

Lookahead Techniques

Marijn J.H. Heule

**Carnegie
Mellon
University**

<http://www.cs.cmu.edu/~mheule/15816-f22/>

Automated Reasoning and Satisfiability

September 28, 2022

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Davis Putnam Logemann Loveland [DP60,DLL62]

Recursive procedure that in each recursive call:

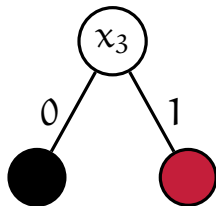
- Simplifies the formula (using unit propagation)
- Splits the formula into two subformulas
 - Variable selection heuristics (which variable to split on)
 - Direction heuristics (which subformula to explore first)

DPLL: Example

$$F_{\text{DPLL}} := (\chi_1 \vee \chi_2 \vee \bar{\chi}_3) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge \\ (\bar{\chi}_1 \vee \bar{\chi}_2 \vee \chi_3) \wedge (\chi_1 \vee \chi_3) \wedge (\bar{\chi}_1 \vee \bar{\chi}_3)$$

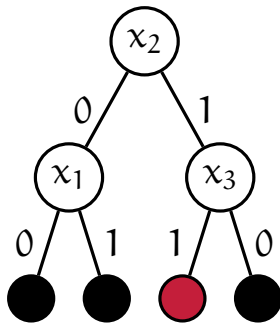
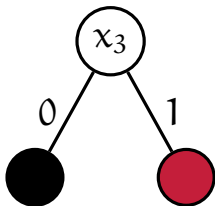
DPLL: Example

$$F_{\text{DPLL}} := (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3)$$



DPLL: Example

$$F_{\text{DPLL}} := (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3)$$



Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead:

- Assign a variable to a truth value

Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead:

- Assign a variable to a truth value
- Simplify the formula

Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead:

- Assign a variable to a truth value
- Simplify the formula
- Measure the reduction

Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead:

- Assign a variable to a truth value
- Simplify the formula
- Measure the reduction
- Learn if possible

Look-ahead: Definition

DPLL with selection of (effective) decision variables by **look-aheads** on variables

Look-ahead:

- Assign a variable to a truth value
- Simplify the formula
- Measure the reduction
- Learn if possible
- Backtrack

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Look-ahead: Example

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Look-ahead: Example

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0\}$$

Look-ahead: Example

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0\}$$

Look-ahead: Example

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0\}$$

Look-ahead: Example

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

Look-ahead: Properties

- Very expensive
- Effective compared to cheap heuristics
- Detection of failed literals (and more)
- Strong on random k -SAT formulae
- Examples: march, OKsolver, kcnfs

DEMO

Look-ahead: Reduction heuristics

- Number of satisfied clauses

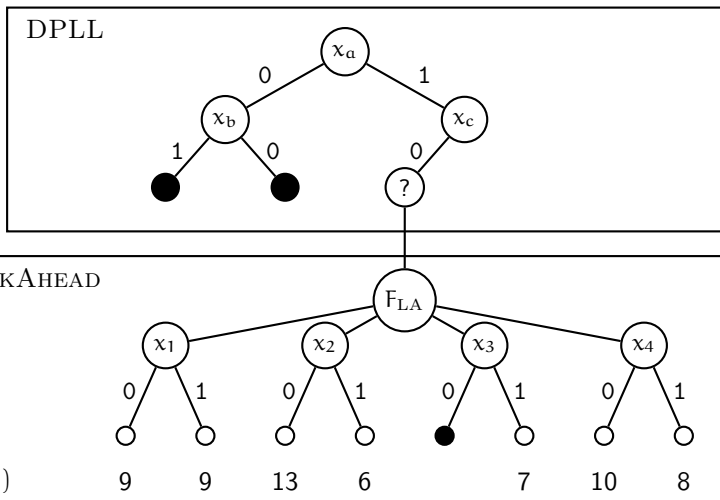
Look-ahead: Reduction heuristics

- Number of satisfied clauses
- Number of implied variables

Look-ahead: Reduction heuristics

- Number of satisfied clauses
- Number of implied variables
- New (reduced, not satisfied) clauses
 - Smaller clauses more important
 - Weights based on occurrences

Look-ahead: Architecture



Look-ahead: Pseudo-code of DPLL with lookahead

- 1: $F := \text{Simplify}(F)$
- 2: **if** F is empty **then return** satisfiable
- 3: **if** $\perp \in F$ **then return** unsatisfiable
- 4: $\langle F; l_{\text{decision}} \rangle := \text{LookAhead}(F)$
- 5: **if** ($\text{DPLL}(F(l_{\text{decision}} \leftarrow 1)) = \text{satisfiable}$) **then**
- 6: **return** satisfiable
- 7: **return** $\text{DPLL}(F(l_{\text{decision}} \leftarrow 0))$

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Local Learning

Look-ahead solvers do not perform global learning, in contrast to conflict-driven clause learning (CDCL) solvers

Instead, look-ahead solvers learn locally:

- Learn small (typically unit or binary) clauses that are valid for the current node and lower in the DPLL tree
- Locally learnt clauses have to be removed during backtracking

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_4 = 0, x_6 = 1\}$$

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_4 = 0, x_6 = 1, x_1 = 1\}$$

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_4 = 0, x_6 = 1, x_1 = 1, x_2 = 1\}$$

Failed Literals and Double Look-aheads

A literal l is called a **failed literal** if the look-ahead on $l = 1$ results in a conflict:

- failed literal l is forced to false followed by unit propagation
- if both x and \bar{x} are failed literals, then backtrack

Failed literals can be generalized by **double lookahead**: assign two literals and learn a binary clause in case of a conflict.

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

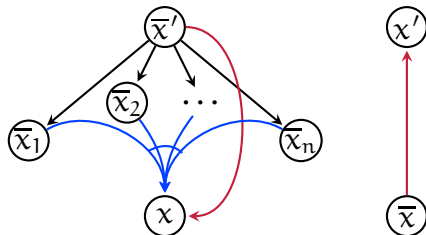
$$\alpha = \{x_4 = 0, x_6 = 1, x_1 = 1, x_2 = 1, x_3 = 1\}$$

Hyper Binary Resolution [Bacchus 2002]

Definition (Hyper Binary Resolution Rule)

$$\frac{(x \vee x_1 \vee x_2 \vee \dots \vee x_n) \quad (\bar{x}_1 \vee x') \quad (\bar{x}_2 \vee x') \quad \dots \quad (\bar{x}_n \vee x')}{(x \vee x')}$$

binary edge \longrightarrow
hyper edge \longrightarrow
hyper binary edge \longrightarrow



Hyper Binary Resolution Rule:

- combines multiple resolution steps into one
- uses one n-ary clauses and multiple binary clauses
- special case *hyper unary resolution* where $x = x'$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0\}$$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0\}$$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0\}$$

Look-ahead: Hyper Binary Resolvents

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

hyper binary resolvents:

$$(x_2 \vee \bar{x}_6) \text{ and } (x_2 \vee x_3)$$

Look-ahead: Hyper Binary Resolvents

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

hyper binary resolvents:

$(x_2 \vee \bar{x}_6)$ and $(x_2 \vee x_3)$

Which one is more useful?

Look-ahead: Necessary assignments

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Look-ahead: Necessary assignments

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_1 = 1\}$$

Look-ahead: Necessary assignments

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_1 = 1, x_2 = 1\}$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1\}$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 0\}$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 0, x_6 = 0\}$$

Look-ahead: Necessary assignments

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 0, x_6 = 0, x_3 = 1\}$$

Stålmarck's Method

In short, Stålmarck's Method is a procedure that generalizes the concept of necessary assignments.

For each variable x , $(\text{Simplify}(F|x) \cap \text{Simplify}(F|\bar{x})) \setminus F$ is added to F .

The above is repeated until **fixpoint**, i.e., until

$$\forall x : (\text{Simplify}(F|x) \cap \text{Simplify}(F|\bar{x})) \setminus F = \emptyset$$

Afterwards the procedure is repeated using **all pairs** for variables x and y : Add $(\text{Simplify}(F|x y) \cap \text{Simplify}(F|x \bar{y}) \cap \text{Simplify}(F|\bar{x} y) \cap \text{Simplify}(F|\bar{x} \bar{y})) \setminus F$ to F .

The second round is **very expensive** and can typically not be finished in reasonable time.

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Look-ahead: Autarky definition

An **autarky** is a partial assignment that satisfies all clauses that are “touched” by the assignment

Look-ahead: Autarky definition

An **autarky** is a partial assignment that satisfies all clauses that are “touched” by the assignment

- a pure literal is an autarky

Look-ahead: Autarky definition

An **autarky** is a partial assignment that satisfies all clauses that are “touched” by the assignment

- a pure literal is an autarky
- each satisfying assignment is an autarky

Look-ahead: Autarky definition

An **autarky** is a partial assignment that satisfies all clauses that are “touched” by the assignment

- a pure literal is an autarky
- each satisfying assignment is an autarky
- the remaining formula is **satisfiability equivalent** to the original formula

Look-ahead: Autarky definition

An **autarky** is a partial assignment that satisfies all clauses that are “touched” by the assignment

- a pure literal is an autarky
- each satisfying assignment is an autarky
- the remaining formula is **satisfiability equivalent** to the original formula

An **1-autarky** is a partial assignment that satisfies all touched clauses except one

Look-ahead: Autarky detection

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1\}$$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1\}$$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1\}$$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

F_{learning} satisfiability equivalent to $(x_5 \vee \bar{x}_6)$

Look-ahead: Autarky detection

$$F_{\text{learning}} := (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6)$$

$$\alpha = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1\}$$

F_{learning} satisfiability equivalent to $(x_5 \vee \bar{x}_6)$

Could reduce computational cost on UNSAT

Look-ahead: Autarky or Conflict on 2-SAT Formulae

Lookahead techniques can solve 2-SAT formulae in polynomial time. Each lookahead on l results:

1. in an autarky: forcing l to be true
2. in a conflict: forcing l to be false

Look-ahead: Autarky or Conflict on 2-SAT Formulae

Lookahead techniques can solve 2-SAT formulae in polynomial time. Each lookahead on l results:

1. in an autarky: forcing l to be true
2. in a conflict: forcing l to be false

SAT Game

by Olivier Roussel

<http://www.cs.utexas.edu/~marijn/game/2SAT>

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0\}$$

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0\}$$

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0\}$$

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

Look-ahead: 1-Autarky learning

$$\begin{aligned} F_{\text{learning}} := & (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \\ & (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_6) \wedge (\bar{x}_1 \vee x_4 \vee \bar{x}_5) \wedge \\ & (x_1 \vee \bar{x}_6) \wedge (x_4 \vee x_5 \vee x_6) \wedge (x_5 \vee \bar{x}_6) \end{aligned}$$

$$\alpha = \{x_2 = 0, x_1 = 0, x_6 = 0, x_3 = 1\}$$

(local) 1-autarky resolvents:

$$(\bar{x}_2 \vee \bar{x}_4) \text{ and } (\bar{x}_2 \vee \bar{x}_5)$$