The DPLL Procedure

Cesare Tinelli

tinelli@cs.uiowa.edu

The University of Iowa

Propositional Satisfiability: SAT

- Deciding the satisfiability of a propositional formula is a well-studied and important problem.
- Theoretical interest: first established NP-Complete problem, phase transition, ...
- Practical interest: applications to scheduling, planning, logic synthesis, verification, ...
  - Development of algorithms and enhancements.
  - Implementation of extremely efficient tools.
  - Solvers based on the DPLL procedure have been the most successful so far.

The Original DPLL

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

The Original DPLL – Example

Operation	**Assign.**	**Formula**
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	**Assign.**	**Formula**
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1,\ \overline{2},\ 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1,\ \overline{2},\ 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 4	$1,\ \overline{2},\ 3,\ 4$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1,\ \overline{2},\ 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 4	$1,\ \overline{2},\ 3,\ 4$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

Inconsistency!

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1,\ \overline{2},\ 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 4	$1,\ \overline{2},\ 3,\ 4$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
undo 3	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1,\ \overline{2},\ 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 4	$1,\ \overline{2},\ 3,\ 4$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
undo 3	$1,\ \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try $\overline{3}$	$1,\ \overline{2},\ \overline{3}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

The Original DPLL – Example

Operation	Assign.	Formula
		$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 1	1	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce $\overline{2}$	$1, \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try 3	$1, \overline{2}, 3$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
deduce 4	$1, \overline{2}, 3, 4$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
undo 3	$1, \overline{2}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$
try $\overline{3}$	$1, \overline{2}, \overline{3}$	$1 \vee 2,\ 2 \vee \overline{3} \vee 4,\ \overline{1} \vee \overline{2},\ \overline{1} \vee \overline{3} \vee \overline{4},\ 1$

Model Found!

An Abstract Framework for DPLL

- The DPLL procedure can be described declaratively by simple sequent-style calculi.
- Such calculi however cannot model meta-logical features such as backtracking, learning and restarts.
- We model DPLL and its enhancements as transition systems instead.
- A transition system is a binary relation over states, induced by a set of conditional transition rules.

An Abstract Framework for DPLL

Our states:

$$\mathit{fail} \quad \text{or} \quad M \parallel F$$

where F is a CNF formula, a set of clauses, and
M is a sequence of annotated literals
denoting a partial truth assignment.

An Abstract Framework for DPLL

Our states:

$$fail \quad \text{or} \quad M \parallel F$$

Initial state:

- $\emptyset \parallel F$, where F is to be checked for satisfiability.

Expected final states:

- $fail$, if F is unsatisfiable
- $M \parallel G$, where M is a model of G and G is logically equivalent to F.

Transition Rules for the Original DPLL

Extending the assignment:

Propagate

$$M \parallel F, C \vee l \quad \rightarrow \quad M\ l \parallel F, C \vee l \quad \textbf{if} \begin{cases} M \text{ falsifies } C, \\ l \text{ is undefined in } M \end{cases}$$

Transition Rules for the Original DPLL

Extending the assignment:

Propagate

$$M \parallel F, C \vee l \quad \to \quad M\, l \parallel F, C \vee l \quad \textbf{if} \begin{cases} M \text{ falsifies } C, \\ l \text{ is undefined in } M \end{cases}$$

Decide

$$M \parallel F \quad \to \quad M\, l^{\bullet} \parallel F \quad \textbf{if} \begin{cases} l \text{ or } \bar{l} \text{ occurs in } F, \\ l \text{ is undefined in } M \end{cases}$$

Notation: $l^{\bullet}$ annotates l as a decision literal.

Transition Rules for the Original DPLL

Repairing the assignment:

Fail

$$M \parallel F, C \quad \rightarrow \quad \mathit{fail} \quad \textbf{if} \begin{cases} M \text{ falsifies } C, \\ M \text{ contains no decision literals} \end{cases}$$

Transition Rules for the Original DPLL

Repairing the assignment:

Fail

$$M \parallel F, C \quad \to \quad \mathit{fail} \quad \textbf{if} \begin{cases} M \text{ falsifies } C, \\ M \text{ contains no decision literals} \end{cases}$$

Backtrack

$$M\, l^{\bullet}\, N \parallel F, C \quad \to \quad M\, \bar{l} \parallel F, C \quad \textbf{if} \begin{cases} M\, l^{\bullet}\, N \text{ falsifies } C, \\ l \text{ last decision literal} \end{cases}$$

Original DPLL System – Example

$$F := 1.\ \overline{p_1} \vee p_2,\quad 2.\ \overline{p_3} \vee p_4,\quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6,\quad 5.\ p_5 \vee p_7,\quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

Original DPLL System – Example

$$F := \begin{array}{lll} 1.\ \overline{p_1} \vee p_2, & 2.\ \overline{p_3} \vee p_4, & 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2} \\ 4.\ \overline{p_5} \vee p_6, & 5.\ p_5 \vee p_7, & 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7} \end{array}$$

M	Rule
$p_1{}^\bullet$	Decide

Original DPLL System – Example

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2,\quad 2.\ \overline{p_3} \vee p_4,\quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6,\quad 5.\ p_5 \vee p_7,\quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2$	Propagate 1.

Original DPLL System – Example

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide

Original DPLL System – Example

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
${p_1}^\bullet$	Decide
${p_1}^\bullet,\ p_2$	Propagate 1.
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet$	Decide
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet,\ p_4$	Propagate 2.

Original DPLL System – Example

$$F \;:=\; 1.\ \overline{p_1} \vee p_2,\quad 2.\ \overline{p_3} \vee p_4,\quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6,\quad 5.\ p_5 \vee p_7,\quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide

Original DPLL System – Example

$$F \;:=\; 1.\ \overline{p_1} \vee p_2,\quad 2.\ \overline{p_3} \vee p_4,\quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6,\quad 5.\ p_5 \vee p_7,\quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
${p_1}^\bullet$	Decide
${p_1}^\bullet,\ p_2$	Propagate 1.
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet$	Decide
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet,\ p_4$	Propagate 2.
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet,\ p_4,\ {p_5}^\bullet$	Decide
${p_1}^\bullet,\ p_2,\ {p_3}^\bullet,\ p_4,\ {p_5}^\bullet,\ \overline{p_6}$	Propagate 3.

Original DPLL System – Example

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2$	Propagate 1.
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4$	Propagate 2.
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4,\ p_5{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4,\ p_5{}^{\bullet},\ \overline{p_6}$	Propagate 3.
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4,\ \overline{p_5}$	Backtrack 4.

Original DPLL System – Example

$$F \;:=\; 1.\ \overline{p_1} \vee p_2,\quad 2.\ \overline{p_3} \vee p_4,\quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6,\quad 5.\ p_5 \vee p_7,\quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet,\ \overline{p_6}$	Propagate 3.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5}$	Backtrack 4.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5},\ p_7$	Propagate 5.

Original DPLL System – Example

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet,\ \overline{p_6}$	Propagate 3.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5}$	Backtrack 4.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5},\ p_7$	Propagate 5.
$p_1{}^\bullet,\ p_2,\ \overline{p_3}$	Backtrack 6.

Original DPLL System – Example

$$F \;:=\; 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet,\ \overline{p_6}$	Propagate 3.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5}$	Backtrack 4.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ \overline{p_5},\ p_7$	Propagate 5.
$p_1{}^\bullet,\ p_2,\ \overline{p_3}$	Backtrack 6.

$\ldots$

From Backtracking to Backjumping

Backtrack

$$M\, l^{\bullet}\, N \parallel F, C \quad \to \quad M\, \bar{l} \parallel F, C \quad \textbf{if} \begin{cases} M\, l^{\bullet}\, N \text{ falsifies } C, \\ l \text{ last decision literal} \end{cases}$$

From Backtracking to Backjumping

Backtrack

$$M\, l^\bullet\, N \parallel F, C \quad \to \quad M\, \bar{l} \parallel F, C \quad \textbf{if} \begin{cases} M\, l^\bullet\, N \text{ falsifies } C, \\ l \text{ last decision literal} \end{cases}$$

Backjump

$$M\, l^\bullet\, N \parallel F, C \quad \to \quad M\, k \parallel F, C \quad \textbf{if} \begin{cases} 1.\ M\, l^\bullet\, N \text{ falsifies } C, \\ 2.\ \text{for some clause } D \vee k\text{:} \\ \quad F, C \models D \vee k, \\ \quad M \text{ falsifies } D, \\ \quad k \text{ is undefined in } M, \\ \quad k \text{ or } \overline{k} \text{ occurs in} \\ \quad M\, l^\bullet\, N \parallel F, C \end{cases}$$

From Backtracking to Backjumping

Backtrack

$$M\, l^\bullet\, N \parallel F, C \quad \to \quad M\, \bar{l} \parallel F, C \quad \textbf{if} \begin{cases} M\, l^\bullet\, N \text{ falsifies } C, \\ l \text{ last decision literal} \end{cases}$$

Backjump

$$M\, l^\bullet\, N \parallel F, C \quad \to \quad M\, k \parallel F, C \quad \textbf{if} \begin{cases} 1.\ M\, l^\bullet\, N \text{ falsifies } C, \\ 2.\ \text{for some clause } D \vee k\text{:} \\ \quad F, C \models D \vee k, \\ \quad M \text{ falsifies } D, \\ \quad k \text{ is undefined in } M, \\ \quad k \text{ or } \overline{k} \text{ occurs in} \\ \quad M\, l^\bullet\, N \parallel F, C \end{cases}$$

Note: $D \vee k$ is computed by conflict analysis.

Example Revised

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet,\ \overline{p_6}$	Propagate 3.

Example Revised

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2$	Propagate 1.
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4$	Propagate 2.
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4,\ p_5{}^{\bullet}$	Decide
$p_1{}^{\bullet},\ p_2,\ p_3{}^{\bullet},\ p_4,\ p_5{}^{\bullet},\ \overline{p_6}$	Propagate 3.
$p_1{}^{\bullet},\ p_2,\ \overline{p_5}$	Backjump with $\overline{p_2} \vee \overline{p_5}$.

Example Revised

$$F \quad := \quad 1.\ \overline{p_1} \vee p_2, \quad 2.\ \overline{p_3} \vee p_4, \quad 3.\ \overline{p_6} \vee \overline{p_5} \vee \overline{p_2}$$
$$4.\ \overline{p_5} \vee p_6, \quad 5.\ p_5 \vee p_7, \quad 6.\ \overline{p_1} \vee p_5 \vee \overline{p_7}$$

M	Rule
$p_1{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2$	Propagate 1.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4$	Propagate 2.
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet$	Decide
$p_1{}^\bullet,\ p_2,\ p_3{}^\bullet,\ p_4,\ p_5{}^\bullet,\ \overline{p_6}$	Propagate 3.
$p_1{}^\bullet,\ p_2,\ \overline{p_5}$	Backjump with $\overline{p_2} \vee \overline{p_5}$.

...

Basic DPLL System

At the core, current DPLL-based SAT solvers are implementations of the transition system:

Basic DPLL

- Propagate
- Decide
- Fail
- Backjump

The Basic DPLL System – Correctness

Some terminology

Irreducible state: state to which no transition rule applies.

Execution: sequence of transitions allowed by the rules and starting with states of the form $\emptyset \parallel F$.

Exhausted execution: execution ending in an irreducible state.

The Basic DPLL System – Correctness

Some terminology

Irreducible state: state to which no transition rule applies.

Execution: sequence of transitions allowed by the rules and starting with states of the form $\emptyset \parallel F$.

Exhausted execution: execution ending in an irreducible state.

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

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

The Basic DPLL System – Correctness

Some terminology

Irreducible state: state to which no transition rule applies.

Execution: sequence of transitions allowed by the rules and starting with states of the form $\emptyset \parallel F$.

Exhausted execution: execution ending in an irreducible state.

Proposition (Soundness) For every exhausted execution starting with $\emptyset \parallel F$ and ending in $M \parallel F$, M satisfies F.

Proposition (Completeness) If F is unsatisfiable, every exhausted execution starting with $\emptyset \parallel F$ ends with $fail$.

Enhancements to Basic DPLL

Enhancements to Basic DPLL

Learn

$$M \parallel F \;\; \longrightarrow \;\; M \parallel F, C \quad \textbf{if} \begin{cases} \text{all atoms of } C \text{ occur in } F, \\ F \models C \end{cases}$$

Enhancements to Basic DPLL

Learn

$$M \parallel F \;\; \rightarrow \;\; M \parallel F, C \quad \textbf{if} \begin{cases} \text{all atoms of } C \text{ occur in } F, \\ F \models C \end{cases}$$

Forget

$$M \parallel F, C \;\; \rightarrow \;\; M \parallel F \quad \textbf{if } F \models C$$

Enhancements to Basic DPLL

Learn

$$M \parallel F \quad \rightarrow \quad M \parallel F, C \quad \textbf{if} \begin{cases} \text{all atoms of } C \text{ occur in } F, \\ F \models C \end{cases}$$

Forget

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

Usually C is a clause identified during conflict analysis.

Enhancements to Basic DPLL

Learn

$$M \parallel F \;\; \rightarrow \;\; M \parallel F, C \;\; \textbf{if} \begin{cases} \text{all atoms of } C \text{ occur in } F, \\ F \models C \end{cases}$$

Forget

$$M \parallel F, C \;\; \rightarrow \;\; M \parallel F \;\; \textbf{if } F \models C$$

Restart

$M \parallel F \;\; \rightarrow \;\; \emptyset \parallel F \;\; \textbf{if}$ *. . . you want to*

Enhancements to Basic DPLL

Learn

$$M \parallel F \;\; \to \;\; M \parallel F, C \;\; \textbf{if} \begin{cases} \text{all atoms of } C \text{ occur in } F, \\ F \models C \end{cases}$$

Forget

$$M \parallel F, C \;\; \to \;\; M \parallel F \;\; \textbf{if } F \models C$$

Restart

$$M \parallel F \;\; \to \;\; \emptyset \parallel F \;\; \textbf{if } \ldots \textit{you want to}$$

The DPLL system $=$

$\{$Propagate, Decide, Fail, Backjump, Learn, Forget, Restart$\}$

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.
- In practice, Learn is usually (but not only) applied right after Backjump.

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.
- In practice, Learn is usually (but not only) applied right after Backjump.
- A common strategy is to apply the rules with these priorities:

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.
- In practice, Learn is usually (but not only) applied right after Backjump.
- A common strategy is to apply the rules with these priorities:
  1. If $n > 0$ conflicts have been found so far, increase n and apply Restart.

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.
- In practice, Learn is usually (but not only) applied right after Backjump.
- A common strategy is to apply the rules with these priorities:
  1. If $n > 0$ conflicts have been found so far, increase n and apply Restart.
  2. If a current clause is falsified by the current assignment,
     apply Fail or Backjump + Learn.

The DPLL System – Strategies

- Applying one Basic DPLL rule between each two Learn and applying Restart less and less often ensures termination.
- In practice, Learn is usually (but not only) applied right after Backjump.
- A common strategy is to apply the rules with these priorities:
  1. If $n > 0$ conflicts have been found so far, increase n and apply Restart.
  2. If a current clause is falsified by the current assignment, apply Fail or Backjump + Learn.
  3. Apply Propagate

The DPLL System – Correctness

Proposition (Termination) Every execution in which
(a) Learn/Forget are applied only finitely many times and
(b) Restart is applied with increased periodicity
is finite.

The DPLL System – Correctness

Proposition (Termination) Every execution in which
(a) Learn/Forget are applied only finitely many times and
(b) Restart is applied with increased periodicity
is finite.

Proposition (Soundness) For every execution $\emptyset \parallel F \Longrightarrow \cdots \Longrightarrow M \parallel F$ with $M \parallel F$ irreducible wrt. Basic DPLL, M models F.

The DPLL System – Correctness

Proposition (Termination) Every execution in which
(a) Learn/Forget are applied only finitely many times and
(b) Restart is applied with increased periodicity
is finite.

Proposition (Soundness) For every execution $\emptyset \parallel F \Longrightarrow \cdots \Longrightarrow M \parallel F$ with $M \parallel F$ irreducible wrt. Basic DPLL, M models F.

Proposition (Completeness) If F is unsatisfiable, for every execution $\emptyset \parallel F \Longrightarrow \cdots \Longrightarrow S$ with S irreducible wrt. Basic DPLL, $S = \mathit{fail}$.