Lookahead Techniques

Ruben Martins

Carnegie Mellon University

http://www.cs.cmu.edu/~rubenm/15816-f25/

Automated Reasoning and Satisfiability

September 22, 2024

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

SAT Solving: DPLL

Davis Putnam Logemann Loveland [DP60,DLL62]

Recursive procedure that in each recursive call:

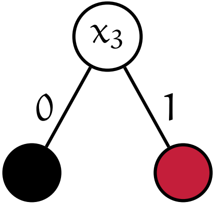
- Simplifies the formula (using unit propagation)
- Splits the formula into two subformulas
  - Variable selection heuristics (which variable to split on)
  - Direction heuristics (which subformula to explore first)

DPLL: Example

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

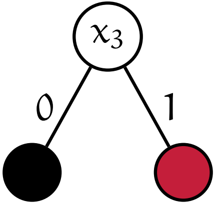
DPLL: Example

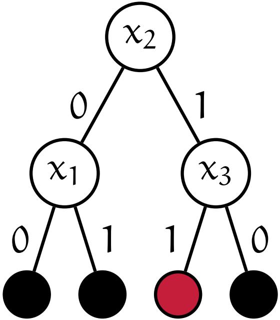
$$\Gamma_{\mathrm{DPLL}} := (x_1 \lor x_2 \lor \overline{x}_3) \land (\overline{x}_1 \lor x_2 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (x_1 \lor x_3) \land (\overline{x}_1 \lor \overline{x}_3)$$



DPLL: Example

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





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

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

Look-ahead: Example

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Example

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

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

Look-ahead: Example

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Example

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

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

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

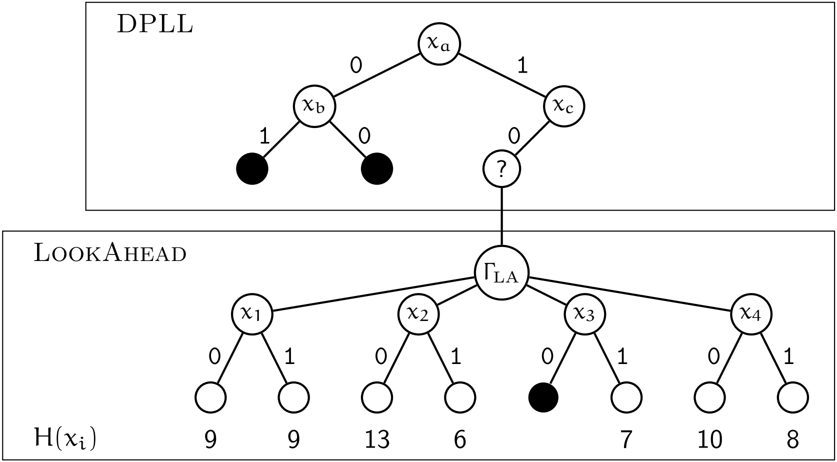
Look-ahead: Reduction heuristics

- 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

$\Gamma :=$ Simplify (Γ)

if Γ is empty **then return** `satisfiable`

if $\bot \in \Gamma$ **then return** `unsatisfiable`

$\langle \Gamma;\ l_{\text{decision}} \rangle :=$ LookAhead (Γ)

if $(\text{DPLL}(\Gamma | l_{\text{decision}}) = \texttt{satisfiable})$ **then**

  return `satisfiable`

return DPLL $(\Gamma | \overline{l_{\text{decision}}})$

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Local Learning

Look-ahead solvers do not perform global learning, in contrast to 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 $\overline{x}$ 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 $\overline{x}$ 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.

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

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 $\overline{x}$ 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.

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

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 $\overline{x}$ 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.

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

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

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.

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

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

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 $\overline{x}$ 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.

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

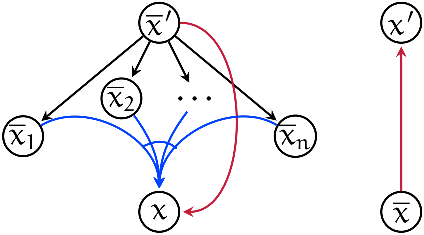
$$\alpha = \{x_4 = 0, x_6 = 1, x_1 = 1, x_2 = 1, x_3 = 1\}$$

Hyper Binary Resolution [Bacchus 2002]

Definition (Hyper Binary Resolution Rule)

$$\frac{(x \vee x_1 \vee x_2 \vee \cdots \vee x_n) \quad (\overline{x}_1 \vee x') \quad (\overline{x}_2 \vee x') \quad \ldots \quad (\overline{x}_n \vee x')}{(x \vee x')}$$

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

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

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

rubenm@cmu.edu

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

hyper binary resolvents:

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

Look-ahead: Hyper Binary Resolvents

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

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

hyper binary resolvents:

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

Which one is more useful?

Look-ahead: Necessary assignments

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

Look-ahead: Necessary assignments

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

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

Look-ahead: Necessary assignments

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Necessary assignments

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land$$
$$(\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land$$
$$(x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Necessary assignments

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Necessary assignments

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

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

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

Look-ahead: Necessary assignments

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: Necessary assignments

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

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

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

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

Look-ahead: Necessary assignments

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

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

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

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

Stålmarck's Method

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

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

The above is repeated until fixpoint, i.e., until $\forall x : (\text{Simplify}(\Gamma|x) \cap \text{Simplify}(\Gamma|\overline{x})) \setminus F = \emptyset$

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

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Γ_{learning} satisfiability equivalent to $(x_5 \vee \overline{x}_6)$

Look-ahead: Autarky detection

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

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

Γ_{learning} satisfiability equivalent to $(x_5 \vee \overline{x}_6)$

Could reduce computational cost on UNSAT

Look-ahead: 1-Autarky learning

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

Look-ahead: 1-Autarky learning

$$\Gamma_{\text{learning}} := (\overline{x}_1 \lor \overline{x}_3 \lor x_4) \land (\overline{x}_1 \lor \overline{x}_2 \lor x_3) \land (\overline{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\overline{x}_1 \lor x_4 \lor \overline{x}_5) \land (x_1 \lor \overline{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \overline{x}_6)$$

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

Look-ahead: 1-Autarky learning

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

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

Look-ahead: 1-Autarky learning

$$\Gamma_{\text{learning}} := (\bar{x}_1 \lor \bar{x}_3 \lor x_4) \land (\bar{x}_1 \lor \bar{x}_2 \lor x_3) \land (\bar{x}_1 \lor x_2) \land (x_1 \lor x_3 \lor x_6) \land (\bar{x}_1 \lor x_4 \lor \bar{x}_5) \land (x_1 \lor \bar{x}_6) \land (x_4 \lor x_5 \lor x_6) \land (x_5 \lor \bar{x}_6)$$

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

Look-ahead: 1-Autarky learning

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

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

Look-ahead: 1-Autarky learning

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

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

(local) 1-autarky resolvents:
$(\bar{x}_2 \vee \bar{x}_4)$ and $(\bar{x}_2 \vee \bar{x}_5)$

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`