AARHUS UNIVERSITY

Formal Software Verification Project Presentation

SAT Solver

Nathan Schmidt

January 7, 2025

Aarhus University

Syntax and Semantics

Syntax

$$p, q ::= x \mid \texttt{true} \mid \texttt{false} \mid p \land q \mid p \lor q \mid p \to q \mid \neg p$$

- Abstract: inductive type `form`
- Concrete: `Notation`

Syntax and Semantics

Syntax

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

- Abstract: inductive type `form`
- Concrete: `Notation`

Semantics

- Valuation v : `id` $\to$ `bool`
- Interpreter `interp` : `valuation` $\times$ `form` $\to$ `bool`

Simplifications

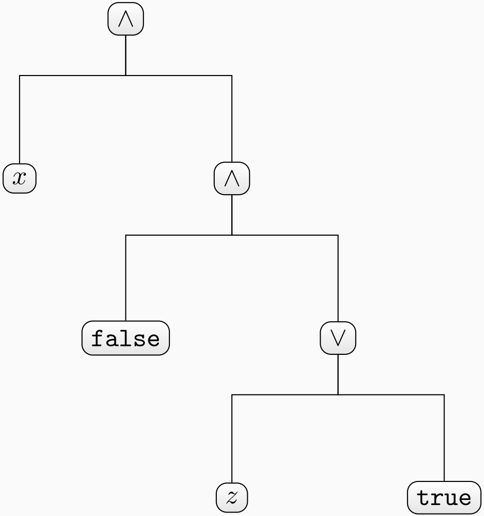
$$\begin{array}{cccc}
\mathtt{true} \wedge p \equiv p & p \wedge \mathtt{true} \equiv p & \mathtt{false} \wedge p \equiv \mathtt{false} & p \wedge \mathtt{false} \equiv \mathtt{false} \\
\mathtt{true} \vee p \equiv \mathtt{true} & p \vee \mathtt{true} \equiv \mathtt{true} & \mathtt{false} \vee p \equiv p & p \vee \mathtt{false} \equiv p \\
\mathtt{true} \rightarrow p \equiv p & p \rightarrow \mathtt{true} \equiv \mathtt{true} & \mathtt{false} \rightarrow p \equiv \mathtt{true} & p \rightarrow \mathtt{false} \equiv \neg p \\
 & \neg\mathtt{true} \equiv \mathtt{false} & \neg\mathtt{false} \equiv \mathtt{true} &
\end{array}$$

Optimizer

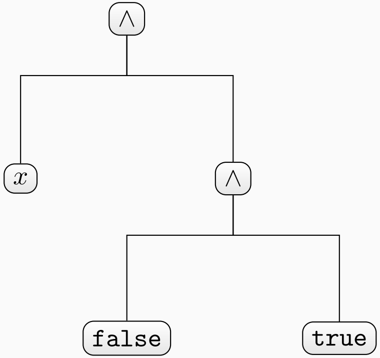
- Syntax-driven
- Depth-First-Search (DFS):
  - ✗ Pre-order: no obviously decreasing arguments
  - ✓ Post-order

Optimizer

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

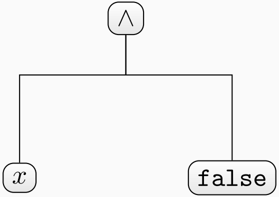


Optimizer



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

Optimizer



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

Optimizer

`false`

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

Optimizer Correctness and Minimality

Correctness

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

Optimizer Correctness and Minimality

Correctness

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

Minimality

- p in minimal form:
  - if $p = \texttt{true}/\texttt{false}$, or
  - $\texttt{true}/\texttt{false} \notin p$

Optimizer Correctness and Minimality

Correctness

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

Minimality

- p in minimal form:
  - if $p = \texttt{true}/\texttt{false}$, or
  - $\texttt{true}/\texttt{false} \notin p \rightsquigarrow$ inductive predicate

Optimizer Correctness and Minimality

Correctness

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

Minimality

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

Solver

$$\boxed{(((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)) \land \texttt{true}}$$

Solver

$$\boxed{(((x \to \neg y) \vee (x \wedge \neg x)) \wedge (y \wedge z)) \wedge \mathtt{true}}$$

$$\big\downarrow \texttt{optim}$$

$$\boxed{((x \to \neg y) \vee (x \wedge \neg x)) \wedge (y \wedge z)}$$

Solver

$$\begin{tikzcd}
\boxed{(((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)) \land \texttt{true}} \arrow[d, "\texttt{optim}"] \\
\boxed{((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)} \arrow[d, "\texttt{collect_ids}"] \\
\boxed{[x;\ y;\ z]}
\end{tikzcd}$$

Solver

$$(((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)) \land \texttt{true}$$

$$\Big\downarrow\ \texttt{optim}$$

$$((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)$$

$$\Big\downarrow\ \texttt{collect\_ids}$$

$$[x;\ y;\ z]$$

$$\Big\downarrow\ \texttt{collect\_vals}$$

$(x\ \texttt{!-> true ;;}\ y\ \texttt{!-> true ;;}\ z\ \texttt{!-> true})^{\text{X}}$; $(x\ \texttt{!-> true ;;}\ y\ \texttt{!-> true})^{\text{X}}$;
$(x\ \texttt{!-> true ;;}\ z\ \texttt{!-> true})^{\text{X}}$; $(x\ \texttt{!-> true})^{\text{X}}$; $(y\ \texttt{!-> true ;;}\ z\ \texttt{!-> true})^{\checkmark}$;
$(y\ \texttt{!-> true})^{?}$; $(z\ \texttt{!-> true})^{?}$; $\texttt{empty\_valuation}^{?}$

Solver

$$(((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)) \land \texttt{true}$$

$$\Big\downarrow \; \texttt{optim}$$

$$((x \to \neg y) \lor (x \land \neg x)) \land (y \land z)$$

$$\Big\downarrow \; \texttt{collect\_ids}$$

$$[x;\; y;\; z]$$

$$\Big\downarrow \; \texttt{collect\_vals}$$

$(x$ `!-> true ;;` y `!-> true ;;` z `!-> true)`$^{\text{X}}$; $(x$ `!-> true ;;` y `!-> true)`$^{\text{X}}$;
$(x$ `!-> true ;;` z `!-> true)`$^{\text{X}}$; $(x$ `!-> true)`$^{\text{X}}$; $(y$ `!-> true ;;` z `!-> true)`$^{\checkmark}$;
$(y$ `!-> true)`$^{?}$; $(z$ `!-> true)`$^{?}$; `empty_valuation`$^{?}$

$$\Big\downarrow \; \texttt{find\_valuation}$$

`Some (`y `!-> true ;;` z `!-> true)`

Solver

$$(((x \rightarrow \neg y) \lor (x \land \neg x)) \land (y \land z)) \land \texttt{true}$$

$$\Big\downarrow \texttt{optim}$$

$$((x \rightarrow \neg y) \lor (x \land \neg x)) \land (y \land z)$$

$$\Big\downarrow \texttt{collect\_ids}$$

$$\texttt{[}x\texttt{; }y\texttt{; }z\texttt{]}$$

$$\Big\downarrow \texttt{collect\_vals}$$

$(x \texttt{ !-> true ;; } y \texttt{ !-> true ;; } z \texttt{ !-> true)}^{\text{X}}$; $(x \texttt{ !-> true ;; } y \texttt{ !-> true)}^{\text{X}}$;
$(x \texttt{ !-> true ;; } z \texttt{ !-> true)}^{\text{X}}$; $(x \texttt{ !-> true)}^{\text{X}}$; $(y \texttt{ !-> true ;; } z \texttt{ !-> true)}^{\checkmark}$;
$(y \texttt{ !-> true)}^{?}$; $(z \texttt{ !-> true)}^{?}$; $\texttt{empty\_valuation}^{?}$

$$\Big\downarrow \texttt{find\_valuation}$$

$$\texttt{Some (}y\texttt{ !-> true ;; }z\texttt{ !-> true)}$$

$$\Big\downarrow \texttt{solver}$$

$$\texttt{true}$$

Solver Soundness and Completeness

Soundness

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

Solver Soundness and Completeness

Soundness

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

Completeness

- **Lemma.** $\forall p : \mathit{satisfiable}\ p \implies \mathit{solver}\ p = \mathit{true}$
- $\texttt{interp}\ v\ p = \texttt{true} \not\Longrightarrow v \in \texttt{collect\_vals}\ (\texttt{collect\_ids}\ p))$

Solver Soundness and Completeness

Soundness

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

Completeness

- **Lemma.** $\forall\, p : \mathit{satisfiable}\ p \Longrightarrow \mathit{solver}\ p = \mathit{true}$
- $\texttt{interp}\ v\ \ p = \texttt{true} \not\Longrightarrow v \in \texttt{collect\_vals}\,(\texttt{collect\_ids}\ \ p))$
- Proof attempt:
  - $\forall\, v, v' : \forall\, x \in p, v\, x = v'\, x \Longrightarrow \texttt{interp}\ v\ p = \texttt{interp}\ v'\ p$
  - $\forall v : \exists v' : \forall\, x \in p, v\, x = v'\, x$ and $v' \in \texttt{collect\_vals}\,(\texttt{collect\_ids}\ \ p))$

Solver Decision Procedure

Theorem

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

Solver Decision Procedure

Theorem

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

i.e., `solver` decision procedure for SAT