

Formal Software Verification Project Presentation

SAT Solver

Nathan Schmidt

January 6, 2025

Aarhus University

Syntax

$$p, q ::= x \mid \text{true} \mid \text{false} \mid p \wedge q \mid p \vee q \mid p \rightarrow q \mid \neg p$$

Semantics

- Valuation $v : \text{id} \rightarrow \text{bool}$
- Interpreter $\text{interp} : \text{valuation} \times \text{form} \rightarrow \text{bool}$

Simplifications

$$\text{true} \wedge p \equiv p$$

$$p \wedge \text{true} \equiv p$$

$$\text{false} \wedge p \equiv \text{false}$$

$$p \wedge \text{false} \equiv \text{false}$$

$$\text{true} \vee p \equiv \text{true}$$

$$p \vee \text{true} \equiv \text{true}$$

$$\text{false} \vee p \equiv p$$

$$p \vee \text{false} \equiv p$$

$$\text{true} \rightarrow p \equiv p$$

$$p \rightarrow \text{true} \equiv \text{true}$$

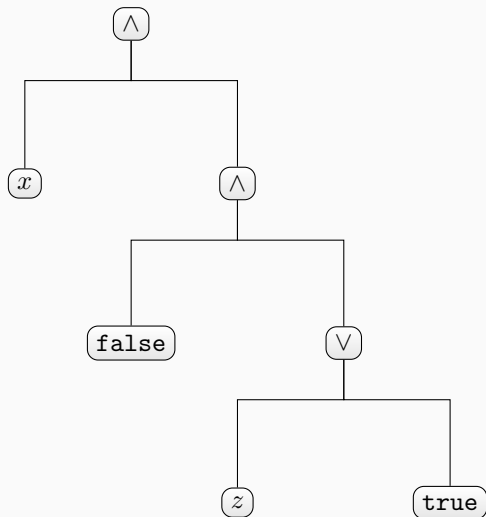
$$\text{false} \rightarrow p \equiv \text{true}$$

$$p \rightarrow \text{false} \equiv \neg p$$

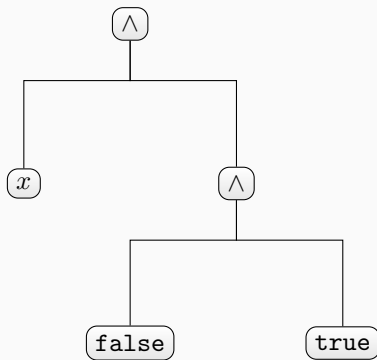
$$\neg \text{true} \equiv \text{false}$$

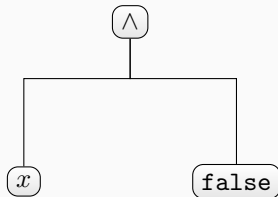
$$\neg \text{false} \equiv \text{true}$$

- Depth-First-Search (DFS):
 - ✗ Pre-order: no obviously decreasing arguments
 - ✓ Post-order



- Depth-First-Search (DFS):
 - ✗ Pre-order: no obviously decreasing arguments
 - ✓ Post-order





- Depth-First-Search (DFS):
 - ✗ Pre-order: no obviously decreasing arguments
 - ✓ Post-order

false

- Depth-First-Search (DFS):
 - ✗ Pre-order: no obviously decreasing arguments
 - ✓ Post-order

Minimality

- **Theorem.** *optim p always in minimal form*
- p in **minimal form**:
 - if $p = \text{true/false}$, or
 - $\text{true/false} \notin p$

Minimality

- **Theorem.** *optim p always in minimal form*
- *p* in **minimal form**:
 - if $p = \text{true/false}$, or
 - $\text{true/false} \notin p$

Correctness

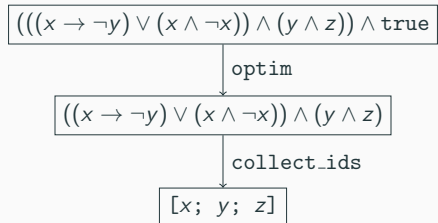
Theorem. $\forall v, p : \text{interp } v \ p = \text{interp } v \ (\text{optim } p)$

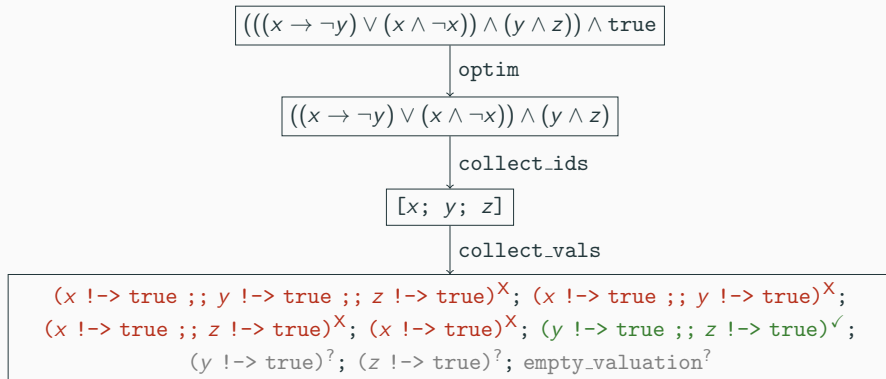
$$(((x \rightarrow \neg y) \vee (x \wedge \neg x)) \wedge (y \wedge z)) \wedge \text{true}$$

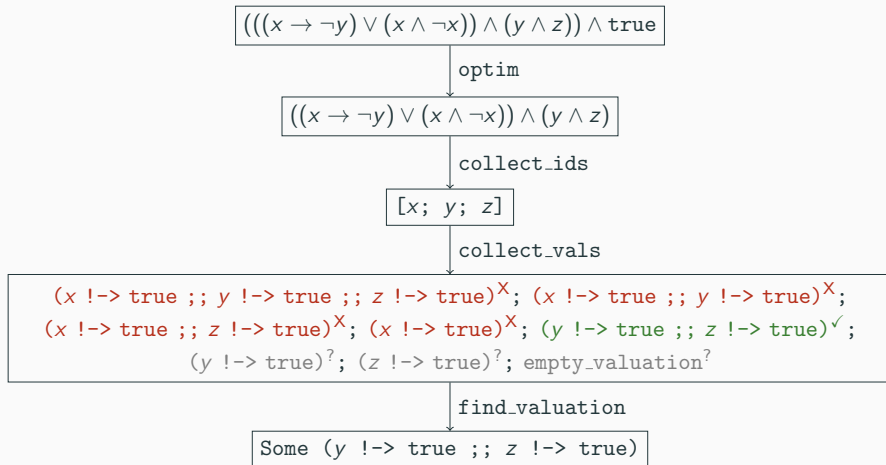
$((x \rightarrow \neg y) \vee (x \wedge \neg x)) \wedge (y \wedge z) \wedge \text{true}$

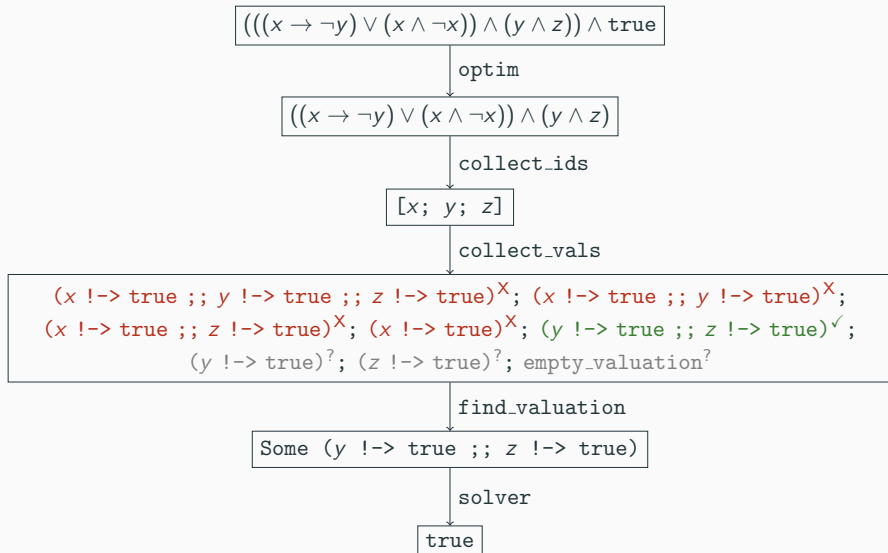
↓
optim

$((x \rightarrow \neg y) \vee (x \wedge \neg x)) \wedge (y \wedge z)$









Solver Soundness and Completeness

Soundness

- **Lemma.** $\forall p : \text{solver } p = \text{true} \implies \text{satisfiable } p$
- Proof by induction
- Do **not** care about **exact** returned valuation

Solver Soundness and Completeness

Soundness

- **Lemma.** $\forall p : \text{solver } p = \text{true} \implies \text{satisfiable } p$
- Proof by induction
- Do **not** care about **exact** returned valuation

Completeness

- **Lemma.** $\forall p : \text{satisfiable } p \implies \text{solver } p = \text{true}$
- $\text{interp } v \text{ } p = \text{true} \not\Rightarrow v \in \text{collect_vals}(\text{collect_ids } p)$

Solver Soundness and Completeness

Soundness

- **Lemma.** $\forall p : \text{solver } p = \text{true} \implies \text{satisfiable } p$
- Proof by induction
- Do **not** care about **exact** returned valuation

Completeness

- **Lemma.** $\forall p : \text{satisfiable } p \implies \text{solver } p = \text{true}$
- $\text{interp } v \text{ } p = \text{true} \not\Rightarrow v \in \text{collect_vals}(\text{collect_ids } p)$
- Proof attempt:
 - $\forall v, v' : \forall x \in p, v x = v' x \implies \text{interp } v \text{ } p = \text{interp } v' \text{ } p$
 - $\forall v : \exists v' : \forall x \in p, v x = v' x \text{ and } v' \in \text{collect_vals}(\text{collect_ids } p)$

Solver Decision Procedure

Theorem

$$\forall p : \text{solver } p = \text{true} \iff \text{satisfiable } p$$

Theorem

$$\forall p: \text{solver } p = \text{true} \iff \text{satisfiable } p$$

i.e., solver decision procedure for SAT