Lookahead Techniques

Marijn J.H. Heule

Carnegie Mellon University

http://www.cs.cmu.edu/~mheule/15816-f19/

Automated Reasoning and Satisfiability, September 26, 2019

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Tree-based Look-ahead

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Tree-based Look-ahead

SAT Solving: DPLL

Davis Putnam Logemann Loveland [DP60,DLL62]

Recursive procedure that in each recursive call:

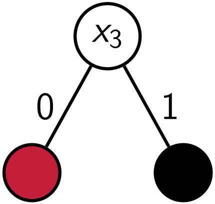
- Simplifies the formula (using unit propagation)
- Splits the formula into two subformulas
    - Variable selection heuristics (which variable to split on)
    - Direction heuristics (which subformula to explore first)

DPLL: Example

$$\mathcal{F}_{\mathrm{DPLL}} := (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge$$
$$(\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)$$

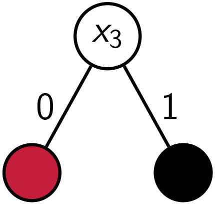
DPLL: Example

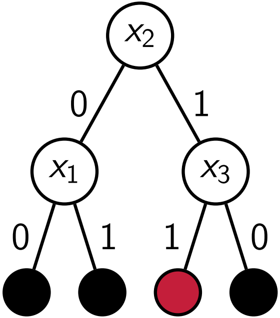
$$\mathcal{F}_{\mathrm{DPLL}} := (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)$$



DPLL: Example

$$\mathcal{F}_{\mathrm{DPLL}} := (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)$$





Slightly Harder Example

Construct a DPLL tree for:

$$(a \vee b \vee \overline{c}) \wedge (\overline{a} \vee \overline{b} \vee c) \wedge \\ (b \vee c \vee \overline{d}) \wedge (\overline{b} \vee \overline{c} \vee d) \wedge \\ (a \vee c \vee d) \wedge (\overline{a} \vee \overline{c} \vee \overline{d}) \wedge \\ (\overline{a} \vee b \vee d)$$

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

Look-ahead: Example

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

Look-ahead: Example

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

$$\varphi = \{x_2{=}0\}$$

Look-ahead: Example

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

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

Look-ahead: Example

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

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

Look-ahead: Example

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

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

Look-ahead: Properties

- Very expensive

Look-ahead: Properties

- Very expensive
- Effective compared to cheap heuristics

Look-ahead: Properties

- Very expensive
- Effective compared to cheap heuristics
- Detection of failed literals (and more)

Look-ahead: Properties

- Very expensive
- Effective compared to cheap heuristics
- Detection of failed literals (and more)
- Strong on random k-SAT formulae

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

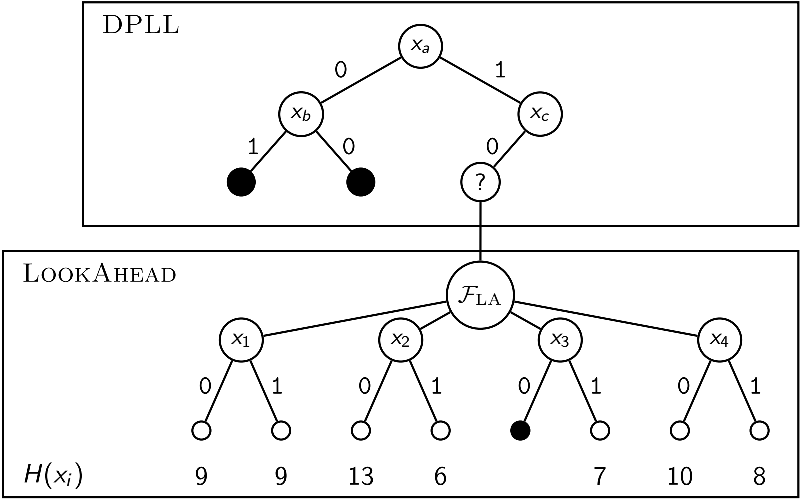
Look-ahead: Reduction heuristics

- 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 occurring

Look-ahead: Architecture



Look-ahead: Pseudo-code

$\mathcal{F}$:= Simplify ($\mathcal{F}$)

if $\mathcal{F}$ is empty **then return** `satisfiable`

if $\emptyset \in \mathcal{F}$ **then return** `unsatisfiable`

$\langle \mathcal{F};\ l_{\text{decision}} \rangle$:= LookAhead ($\mathcal{F}$)

if (DPLL($\mathcal{F}(l_{\text{decision}} \leftarrow 1)$) = `satisfiable`) **then**

    return `satisfiable`

return DPLL ($\mathcal{F}(l_{\text{decision}} \leftarrow 0)$)

DPLL Procedure

Look-ahead Architecture

Look-ahead Learning

Autarky Reasoning

Tree-based Look-ahead

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.

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

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.

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

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

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.

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

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

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 $\overline{x}$ 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.

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

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

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.

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

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

Hyper Binary Resolution [Bacchus 2002]

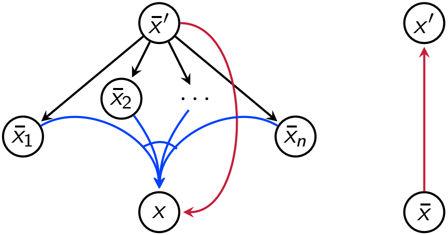
Definition (Hyper Binary Resolution Rule)

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

binary edge →

hyper edge →

hyper binary edge →



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

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

Look-ahead: Hyper Binary Resolvents

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

$$\varphi = \{x_2{=}0\}$$

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

Look-ahead: Hyper Binary Resolvents

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

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

hyper binary resolvents:

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

Look-ahead: Hyper Binary Resolvents

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

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

hyper binary resolvents:

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

Which one is more useful?

Look-ahead: Necessary assignments

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

Look-ahead: Necessary assignments

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

$$\varphi = \{x_1{=}1\}$$

Look-ahead: Necessary assignments

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

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

Look-ahead: Necessary assignments

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

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

Look-ahead: Necessary assignments

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

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

Look-ahead: Necessary assignments

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

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

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

Look-ahead: Necessary assignments

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

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

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

$$\varphi = \{x_1{=}0\}$$

Look-ahead: Necessary assignments

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

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

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

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

Look-ahead: Necessary assignments

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

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

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

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

Look-ahead: Necessary assignments

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

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

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

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

Stålmarck's Method

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

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

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

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

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

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

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

Look-ahead: Autarky detection

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

$\varphi = \{x_1{=}1\}$

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

Look-ahead: Autarky detection

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

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

$\mathcal{F}_{\text{learning}}$ satisfiability equivalent to $(x_5 \vee \overline{x}_6)$

Look-ahead: Autarky detection

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

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

$\mathcal{F}_{\text{learning}}$ satisfiability equivalent to $(x_5 \vee \overline{x}_6)$

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

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

Look-ahead: 1-Autarky learning

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

$$\varphi = \{x_2{=}0\}$$

Look-ahead: 1-Autarky learning

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

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

Look-ahead: 1-Autarky learning

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

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

Look-ahead: 1-Autarky learning

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

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

Look-ahead: 1-Autarky learning

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

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

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

DPLL Procedure

Look-ahead Architecture

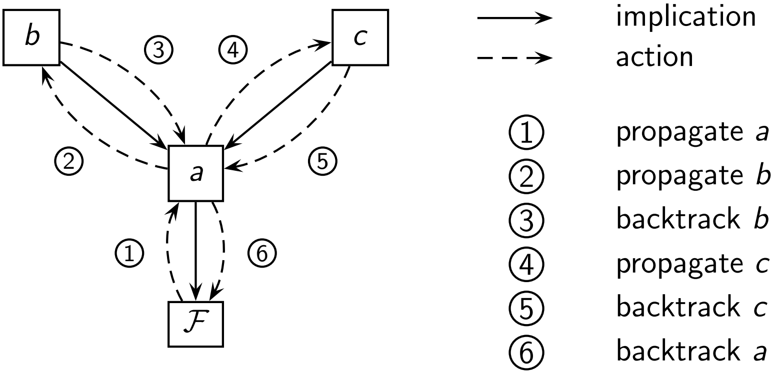
Look-ahead Learning

Autarky Reasoning

Tree-based Look-ahead

Tree-based Look-ahead

Given a formula F which includes the clauses $(a \vee \overline{b})$ and $(a \vee \overline{c})$, tree-based look-ahead can reduce the look-ahead costs.



Lookahead Techniques

Marijn J.H. Heule

Carnegie Mellon University

http://www.cs.cmu.edu/~mheule/15816-f19/

Automated Reasoning and Satisfiability, September 26, 2019