \ldots l_n$ model

UnitPropagate:

$$M \parallel F,\, C \vee l \;\Longrightarrow\; M\; l \parallel F,\, C \vee l \quad \textbf{if} \begin{cases} M \models \neg C \\ l \text{ is undefined in } M. \end{cases}$$

PureLiteral:

$$M \parallel F \;\Longrightarrow\; M\; l \parallel F \quad \textbf{if} \begin{cases} l \text{ occurs in some clause of } F \\ \neg l \text{ occurs in no clause of } F \\ l \text{ is undefined in } M. \end{cases}$$

Decide:

$$M \parallel F \;\Longrightarrow\; M\; l^{\mathsf{d}} \parallel F \quad \textbf{if} \begin{cases} l \text{ or } \neg l \text{ occurs in a clause of } F \\ l \text{ is undefined in } M. \end{cases}$$

Fail:

$$M \parallel F,\, C \;\Longrightarrow\; \mathit{FailState} \quad \textbf{if} \begin{cases} M \models \neg C \\ M \text{ contains no decision literals.} \end{cases}$$

Backtrack:

$$M\; l^{\mathsf{d}}\; N \parallel F,\, C \;\Longrightarrow\; M\; \neg l \parallel F,\, C \quad \textbf{if} \begin{cases} M\; l^{\mathsf{d}}\; N \models \neg C \\ N \text{ contains no decision literals.} \end{cases}$$

Classical DPLL system $\Longrightarrow Cl$

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

UnitPropagate :

$$M \parallel F, C \vee l \quad \Longrightarrow \quad M\ l \parallel F, C \vee l \quad \textbf{if} \begin{cases} M \models \neg C \\ l \text{ is undefined in } M. \end{cases}$$

No order (strategy) of rule application **prescribed.**

in some clause of F
rs in no clause of F
fined in M.

Decide :

$$M \parallel F \quad \Longrightarrow \quad M\ l^{\mathsf{d}} \parallel F \quad \textbf{if} \begin{cases} l \text{ or } \neg l \text{ occurs in a clause of } F \\ l \text{ is undefined in } M. \end{cases}$$

Fail :

$$M \parallel F, C \quad \Longrightarrow \quad \mathit{FailState} \quad \textbf{if} \begin{cases} M \models \neg C \\ M \text{ contains no decision literals.} \end{cases}$$

Backtrack :

$$M\ l^{\mathsf{d}}\ N \parallel F, C \quad \Longrightarrow \quad M\ \neg l \parallel F, C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C \\ N \text{ contains no decision literals.} \end{cases}$$

Classical DPLL system $\Longrightarrow_{Cl}$

p atom
l literal
$C = l_1 \vee \cdots \vee l_n$ clause
$F = C_1, \ldots, C_m$ CNF
$M = l_1 l_2 \ldots l_n$ model

UnitPropagate:

$$M \parallel F, C \vee l \Longrightarrow M\, l \parallel F, C \vee l \quad \textbf{if} \begin{cases} M \models \neg C \\ l \text{ is undefined in } M. \end{cases}$$

No order (strategy) of rule application **prescribed.**

... in some clause of F
...s in no clause of F
...fined in M.

Decide:

$$M \parallel F \Longrightarrow M\, l^{\mathsf{d}} \parallel F \quad \textbf{if} \begin{cases} l \text{ or } \neg l \text{ occurs in a clause of } F \\ \ldots \end{cases}$$

Fail:

$M \parallel F, C$

These rules produce **models** which are **total**, i.e, every atom of the formula appears in the model.

Backtrack:

$$M\, l^{\mathsf{d}}\, N \parallel F, C \Longrightarrow M\, \neg l \parallel F, C \quad \textbf{if} \begin{cases} M\, l^{\mathsf{d}}\, N \models \neg C \\ N \text{ contains no decision literals.} \end{cases}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$\overbrace{\qquad}^{M} \quad \overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{F}$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$\overbrace{\quad\quad\emptyset}^{M} \;\|\; \overbrace{\overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{ (Decide)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

1 is undefined in $\emptyset$

$$\overbrace{\qquad\emptyset}^{M} \;\|\; \overbrace{\overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{ (Decide)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$\overbrace{\emptyset}^{M} \parallel \overbrace{\overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \ \text{(Decide)}$$

$$1^{\mathsf{d}} \parallel \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$\overline{2}$ is undefined in 1^d and $1^d \models \neg\overline{1}$

$$\underbrace{\emptyset}_{M} \parallel \underbrace{\overline{1} \vee \overline{2},\ \ 2 \vee 3,\ \ \overline{1} \vee \overline{3} \vee 4,\ \ 2 \vee \overline{3} \vee \overline{4},\ \ 1 \vee 4}_{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{(Decide)}$$

$$1^{\mathrm{d}} \parallel \overline{1} \vee \overline{2},\ \ 2 \vee 3,\ \ \overline{1} \vee \overline{3} \vee 4,\ \ 2 \vee \overline{3} \vee \overline{4},\ \ 1 \vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$\overbrace{\emptyset}^{M} \parallel \overbrace{\overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \quad \text{(Decide)}$$

$$1^{\mathsf{d}} \parallel \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \quad \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\ \overline{2} \parallel \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \quad \text{(UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

3 is undefined in $1^d\overline{2}$ and $1^d\overline{2} \models \neg 2$

$$\overbrace{\emptyset}^{M} \parallel \overbrace{\overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4}^{F} \quad \Longrightarrow_{\text{Cl}} \text{(Decide)}$$

$$1^{\mathsf{d}} \parallel \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\text{Cl}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\,\overline{2} \parallel \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\text{Cl}} \text{(UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$\overbrace{\emptyset}^{M} \parallel \overbrace{\overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{(Decide)}$$

$$1^{\mathsf{d}} \parallel \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\,\overline{2} \parallel \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\,\overline{2}\,3 \parallel \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

4 is undefined in $1^d\overline{2}3$ and $1^d\overline{2}3 \models \neg(\overline{1} \vee \overline{3})$

$$\overbrace{\emptyset}^{M} \;\|\; \overbrace{\overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{ (Decide)}$$

$$1^{\mathsf{d}} \;\|\; \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{ (UnitPropagate)}$$

$$1^{\mathsf{d}}\ \overline{2} \;\|\; \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{ (UnitPropagate)}$$

$$1^{\mathsf{d}}\ \overline{2}\ 3 \;\|\; \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{ (UnitPropagate)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$
\begin{array}{rcll}
\overbrace{\emptyset}^{M} & \| & \overbrace{\overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4}^{F} & \Longrightarrow_{\mathrm{Cl}} \text{(Decide)} \\
1^{\mathsf{d}} & \| & \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2} & \| & \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3 & \| & \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3\ 4 & \| & \overline{1}\vee\overline{2},\ \ 2\vee 3,\ \ \overline{1}\vee\overline{3}\vee 4,\ \ 2\vee\overline{3}\vee\overline{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(Backtrack)}
\end{array}
$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$1^d\overline{2}34 \models \neg(2 \vee \overline{3} \vee \overline{4})$$

$$\overbrace{\qquad\emptyset}^{M} \parallel \overbrace{\overline{1}\vee\overline{2},\ 2\vee 3,\ \overline{1}\vee\overline{3}\vee 4,\ 2\vee\overline{3}\vee\overline{4},\ 1\vee 4}^{F} \quad \Longrightarrow_{\mathrm{Cl}} \text{(Decide)}$$

$$1^d \parallel \overline{1}\vee\overline{2},\ 2\vee 3,\ \overline{1}\vee\overline{3}\vee 4,\ 2\vee\overline{3}\vee\overline{4},\ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

$$1^d\ \overline{2} \parallel \overline{1}\vee\overline{2},\ 2\vee 3,\ \overline{1}\vee\overline{3}\vee 4,\ 2\vee\overline{3}\vee\overline{4},\ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

$$1^d\ \overline{2}\ 3 \parallel \overline{1}\vee\overline{2},\ 2\vee 3,\ \overline{1}\vee\overline{3}\vee 4,\ 2\vee\overline{3}\vee\overline{4},\ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}$$

$$1^d\ \overline{2}\ 3\ 4 \parallel \overline{1}\vee\overline{2},\ 2\vee 3,\ \overline{1}\vee\overline{3}\vee 4,\ \underline{2\vee\overline{3}\vee\overline{4},}\ 1\vee 4 \quad \Longrightarrow_{\mathrm{Cl}} \text{(Backtrack)}$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$
\begin{array}{rcll}
\overbrace{\emptyset}^{M} & \| & \overbrace{\overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4}^{F} & \Longrightarrow_{\mathrm{Cl}} \text{(Decide)} \\
1^{\mathsf{d}} & \| & \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2} & \| & \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3 & \| & \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3\ 4 & \| & \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad \underline{2\vee\overline{3}\vee\overline{4},}\quad 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(Backtrack)} \\
\overline{1} & \| & \overline{1}\vee\overline{2},\quad 2\vee 3,\quad \overline{1}\vee\overline{3}\vee 4,\quad 2\vee\overline{3}\vee\overline{4},\quad 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}
\end{array}
$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$
\begin{array}{rcl}
\overbrace{\emptyset}^{M} \parallel \overbrace{\overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4}^{F} & \Longrightarrow_{\mathrm{Cl}} & \text{(Decide)} \\
1^{\mathsf{d}} \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2} \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3 \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3\ 4 \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ \underline{2\lor\overline{3}\lor\overline{4},}\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(Backtrack)} \\
\overline{1} \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
\overline{1}\ 4 \parallel \overline{1}\lor\overline{2},\ \ 2\lor 3,\ \ \overline{1}\lor\overline{3}\lor 4,\ \ 2\lor\overline{3}\lor\overline{4},\ \ 1\lor 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(Decide)}
\end{array}
$$

Classical DPLL Example

```
Strategy: apply Decide if other rules don't apply
```

$$
\begin{array}{rcll}
\overbrace{\emptyset}^{M} & \| & \overbrace{\overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4}^{F} & \Longrightarrow_{\mathrm{Cl}} \text{(Decide)} \\
1^{\mathsf{d}} & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2} & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3 & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \overline{2}\ 3\ 4 & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; \underline{2\vee\overline{3}\vee\overline{4}},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(Backtrack)} \\
\overline{1} & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)} \\
\overline{1}\ 4 & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(Decide)} \\
\overline{1}\ 4\ \overline{3}^{\mathsf{d}} & \| & \overline{1}\vee\overline{2},\;\; 2\vee 3,\;\; \overline{1}\vee\overline{3}\vee 4,\;\; 2\vee\overline{3}\vee\overline{4},\;\; 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \text{(UnitPropagate)}
\end{array}
$$

Classical DPLL Example

```
Strategy: apply Decide if other rules don't apply
```

$$
\begin{array}{rcl ll}
\overbrace{\qquad\qquad}^{M} & & \overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{F} & & \\
\emptyset & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(Decide)} \\
1^{\mathrm{d}} & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathrm{d}}\ \bar{2} & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathrm{d}}\ \bar{2}\ 3 & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
1^{\mathrm{d}}\ \bar{2}\ 3\ 4 & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ \underline{2\vee\bar{3}\vee\bar{4},}\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(Backtrack)} \\
\bar{1} & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
\bar{1}\ 4 & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(Decide)} \\
\bar{1}\ 4\ \bar{3}^{\mathrm{d}} & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} & \text{(UnitPropagate)} \\
\bar{1}\ 4\ \bar{3}^{\mathrm{d}}\ 2 & \| & \bar{1}\vee\bar{2},\ \ 2\vee 3,\ \ \bar{1}\vee\bar{3}\vee 4,\ \ 2\vee\bar{3}\vee\bar{4},\ \ 1\vee 4 & \not\Longrightarrow_{\mathrm{Cl}} &
\end{array}
$$

Classical DPLL Example

Strategy: apply Decide if other rules don't apply

$$
\begin{array}{rcllllll}
\overbrace{\qquad\qquad}^{M} & & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{F}} & \\
\emptyset & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(Decide)} \\
1^{\mathsf{d}} & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \bar{2} & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \bar{2}\ 3 & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ \bar{2}\ 3\ 4 & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & \underline{2\vee\bar{3}\vee\bar{4},} & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(Backtrack)} \\
\bar{1} & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)} \\
\bar{1}\ 4 & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(Decide)} \\
\bar{1}\ 4\ \bar{3}^{\mathsf{d}} & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \Longrightarrow_{\mathrm{Cl}} \ \text{(UnitPropagate)} \\
\bar{1}\ 4\ \bar{3}^{\mathsf{d}}\ 2 & \| & \bar{1}\vee\bar{2}, & 2\vee 3, & \bar{1}\vee\bar{3}\vee 4, & 2\vee\bar{3}\vee\bar{4}, & 1\vee 4 & \not\Longrightarrow_{\mathrm{Cl}}
\end{array}
$$

Final state wrt $\Longrightarrow_{\mathrm{Cl}}$ and thus $M \models F$

Extensions To DPLL

Abstract DPLL + Backjump + Learning + Restart

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$
\begin{array}{rcll}
\emptyset & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{\mathsf{d}} & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2 & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}} & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4 & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}} & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}}\ \overline{6} & \| & \overline{1}\vee 2,\ \ \overline{3}\vee 4,\ \ \overline{5}\vee\overline{6},\ \ 6\vee\overline{5}\vee\overline{2} &
\end{array}
$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$\begin{array}{rcll}
\emptyset & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(Decide)} \\
1^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{B} \text{(UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}}\ \bar{6} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ \underline{6 \vee \bar{5} \vee \bar{2}} &
\end{array}$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$
\begin{array}{rcll}
\emptyset & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{d} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{d}\ 2 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{d}\ 2\ 3^{d} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{d}\ 2\ 3^{d}\ 4 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(Decide)} \\
1^{d}\ 2\ 3^{d}\ 4\ 5^{d} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)} \\
1^{d}\ 2\ 3^{d}\ 4\ 5^{d}\ \bar{6} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ \underline{6 \vee \bar{5} \vee \bar{2}} &
\end{array}
$$

neg

$$\Longrightarrow_{\mathrm{B}} \text{(Backtrack)}$$

$$1^{d}\ 2\ 3^{d}\ 4\ \bar{5}\ \|\ \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2}$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$\emptyset \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{d} \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{d}\ 2 \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{d}\ 2\ 3^{d} \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{d}\ 2\ 3^{d}\ 4 \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{d}\ 2\ 3^{d}\ 4\ 5^{d} \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{d}\ 2\ 3^{d}\ 4\ 5^{d}\ \overline{6} \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad \underline{6\vee\overline{5}\vee\overline{2}}$$

neg

$$\Longrightarrow_{\mathrm{B}} \text{(Backtrack)}$$

$$1^{d}\ 2\ 3^{d}\ 4\ \overline{5} \parallel \overline{1}\vee 2,\quad \overline{3}\vee 4,\quad \overline{5}\vee\overline{6},\quad 6\vee\overline{5}\vee\overline{2}$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$\emptyset \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(Decide)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{(UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}}\ \bar{6} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

neg

$$\Longrightarrow_{\mathrm{B}} \text{(Backtrack)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ \bar{5} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

Non Chronological Backtracking

`(Informal)` *Example* 2.3.

$$\emptyset \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (Decide)}$$

$$1^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (Decide)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (Decide)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \quad \Longrightarrow_{\mathrm{B}} \text{ (UnitPropagate)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}}\ \bar{6} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

neg

$$\Longrightarrow_{\mathrm{B}} \text{ (Backtrack)}$$

$$1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ \bar{5} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$
\begin{array}{rcll}
\emptyset & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (Decide)} \\
1^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (UnitPropagate)} \\
1^{\mathsf{d}}\ 2 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4 & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (Decide)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} & \Longrightarrow_B \text{ (UnitPropagate)} \\
1^{\mathsf{d}}\ 2\ 3^{\mathsf{d}}\ 4\ 5^{\mathsf{d}}\ \bar{6} & \| & \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2} &
\end{array}
$$

$$\Longrightarrow$$

$$1^{\mathrm{d}} 2 \bar{5} \ \| \ \bar{1} \vee 2,\ \ \bar{3} \vee 4,\ \ \bar{5} \vee \bar{6},\ \ 6 \vee \bar{5} \vee \bar{2}$$

Non Chronological Backtracking

(Informal) *Example* 2.3.

$$\emptyset \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

$$1^d \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

$$1^d\ 2 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

$$1^d\ 2\ 3^d \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \qquad \Longrightarrow_{\mathrm{B}}$$

$$1^d\ 2\ 3^d\ 4 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \qquad \Longrightarrow_{\mathrm{B}} \text{ (Decide)}$$

$$1^d\ 2\ 3^d\ 4\ 5^d \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2} \qquad \Longrightarrow_{\mathrm{B}} \text{ (UnitPropagate)}$$

$$1^d\ 2\ 3^d\ 4\ 5^d\ 6 \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

5 and 1
are **incompatible**
or consequently
5 and 2
are **incompatible**
we **learn** that

$$\overline{5\wedge 2} = \bar{5}\vee\bar{2}$$

$$\Longrightarrow$$

$$1^d 2\bar{5} \parallel \bar{1}\vee 2,\ \ \bar{3}\vee 4,\ \ \bar{5}\vee\bar{6},\ \ 6\vee\bar{5}\vee\bar{2}$$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

Conflict Driven Backtracking

l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

Definition 2.4.

UnitPropagate + Decide + Fail

+

Backjump :

$$M\ l^{\mathsf{d}}\ N \parallel F,\, C \quad \Longrightarrow \quad M\ l' \parallel F,\, C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

C **conflict clause**

$C' \vee l'$ **backjump clause**
unit clause wrt M
satisfiable under the **same models** as **F,C**

Backtracks further than backtrack

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$
Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$$M\ l^{\mathsf{d}}\ N \parallel F,\ C \quad \Longrightarrow \quad M\ l' \parallel F,\ C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$
Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$$M\ l^{\mathsf{d}}\ N \,\|\, F,\, C \quad \Longrightarrow \quad M\ l' \,\|\, F,\, C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M\text{, and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Suppose

$$\overbrace{M_0 l_1^{\mathrm{d}} M_1 \ldots}^{M} l_n^{\mathrm{d}} \overbrace{M_n}^{N} \;||\; F, C$$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$$M\ l^{\mathsf{d}}\ N \,\|\, F,\, C \;\Longrightarrow\; M\ l' \,\|\, F,\, C \quad \textbf{if} \quad \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Suppose

$$\overbrace{M_0 l_1^{\mathrm{d}} M_1 \ldots}^{M} l_n^{\mathrm{d}} \overbrace{M_n}^{N} \;||\; F, C$$

Take $C' \vee l' = \neg l_1 \vee \cdots \vee \neg l_n$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$$M\ l^{\mathsf{d}}\ N \parallel F,\ C \ \Longrightarrow\ M\ l' \parallel F,\ C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Suppose

$$\overbrace{M_0 l_1^{\mathrm{d}} M_1 \ldots}^{M} l_n^{\mathrm{d}} \overbrace{M_n}^{N} \,||\, F, C$$

Take $C' \vee l' = \neg l_1 \vee \cdots \vee \neg l_n$

Have $M \models \neg(\neg l_1 \vee \cdots \vee \neg l_{n-1}) \iff M \models l_1, \ldots, l_{n-1}$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$$M\ l^{\mathsf{d}}\ N \parallel F,\ C \implies M\ l' \parallel F,\ C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Suppose

$$\overbrace{M_0 l_1^{\mathrm{d}} M_1 \ldots}^{M} l_n^{\mathrm{d}} \overbrace{M_n}^{N} \mid\mid F, C$$

Take $C' \vee l' = \neg l_1 \vee \cdots \vee \neg l_n$

Have $M \models \neg(\neg l_1 \vee \cdots \vee \neg l_{n-1}) \iff M \models l_1, \ldots, l_{n-1}$

And $F, C, l_1, \ldots, l_n$ is **unsat** $\iff F, C \models \neg l_1 \vee \cdots \vee \neg l_n$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

Conflict Driven Backtracking

p atom
l literal
$C = l_1 \vee \cdots \vee l_n$ clause
$F = C_1, \ldots, C_m$ CNF
$M = l_1 l_2 \ldots l_n$ model

$$M\ l^{\mathsf{d}}\ N \parallel F,\ C \implies M\ l' \parallel F,\ C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Sanity Check: **Modeling Backtrack with Backjump**

Suppose

Thus

$$\overbrace{M_0 l_1^{\mathrm{d}} M_1 \ldots l_n^{\mathrm{d}}}^{M} \overbrace{M_n}^{N} \,||\, F, C \quad \Longrightarrow \quad M_0 l_1^{\mathrm{d}} M_1 \ldots \neg l_n \,||\, F, C$$

Take $C' \vee l' = \neg l_1 \vee \cdots \vee \neg l_n$

Have $M \models \neg(\neg l_1 \vee \cdots \vee \neg l_{n-1}) \iff M \models l_1, \ldots, l_{n-1}$

And $F, C, l_1, \ldots, l_n$ is **unsat** $\iff F, C \models \neg l_1 \vee \cdots \vee \neg l_n$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$
Conflict Driven Backtracking

p **atom**
l **literal**
$C = l_1 \vee \cdots \vee l_n$ **clause**
$F = C_1, \ldots, C_m$ **CNF**
$M = l_1 l_2 \ldots l_n$ **model**

$M\ l^{\mathsf{d}}\ N \models \neg C,$ and there is

Backjump is a backtracking mechanism.

LEMMA 2.8. *Assume that $\emptyset \parallel F \Longrightarrow^*_{\mathrm{L}} M \parallel F'$ and that $M \models \neg C$ for some clause C in F'. Then either Fail or Backjump applies to $M \parallel F'$.*

This follows by a bit more generalized construction of the presented one.

S

$$M_0 l_1^{\mathrm{d}} M_1 \ldots l_n^{\mathrm{d}} M_n \parallel F, C \quad \Longrightarrow \quad M_0 l_1^{\mathrm{d}} M_1 \ldots \neg l_n \parallel F, C$$

Take $C' \vee l' = \neg l_1 \vee \cdots \vee \neg l_n$

Have $M \vDash \neg(\neg l_1 \vee \cdots \vee \neg l_{n-1}) \iff M \vDash l_1, \ldots, l_{n-1}$

And $F, C, l_1, \ldots, l_n$ is **unsat** $\iff F, C \vDash \neg l_1 \vee \cdots \vee \neg l_n$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

with **Backjump**

Sound and **Complete**

THEOREM 2.13. *If* $\emptyset \parallel F \Longrightarrow_{\mathrm{B}}^{*} S$ *where* S *is final with respect to Basic DPLL, then*

(1) S *is FailState if, and only if,* F *is unsatisfiable.*

(2) *If* S *is of the form* $M \parallel F'$, *then* M *is a model of* F. **As** $F' = F$

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

with **Backjump**

Terminates

THEOREM 2.10. *There are no infinite derivations of the form* $\emptyset \parallel F \Longrightarrow_{\mathrm{B}} S_1 \Longrightarrow_{\mathrm{B}} \cdots$.

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

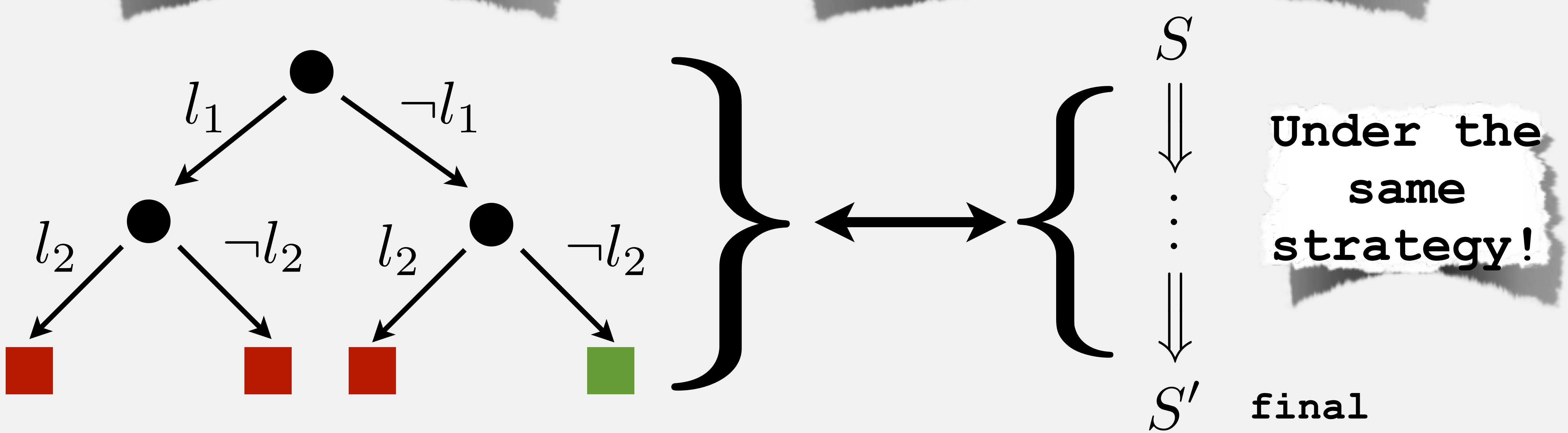
with Backjump

Terminates

THEOREM 2.10. *There are no infinite derivations of the form* $\emptyset \,\|\, F \Longrightarrow_{\mathrm{B}} S_1 \Longrightarrow_{\mathrm{B}} \cdots$.

Why?

Usual DPLL | **Abstract DPLL**



Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

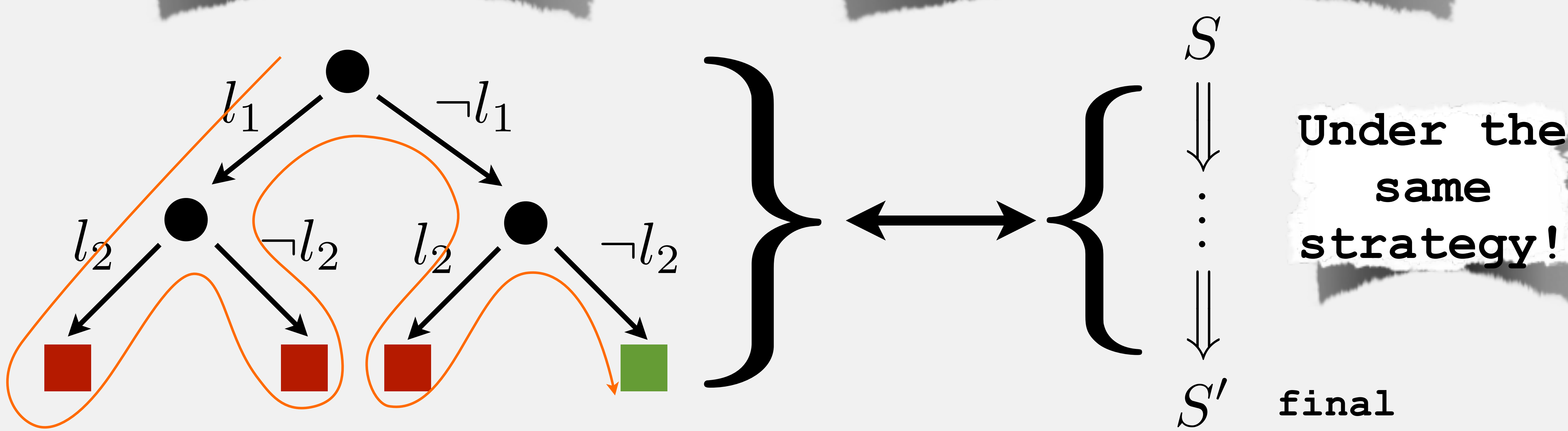
with Backjump

Terminates

THEOREM 2.10. *There are no infinite derivations of the form* $\emptyset \parallel F \Longrightarrow_{\mathrm{B}} S_1 \Longrightarrow_{\mathrm{B}} \cdots$.

Why?

Usual DPLL **Abstract DPLL**



Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

with **Backjump**

Terminates

THEOREM 2.10. *There are no infinite derivations of the form $\emptyset \parallel F \Longrightarrow_{\mathrm{B}} S_1 \Longrightarrow_{\mathrm{B}} \cdots$.*

Search **Progress**

progress at **decision** level ***i***
=
number of literals at **dl *i***

$$M_0 l_1^{\mathrm{d}} M_1 l_2 \ldots l_n^{\mathrm{d}} M_n \,||\, F$$

For $M||F \Longrightarrow M'||F'$

$M'||F'$ is more **progressed** than $M||F$

if there is a **least** more progressed decision level, or the **model** is more progressed.

Basic DPLL system $\Longrightarrow_{\mathrm{B}}$

with **Backjump**

Terminates

THEOREM 2.10. *There are no infinite derivations of the form* $\emptyset \,\|\, F \Longrightarrow_{\mathrm{B}} S_1 \Longrightarrow_{\mathrm{B}} \cdots$.

Search **Progress**

progress at **decision** level ***i***
=
number of literals at **dl** ***i***

$$M_0 l_1^{\mathrm{d}} M_1 l_2 \ldots l_n^{\mathrm{d}} M_n \,||\, F$$

For $M||F \Longrightarrow$

$M'||F'$ is more **progressed**

if there is a **leas**

level, or the **model** is more progressed.

Rules progress search up to the number of atoms of the formula.

Conflict Driven Clause Learning
Example (Conflict Graph)

Consider $M||F$ where

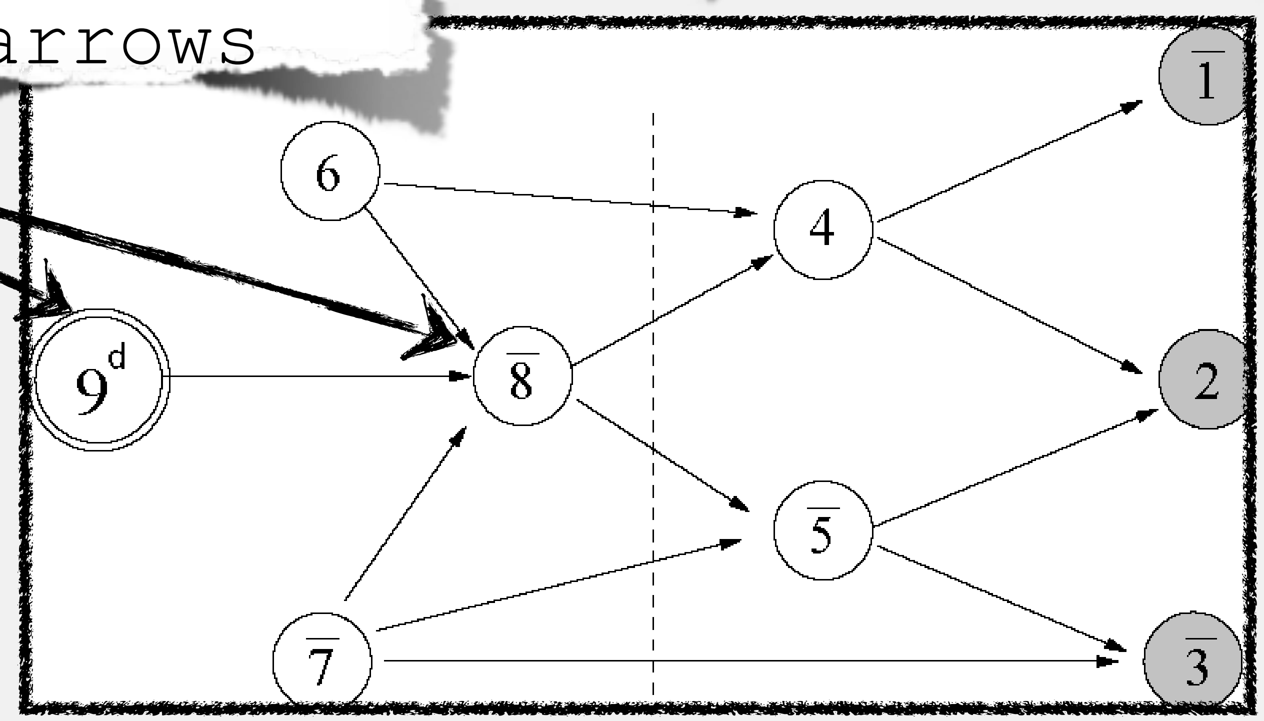
$$F = \ldots \overline{9} \vee \overline{6} \vee 7 \vee \overline{8} \quad 8 \vee 7 \vee \overline{5} \quad \overline{6} \vee 8 \vee 4 \quad \overline{4} \vee \overline{1} \quad \overline{4} \vee 5 \vee 2 \quad 5 \vee 7 \vee \overline{3} \quad \underline{1 \vee \overline{2} \vee 3}$$

$$M = \ldots 6 \ldots \overline{7} \ldots 9^{d}\ \overline{8}\ \overline{5}\ 4\ \overline{1}\ 2\ \overline{3}.$$

No outgoing arrows

No incoming arrows

Unique Implication Point



Conflict Driven Clause Learning
Example (Conflict Graph)

Consider $M||F$ where

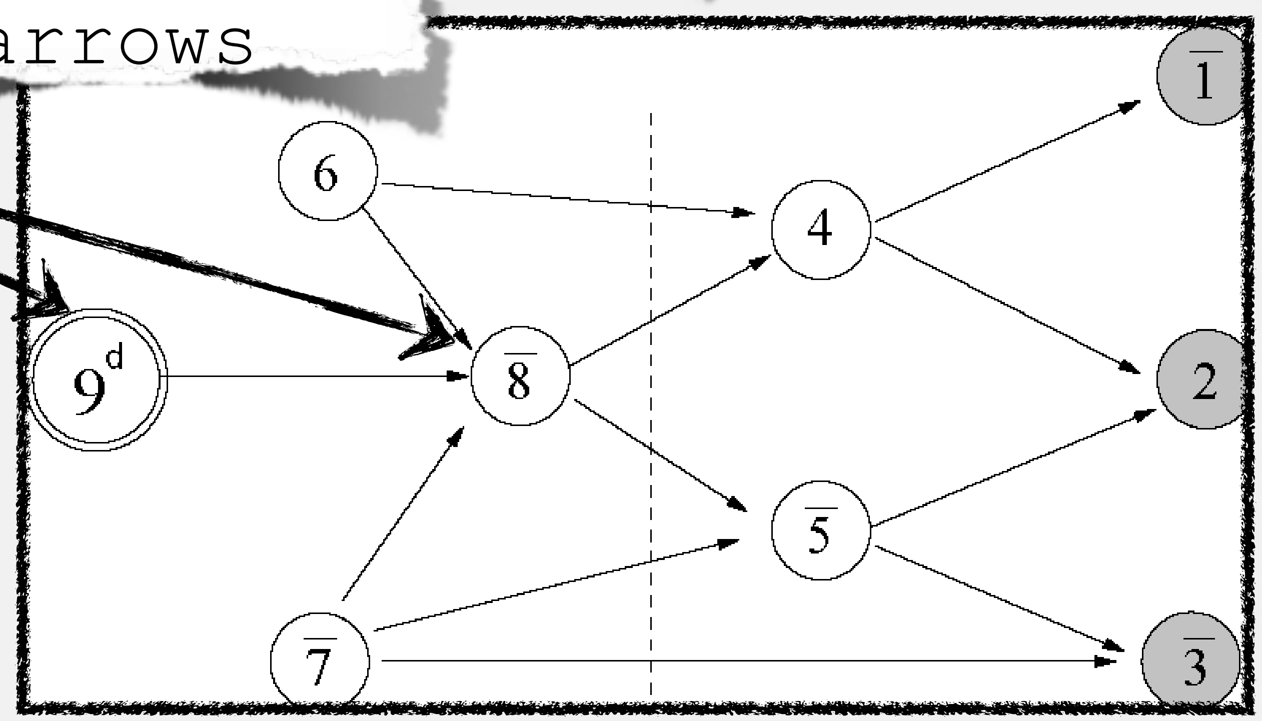
$$F = \ldots \overline{9} \vee \overline{6} \vee 7 \vee \overline{8} \quad 8 \vee 7 \vee \overline{5} \quad \overline{6} \vee 8 \vee 4 \quad \overline{4} \vee \overline{1} \quad \overline{4} \vee 5 \vee 2 \quad 5 \vee 7 \vee \overline{3} \quad 1 \vee \overline{2} \vee 3$$

$$M = \ldots 6 \ldots \overline{7} \ldots 9^d \, \overline{8} \, \overline{5} \, 4 \, \overline{1} \, 2 \, \overline{3}.$$

No outgoing arrows

No incoming arrows

Unique Implication Point



Conflict Driven Clause Learning
Example (Conflict Graph)

Consider $M||F$ where

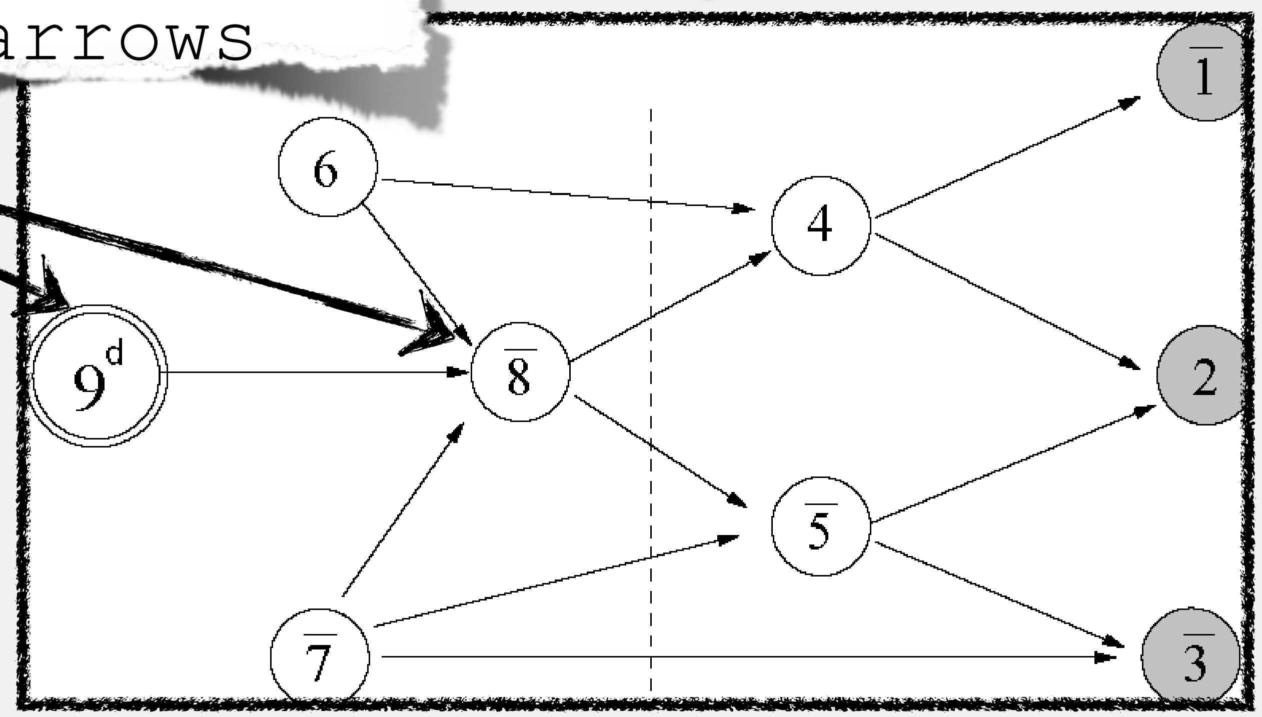
$$F = \ldots \overline{9} \vee \overline{6} \vee 7 \vee \overline{8} \quad 8 \vee 7 \vee \overline{5} \quad \overline{6} \vee 8 \vee 4 \quad \overline{4} \vee \overline{1} \quad \overline{4} \vee 5 \vee 2 \quad 5 \vee 7 \vee \overline{3} \quad 1 \vee \overline{2} \vee 3$$

$$M = \ldots 6 \ldots \overline{7} \ldots 9^d\ \overline{8}\ \overline{5}\ 4\ \overline{1}\ 2\ \overline{3}.$$

No outgoing arrows

No incoming arrows

Unique Implication Point



We can learn $\overline{6} \vee 8 \vee 7$

Conflict Driven Clause Learning Example (Backward Conflict Resolution)

Consider $M||F$ where

$$F = \ldots \bar{9}\vee\bar{6}\vee 7\vee\bar{8} \quad 8\vee 7\vee\bar{5} \quad \bar{6}\vee 8\vee 4 \quad \bar{4}\vee\bar{1} \quad \bar{4}\vee 5\vee 2 \quad 5\vee 7\vee\bar{3} \quad 1\vee\bar{2}\vee 3$$

$$M = \ldots 6 \ldots \bar{7} \ldots 9^d\ \bar{8}\ \bar{5}\ 4\ \bar{1}\ 2\ \bar{3}.$$

in reverse order of

$$\dfrac{8\vee 7\vee\bar{\mathbf{5}} \qquad \dfrac{\bar{6}\vee 8\vee\mathbf{4} \qquad \dfrac{\bar{4}\vee\bar{\mathbf{1}} \qquad \dfrac{\bar{4}\vee 5\vee\mathbf{2} \qquad \dfrac{5\vee 7\vee\bar{\mathbf{3}} \qquad 1\vee\bar{2}\vee\mathbf{3}}{5\vee 7\vee 1\vee\bar{\mathbf{2}}}}{\bar{4}\vee 5\vee 7\vee\mathbf{1}}}{5\vee 7\vee\bar{\mathbf{4}}}}{\bar{6}\vee 8\vee 7\vee\mathbf{5}}}{8\vee 7\vee\bar{6}}$$

until on literal left of the current decision level

Conflict Driven Clause Learning Example (Backward Conflict Resolution)

Consider $M||F$ where

$$F = \ldots \overline{9} \vee \overline{6} \vee 7 \vee \overline{8} \quad 8 \vee 7 \vee \overline{5} \quad \overline{6} \vee 8 \vee 4 \quad \overline{4} \vee \overline{1} \quad \overline{4} \vee 5 \vee 2 \quad 5 \vee 7 \vee \overline{3} \quad 1 \vee \overline{2} \vee 3$$

$$M = \ldots 6 \ldots \overline{7} \ldots 9^{d}\, \overline{8}\, \overline{5}\, 4\, \overline{1}\, 2\, \overline{3}.$$

in reverse order of

$$\cfrac{8 \vee 7 \vee \overline{\mathbf{5}} \qquad \cfrac{\overline{6} \vee 8 \vee \mathbf{4} \qquad \cfrac{\overline{4} \vee \overline{\mathbf{1}} \qquad \cfrac{\overline{4} \vee 5 \vee \mathbf{2} \qquad \cfrac{5 \vee 7 \vee \overline{\mathbf{3}} \qquad 1 \vee \overline{2} \vee \mathbf{3}}{5 \vee 7 \vee 1 \vee \overline{\mathbf{2}}}}{\overline{4} \vee 5 \vee 7 \vee \mathbf{1}}}{5 \vee 7 \vee \overline{\mathbf{4}}}}{\overline{6} \vee 8 \vee 7 \vee \mathbf{5}}}{8 \vee 7 \vee \overline{6}}$$

until on literal left of the current decision level

We can learn $\overline{6} \vee 8 \vee 7$

DPLL System With Learning $\Longrightarrow_L$

Definition 2.5.

Basic DPLL

+

Learn :

$$M \parallel F \quad \Longrightarrow \quad M \parallel F,\, C \quad \textbf{if} \begin{cases} \text{each atom of } C \text{ occurs in } F \text{ or in } M \\ F \models C. \end{cases}$$

Forget :

$$M \parallel F,\, C \quad \Longrightarrow \quad M \parallel F \quad \textbf{if} \begin{cases} F \models C. \end{cases}$$

DPLL System With Learning $\Longrightarrow_L$

Learn :

$$M \parallel F \quad \Longrightarrow \quad M \parallel F,\ C \quad \textbf{if} \begin{cases} \text{each atom of } C \text{ occurs in } F \text{ or in } M \\ F \models C. \end{cases}$$

Forget :

$$M \parallel F,\ C \quad \Longrightarrow \quad M \parallel F \quad \textbf{if} \ \{ F \models C.$$

Sound and **Complete**

THEOREM 2.12. *If* $\emptyset \parallel F \Longrightarrow_{\mathrm{L}}^{*} S$ *where* S *is final with respect to Basic DPLL, then*

(1) *S is FailState if, and only if, F is unsatisfiable.*

(2) *If S is of the form $M \parallel F'$ then M is a model of F.*

DPLL System With Learning $\Longrightarrow_L$

Learn :

$$M \parallel F \quad \Longrightarrow \quad M \parallel F,\, C \quad \textbf{if} \begin{cases} \text{each atom of } C \text{ occurs in } F \text{ or in } M \\ F \models C. \end{cases}$$

Forget :

$$M \parallel F,\, C \quad \Longrightarrow \quad M \parallel F \quad \textbf{if} \left\{ F \models C. \right.$$

Sound and Complete

THEOREM 2.12. *If $\emptyset \parallel F \Longrightarrow_{\mathrm{L}}^{*} S$ where S is final with respect to Basic DPLL, then*

(1) *S is FailState if, and only if, F is unsatisfiable.*

(2) *If S is of the form $M \parallel F'$ then M is a model of F.*

Decidable

THEOREM 2.11. *Every derivation $\emptyset \parallel F \Longrightarrow_{\mathrm{L}} S_1 \Longrightarrow_{\mathrm{L}} \cdots$ by the DPLL system with Learning is finite if it contains no infinite subderivations consisting of only* Learn *and* Forget *steps.*

Restarts

Not enough **progress - restart**

Newly **learned clauses** might help **guide** the **search.**

Definition 2.14. The Restart rule is:

$$M \parallel F \implies \emptyset \parallel F.$$

Restarts

Definition 2.14. The Restart rule is:

$$M \parallel F \implies \emptyset \parallel F.$$

Definition 2.15.

$$S_i \underbrace{\implies \ldots \implies}_{m} S_j \underbrace{\implies \ldots \implies}_{n} S_k$$

where m, n **number** of applications of **Restart**

increased periodicity if $m < n$

Restarts

Definition 2.14. The Restart rule is:

$$M \parallel F \implies \emptyset \parallel F.$$

Definition 2.15.

$$S_i \underbrace{\implies \ldots \implies}_{m} S_j \underbrace{\implies \ldots \implies}_{n} S_k$$

where m, n **number** of applications of **Basic DPLL**

increased periodicity if $m < n$

Abstract DPLL + Restart terminates when considering **paths** with **increasing periodicity** of **Restart**

THEOREM 2.16. *Any derivation* $\emptyset \parallel F \implies S_1 \implies \cdots$ *by the transition system L extended with the* Restart *rule is finite if it contains no infinite subderivations consisting of only* Learn *and* Forget *steps, and* Restart *has increasing periodicity in it.*

SMT

Abstract DPLL Modulo Theories

Formal Preliminaries
First-Order Case

F **ground quantifier free** formula in **CNF**

T **theory** is a **set** of **closed first-order formulas**

F **T-satisfiable** (**T-consistent**) if $F \wedge T$ is **first-order satisfiable**

M is **T-model** of F if $M \models F$ **propositional**

entails $F \models_T F'$ if $F \wedge \neg F'$ **T-inconsistent**

It is assumed $\models_T$ **decidable**

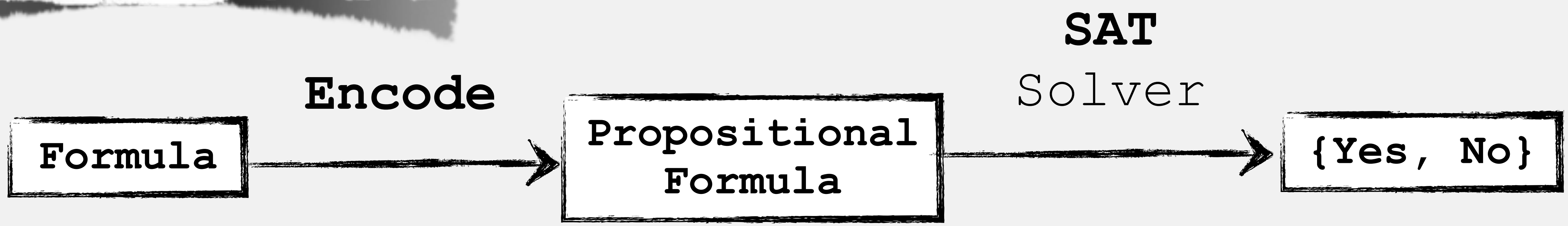
SMT Techniques

Eager SMT

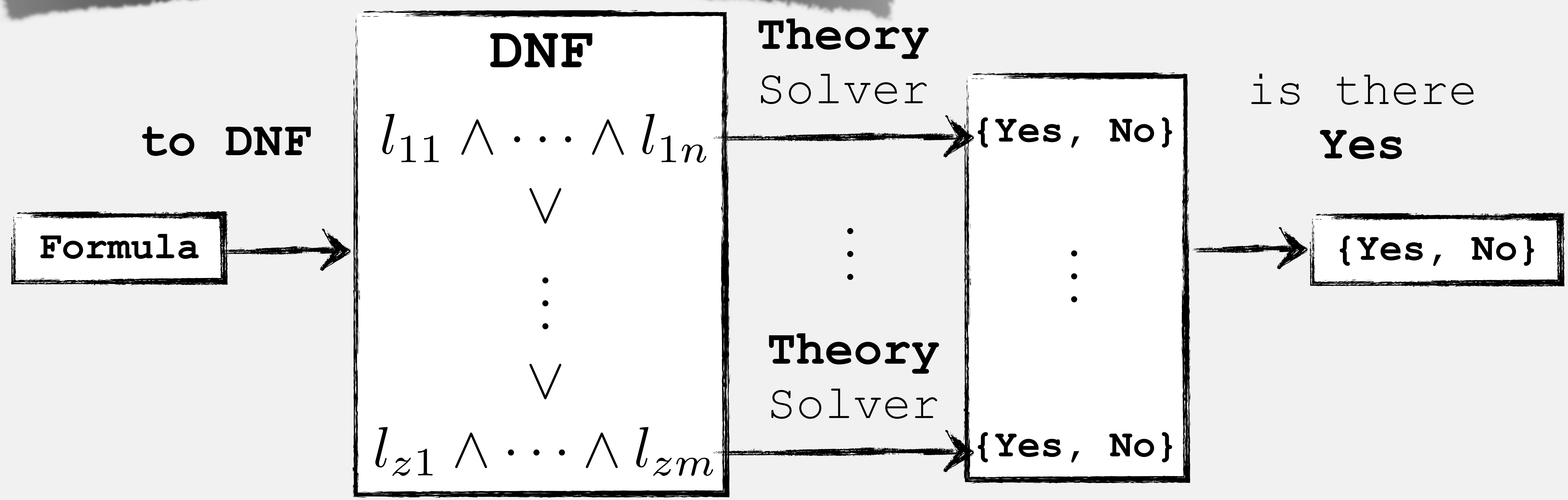
Formula $\xrightarrow{\text{Encode}}$ Propositional Formula $\xrightarrow{\text{SAT Solver}}$ {Yes, No}

SMT Techniques

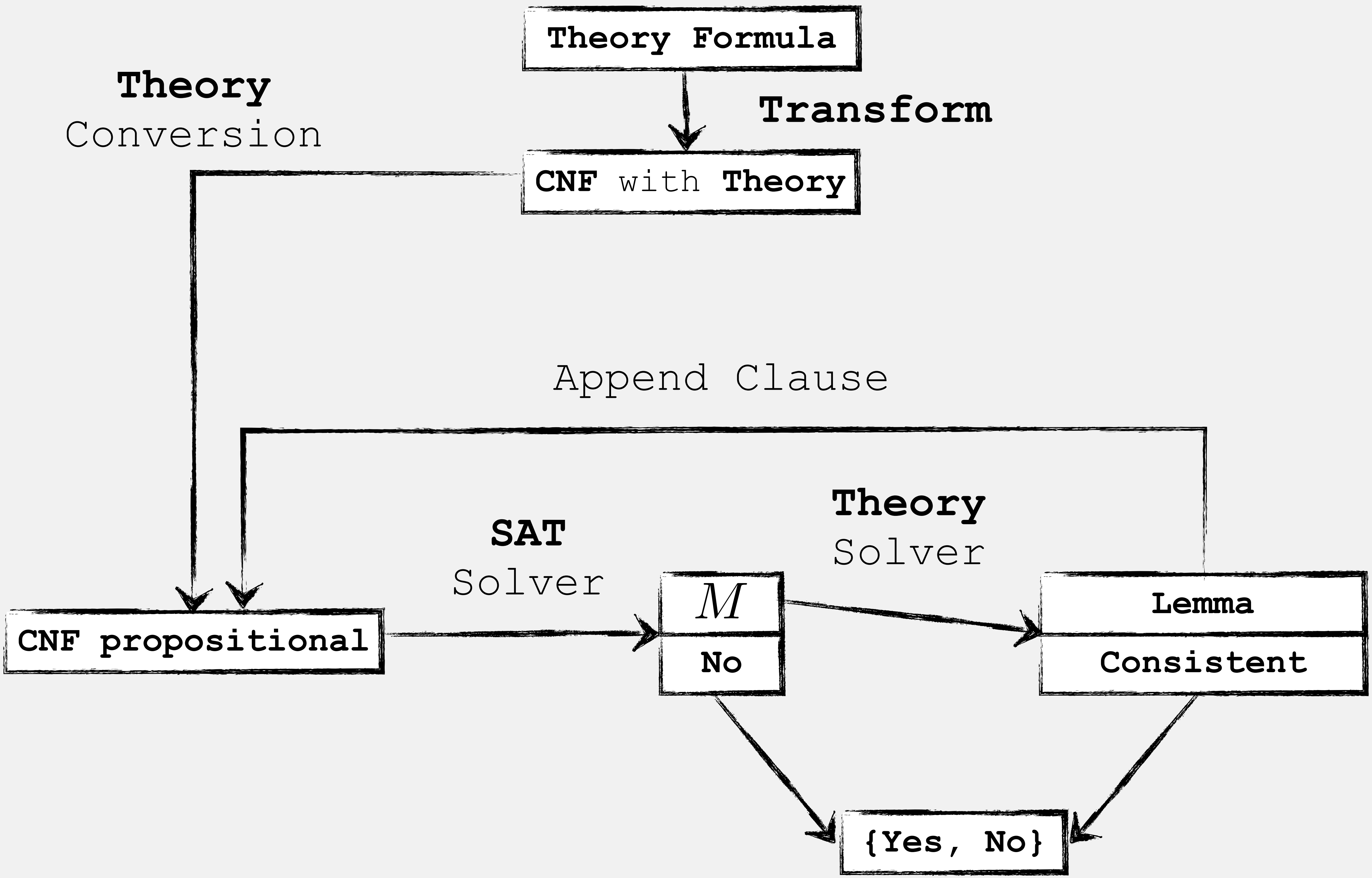
Eager SMT



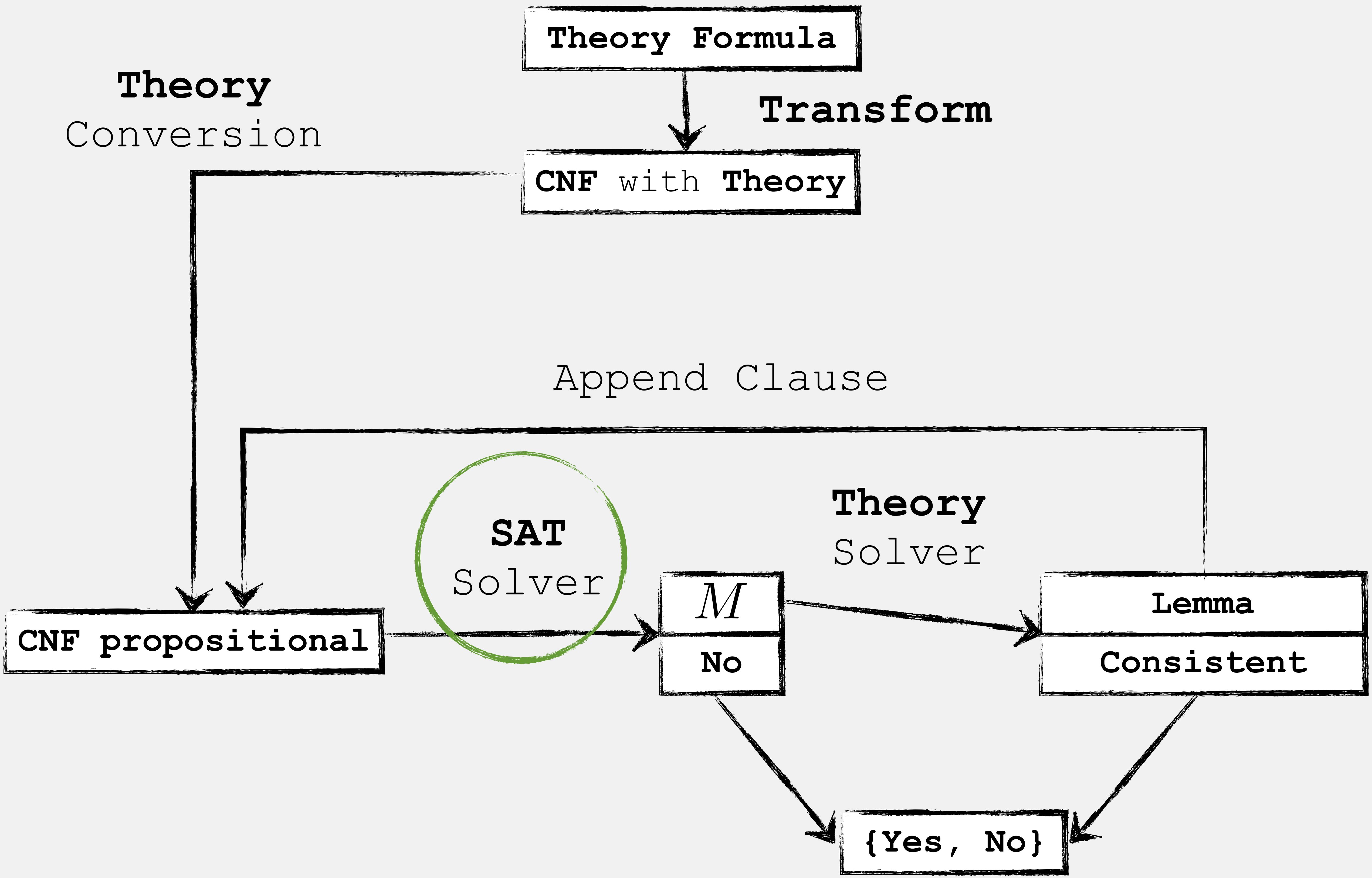
(Very Naive) Lazy SMT



Naive Lazy SMT



Naive Lazy SMT



Lazy SMT based on DPLL

Incremental T-solver

Checks **Theory Consistency** every time a **literal** is **added** to the **assignment.**

On-line SAT Solver

Whenever an **assignment** is **Theory Inconsistent, backtracks** to the point where the theory is still **consistent.**

(Exhaustive) Theory propagation

The **assignment** may be **extended** with the **literals** which are **entailed** by the **theory.**

Abstract DPLL modulo Theories

Definition 3.2.

T-Learn:

$$M \parallel F \quad \Longrightarrow \quad M \parallel F,\ C \quad \textbf{if} \begin{cases} \text{each atom of } C \text{ occurs in } F \text{ or in } M \\ F \models_T C \end{cases}$$

T-Forget:

$$M \parallel F,\ C \quad \Longrightarrow \quad M \parallel F \quad \textbf{if} \begin{cases} F \models_T C \end{cases}$$

T-Backjump:

$$M\ l^{\mathsf{d}}\ N \parallel F,\ C \quad \Longrightarrow \quad M\ l' \parallel F,\ C \quad \textbf{if} \begin{cases} M\ l^{\mathsf{d}}\ N \models \neg C, \text{ and there is} \\ \text{some clause } C' \vee l' \text{ such that:} \\ \quad F, C \models_T C' \vee l' \text{ and } M \models \neg C', \\ \quad l' \text{ is undefined in } M, \text{ and} \\ \quad l' \text{ or } \neg l' \text{ occurs in } F \text{ or in } M\ l^{\mathsf{d}}\ N. \end{cases}$$

Definition 3.3.

TheoryPropagate

$$M \parallel F \quad \Longrightarrow \quad M\ l \parallel F \quad \textbf{if} \begin{cases} M \models_T l \\ l \text{ or } \neg l \text{ occurs in } F \\ l \text{ is undefined in } M. \end{cases}$$

Modeling Lazy SMT with Abstract DPLL

Naive Lazy Approach

$M||F$ is final wrt **Decide**, **Fail**, **UnitPropagate**, **T-Backjump**

If M is **T-incosistent**

There is $\{l_1, \ldots, l_n\} \subseteq M$ s.t. $\emptyset \models_T \neg l_1 \vee \cdots \vee l_n$

$$
\begin{aligned}
M||F \quad &\Longrightarrow_{T-\text{learn}} \quad M||F, \neg l_1 \vee \cdots \vee \neg l_n \\
&\Longrightarrow_{\text{restart}} \quad \emptyset||F, \neg l_1 \vee \cdots \vee \neg l_n
\end{aligned}
$$

Incremental

Same as above, except it applies for any **T-inconsistent** state.

Incremental + Online

Does **not restart**, but uses the fact that learned **lemma** clause is **conflicting** to apply **T-backjump.**

Basic and Full DPLL Modulo Theories system

Definition 3.4. The *Basic DPLL Modulo Theories system* consists of the rules Decide, Fail, UnitPropagate, TheoryPropagate, and T-Backjump.

Definition 3.5. The *Full DPLL Modulo Theories system*, denoted by FT, consists of the rules of Basic DPLL Modulo Theories and the rules T-Learn, T-Forget, and Restart.

Sound and **Complete**

THEOREM 3.10. *Let Der be a derivation* $\emptyset \parallel F \Longrightarrow^{*}_{\mathrm{FT}} S$ *, where (i) S is final with respect to Basic DPLL Modulo Theories, and (ii) if S is of the form* $M \parallel F'$ *then M is T-consistent. Then*

(1) *S is FailState if, and only if, F is T-unsatisfiable.*

(2) *If S is of the form* $M \parallel F'$*, then M is a T-model of F.*

Full DPLL Modulo Theories system

Terminates

THEOREM 3.7 (TERMINATION). *Let Der be a derivation of the form:*
$\emptyset \parallel F = S_0 \Longrightarrow_{\mathrm{FT}} S_1 \Longrightarrow_{\mathrm{FT}} \cdots$
Then Der is finite if the following two conditions hold:

(1) *Der has no infinite subderivations consisting of only T-Learn and T-Forget steps.*

(2) *For every subderivation of Der of the form:*
$S_{i-1} \Longrightarrow_{\mathrm{FT}} S_i \Longrightarrow_{\mathrm{FT}} \cdots \Longrightarrow_{\mathrm{FT}} S_j \Longrightarrow_{\mathrm{FT}} \cdots \Longrightarrow_{\mathrm{FT}} S_k$
where the only three Restart steps are the ones producing S_i, S_j, and S_k, either:
—there are more Basic DPLL Modulo Theories steps in $S_j \Longrightarrow_{\mathrm{FT}} \cdots \Longrightarrow_{\mathrm{FT}} S_k$ than in $S_i \Longrightarrow_{\mathrm{FT}} \cdots \Longrightarrow_{\mathrm{FT}} S_j$, or
—a clause is learned[2] in $S_j \Longrightarrow_{\mathrm{FT}} \cdots \Longrightarrow_{\mathrm{FT}} S_k$ that is not forgotten in Der.

Increasing Periodicity

Do not revisit failed search

DPLL(T)

Abstract DPLL Engine

DPLL(T) Solver

DPLL(X) engine parameterized by a theory solver

$\textbf{Solver}_T$ theory solver for conjunction of formulas

DPLL(T) Solver

DPLL(X) engine parameterized by a theory solver

$Solver_T$ theory solver for conjunction of formulas

$Solver_T$ + DPLL(X) = DPLL(T)

DPLL(T) SM**T** solver

DPLL(T) Architecture

Interface of Solver_T

l : *Literal*

M : *Assignment*

markLitTrue: *Literal* $\longrightarrow$ ***unit***

unmarkLastLits: ***Int*** $\longrightarrow$ **unit**

isConsistent: *Assignment* $\longrightarrow$ *Strength* $\longrightarrow$ **bool**

explainInCons: *Assignment* $\longrightarrow$ *Literal* **set**

Finds and returns $\{l_1, \ldots, l_n\} \subseteq M$ s.t. $\emptyset \models_T \neg l_1 \vee \cdots \vee \neg l_n$

explainTProp: *Assignment* $\longrightarrow$ *Literal* $\longrightarrow$ Literal **set**

Finds and returns

$\{l_1, \ldots, l_n\} \subseteq M$ for a given literal s.t. $l_1, \ldots, l_n \models_T l$

entails: *Assignment* $\longrightarrow$ *Literal* **set** $\longrightarrow$ Literal **set**

Returns $\{\, l \mid M \models_T l \;\&\; l \in L \,\}$

Questions?