INCREMENTAL INPROCESSING IN SAT SOLVING

Katalin Fazekas[1], Armin Biere[1], Christoph Scholl[2]

July 9, 2019, Lisbon

[1]Johannes Kepler University Linz, Austria

[2]Albert–Ludwigs–University, Freiburg, Germany

INTRODUCTION

Boolean Satisfiability Problem (SAT)

- Propositional logic

Boolean Satisfiability Problem (SAT)

- Propositional logic

$$(a \vee \neg b) \wedge (a \vee b) \wedge (\neg a \vee \neg b)$$

Boolean Satisfiability Problem (SAT)

- Propositional logic
- NP-complete problem: Is this set of clauses satisfiable?

$$(a \lor \neg b) \land (a \lor b) \land (\neg a \lor \neg b)$$

Boolean Satisfiability Problem (SAT)

- Propositional logic
- NP-complete problem: Is this set of clauses satisfiable?
  - If yes: Provide satisfying truth assignment

$$(a \vee \neg b) \wedge (a \vee b) \wedge (\neg a \vee \neg b)$$

Boolean Satisfiability Problem (SAT)

- Propositional logic
- NP-complete problem: Is this set of clauses satisfiable?
  - If yes: Provide satisfying truth assignment

$$\{a = \top, b = \bot\}$$

$$(a \vee \neg b) \wedge (a \vee b) \wedge (\neg a \vee \neg b)$$

Boolean Satisfiability Problem (SAT)

- Propositional logic
- NP-complete problem: Is this set of clauses satisfiable?
  - If yes: Provide satisfying truth assignment

$$\{a = \top, b = \bot\} = \{a, \neg b\}$$

$$(a \vee \neg b) \wedge (a \vee b) \wedge (\neg a \vee \neg b)$$

Boolean Satisfiability Problem (SAT)

- Propositional logic
- NP-complete problem: Is this set of clauses satisfiable?
  - If yes: Provide satisfying truth assignment

$$\{a = \top, b = \bot\} = \{a, \neg b\}$$

$$(a \vee \neg b) \wedge (a \vee b) \wedge (\neg a \vee \neg b)$$

Incremental SAT Solving

$C_1 \wedge \dots \wedge C_n$ $\qquad$ $SAT?$

Incremental SAT Solving

$\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}$ $\qquad$ $SAT?$

Incremental SAT Solving

$$\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0}$$

$SAT?$

Incremental SAT Solving

$$\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge C'_1 \wedge \cdots \wedge C'_{n'} \qquad SAT?$$

Incremental SAT Solving

$$\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1} \qquad SAT?$$

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{C_1 \land \cdots \land C_n}_{\Delta_0}}_{F^0} \land \underbrace{C'_1 \land \cdots \land C'_{n'}}_{\Delta_1}}_{F^1}$$

$SAT?$

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1}}_{F^1} \wedge C''_1 \wedge \cdots \wedge C''_{n''} \quad SAT?$$

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1}}_{F^1} \wedge \underbrace{C''_1 \wedge \cdots \wedge C''_{n''}}_{\Delta_2} \quad SAT?$$

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1}}_{F^1} \wedge \underbrace{C''_1 \wedge \cdots \wedge C''_{n''}}_{\Delta_2}}_{F^2} \quad SAT?$$

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1}}_{F^1} \wedge \underbrace{C''_1 \wedge \cdots \wedge C''_{n''}}_{\Delta_2}}_{F^2} \quad SAT?$$

For each $i = 0 \ldots m$ is $F^i = \bigwedge_{d=0}^{i} \Delta_d$ satisfiable?

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{\underbrace{C_1 \land \cdots \land C_n}_{\Delta_0}}_{F^0} \land \underbrace{C'_1 \land \cdots \land C'_{n'}}_{\Delta_1}}_{F^1} \land \underbrace{C''_1 \land \cdots \land C''_{n''}}_{\Delta_2}}_{F^2} \quad SAT?$$

For each $i = 0 \dots m$ is $F^i = \bigwedge_{d=0}^{i} \Delta_d$ satisfiable?

- Extend formula with new clauses

Incremental SAT Solving

$$\underbrace{\underbrace{\underbrace{\underbrace{C_1 \wedge \cdots \wedge C_n}_{\Delta_0}}_{F^0} \wedge \underbrace{C'_1 \wedge \cdots \wedge C'_{n'}}_{\Delta_1}}_{F^1} \wedge \underbrace{C''_1 \wedge \cdots \wedge C''_{n''}}_{\Delta_2}}_{F^2} \quad SAT?$$

For each $i = 0 \dots m$ is $F^i = \bigwedge_{d=0}^{i} \Delta_d$ satisfiable?

- Extend formula with new clauses
- Avoid repeated work
  - Keep gathered information (e.g. scores, search state variables)
  - Keep learned clauses

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

- Satisfiability preserving clause addition or removal

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

- Satisfiability preserving clause addition or removal
- Abstract framework that captures generally inprocessing

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

- Satisfiability preserving clause addition or removal
- Abstract framework that captures generally inprocessing
- Deduction rules applied on abstract states

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses σ: Eliminated clauses

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	σ: Eliminated clauses
$a \vee b$		
$\neg a \vee b$		
a		

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	σ: Eliminated clauses
$a \vee b$		
$\neg a \vee b$		
a		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	σ: Eliminated clauses
$a \lor b$	b	
$\neg a \lor b$		
a		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \land C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \land \rho \equiv_{sat} \varphi \land \rho \land C$

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	$\boldsymbol{\sigma}$: Eliminated clauses
$\boldsymbol{a \vee b}$	$\boldsymbol{b}$	
$\boldsymbol{\neg a \vee b}$	$\boldsymbol{a \vee c}$	
$\boldsymbol{a}$		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	σ: Eliminated clauses
$a \vee b$	b	
$\neg a \vee b$	$a \vee c$	
a		

$$\frac{\varphi\ [\,\rho \wedge C\,]\ \sigma}{\varphi \wedge C\ [\,\rho\,]\ \sigma}$$

STRENGTHEN

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses $\boldsymbol{\sigma}$: Eliminated clauses

φ	ρ	σ
$a \lor b$	$\cancel{b}$	
$\neg a \lor b$	$a \lor c$	
a		
b		

$$\frac{\varphi\ [\,\rho \land C\,]\ \sigma}{\varphi \land C\ [\,\rho\,]\ \sigma}$$

STRENGTHEN

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

$\boldsymbol{\varphi}$: Irredundant clauses $\boldsymbol{\rho}$: Redundant clauses $\boldsymbol{\sigma}$: Eliminated clauses

$\boldsymbol{a \lor b}$	$\cancel{\boldsymbol{b}}$	
$\boldsymbol{\neg a \lor b}$	$\boldsymbol{a \lor c}$	
$\boldsymbol{a}$		
$\boldsymbol{b}$		

$$\frac{\varphi\ [\,\rho \land C\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma}$$

FORGET

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses σ: Eliminated clauses

φ	ρ	σ
$a \vee b$	$\cancel{b}$	
$\neg a \vee b$	$\cancel{a \vee c}$	
a		
b		

$$\frac{\varphi\ [\,\rho \wedge C\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma}$$

FORGET

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses	ρ: Redundant clauses	σ: Eliminated clauses
$a \lor b$	$\cancel{b}$	
$\neg a \lor b$	$\cancel{a \lor c}$	
a		
b		

$$\frac{\varphi \land C \; [\, \rho \,] \; \sigma}{\varphi \; [\, \rho \land C \,] \; \sigma \cdot (l : C)} \; \boxed{\flat}$$

WEAKEN

where $\boxed{\flat}$ is $\varphi \land C \equiv^{l}_{sat} \varphi$

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses σ: Eliminated clauses

φ	ρ	σ
$\cancel{a \vee b}$	$\cancel{b}$	$b : (a \vee b)$
$\neg a \vee b$	$\cancel{a \vee c}$	
a	$a \vee b$	
b		

$$\frac{\varphi \wedge C \; [\, \rho \,] \; \sigma}{\varphi \; [\, \rho \wedge C \,] \; \sigma \cdot (l : C)} \; \boxed{\flat}$$

WEAKEN

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{l}_{sat} \varphi$

Inprocessing in SAT Solving [JärvisaloHeuleBiere-IJCAR'12]

φ: Irredundant clauses ρ: Redundant clauses σ: Eliminated clauses

φ	ρ	σ
$\cancel{a \vee b}$	$\cancel{b}$	$b : (a \vee b)$
$\neg a \vee b$	$\cancel{a \vee c}$	
a	$a \vee b$	
b		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp} \qquad \frac{\varphi\ [\,\rho \wedge C\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma} \qquad \frac{\varphi\ [\,\rho \wedge C\,]\ \sigma}{\varphi \wedge C\ [\,\rho\,]\ \sigma} \qquad \frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma \cdot (l : C)}\ \boxed{\flat}$$

LEARN FORGET STRENGTHEN WEAKEN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$ and $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\ell}_{sat} \varphi$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

- Inprocessing is satisfiability but not model preserving

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

- Inprocessing is satisfiability but not model preserving
- Solution reconstruction is necessary

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \boldsymbol{b}) \wedge (\boldsymbol{a})$$

$\boldsymbol{a} \vee \boldsymbol{b}$

$\neg \boldsymbol{a} \vee \boldsymbol{b}$

$\boldsymbol{a}$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \lor b) \land (\neg a \lor b) \land (a)$$

$\cancel{a \lor b}$	$a \lor b$	$b : (a \lor b)$
$\neg a \lor b$		
a		

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \lor b) \land (\neg a \lor b) \land (a)$$

$\cancel{a \lor b}$	$a \lor b$	$b : (a \lor b)$
$\cancel{\neg a \lor b}$	$\neg a \lor b$	$b : (\neg a \lor b)$
a		

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (\boxed{a} \lor b) \land (\neg a \lor b) \land (\boxed{a})$$

$\cancel{a \lor b}$	$a \lor b$	$b : (a \lor b)$
$\cancel{\neg a \lor b}$	$\neg a \lor b$	$b : (\neg a \lor b)$
a		

$$\tau = \{a = \top, b = \bot\}$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \boldsymbol{b}) \wedge (\boldsymbol{a})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{b} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{b} : (\neg \boldsymbol{a} \vee \boldsymbol{b})$
$\boldsymbol{a}$		

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}$$

$$\mathcal{R}(\tau, \varepsilon) = \tau, \qquad \mathcal{R}(\tau, \sigma \cdot (\omega : D)) = \begin{cases} \mathcal{R}(\tau, \sigma) & \text{if } \tau(D) = \top \\ \mathcal{R}((\tau \circ \omega), \sigma) & \text{otherwise} \end{cases}$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \vee b) \wedge (\neg a \vee b) \wedge (a)$$

$\cancel{a \vee b}$	$a \vee b$	$b : (a \vee b)$
$\cancel{\neg a \vee b}$	$\neg a \vee b$	$b : (\neg a \vee b)$
a		

$$\tau = \{a = \top, b = \bot\}$$

$$\mathcal{R}(\{a = \top, b = \bot\}, (b : (a \vee b)) \cdot (b : (\neg a \vee b)))$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \vee b) \wedge (\neg a \vee b) \wedge (a)$$

$\cancel{a \vee b}$	$a \vee b$	$b : (a \vee b)$
$\cancel{\neg a \vee b}$	$\neg a \vee b$	$b : (\neg a \vee b)$
a		

$$\tau = \{a = \top, b = \bot\}$$

$$\mathcal{R}(\{a = \top, b = \bot\}, (b : (a \vee b)) \cdot (b : (\neg a \vee b))) =$$
$$\mathcal{R}(\{a = \top, b = \top\}, (b : (a \vee b)))$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \vee b) \wedge (\neg a \vee b) \wedge (a)$$

$\cancel{a \vee b}$	$a \vee b$	$b : (a \vee b)$
$\cancel{\neg a \vee b}$	$\neg a \vee b$	$b : (\neg a \vee b)$
a		

$$\tau = \{a = \top, b = \bot\}$$

$$\mathcal{R}(\{a = \top, b = \bot\}, (b : (a \vee b)) \cdot (b : (\neg a \vee b))) =$$
$$\mathcal{R}(\{a = \top, b = \top\}, (b : (a \vee b)))$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \boldsymbol{b}) \wedge (\boldsymbol{a})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{b} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{b} : (\neg \boldsymbol{a} \vee \boldsymbol{b})$
$\boldsymbol{a}$		

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}$$

$$\begin{aligned}
&\boldsymbol{\mathcal{R}}(\{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}, (\boldsymbol{b} : (\boldsymbol{a} \vee \boldsymbol{b})) \cdot (\boldsymbol{b} : (\neg \boldsymbol{a} \vee \boldsymbol{b}))) = \\
&\boldsymbol{\mathcal{R}}(\{\boldsymbol{a} = \top, \boldsymbol{b} = \top\}, (\boldsymbol{b} : (\boldsymbol{a} \vee \boldsymbol{b}))) = \\
&\boldsymbol{\mathcal{R}}(\{\boldsymbol{a} = \top, \boldsymbol{b} = \top\}, \varepsilon) = \{\boldsymbol{a} = \top, \boldsymbol{b} = \top\}.
\end{aligned}$$

Solution Reconstruction [Sörensson 2009, JHB-IJCAR'12]

$$\mathcal{F}^0 = (a \lor b) \land (\neg a \lor b) \land (a)$$

$$\begin{array}{c|c|c}
\cancel{a \lor b} & a \lor b & b : (a \lor b) \\
\cancel{\neg a \lor b} & \neg a \lor b & b : (\neg a \lor b) \\
a & &
\end{array}$$

$$\tau = \{a = \top, b = \bot\}$$

$$\begin{aligned}
&\mathcal{R}(\{a = \top, b = \bot\}, (b : (a \lor b)) \cdot (b : (\neg a \lor b))) = \\
&\mathcal{R}(\{a = \top, b = \top\}, (b : (a \lor b))) = \\
&\mathcal{R}(\{a = \top, b = \top\}, \varepsilon) = \{a = \top, b = \top\}.
\end{aligned}$$

INCREMENTAL INPROCESSING

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg\boldsymbol{a} \vee \neg\boldsymbol{b})$$

$\boldsymbol{a} \vee \boldsymbol{b}$		
$\neg\boldsymbol{a} \vee \neg\boldsymbol{b}$		

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \lor \boldsymbol{b}) \land (\neg \boldsymbol{a} \lor \neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \lor \boldsymbol{b}}$	$\boldsymbol{a} \lor \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \lor \boldsymbol{b})$
$\neg \boldsymbol{a} \lor \neg \boldsymbol{b}$		

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg\boldsymbol{a} \vee \neg\boldsymbol{b})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg\boldsymbol{a} \vee \neg\boldsymbol{b}}$	$\neg\boldsymbol{a} \vee \neg\boldsymbol{b}$	$\neg\boldsymbol{b} : (\neg\boldsymbol{a} \vee \neg\boldsymbol{b})$

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}$$

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}$$

$$\mathcal{R}(\{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}, (\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})) \cdot (\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b}))) = \{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}$$

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \neg \boldsymbol{b}) \quad \mathcal{F}^1 = \mathcal{F}^0 \wedge (\neg \boldsymbol{a}) \wedge (\neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$
$\neg \boldsymbol{a}$		
$\neg \boldsymbol{b}$		

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \lor \boldsymbol{b}) \land (\neg \boldsymbol{a} \lor \neg \boldsymbol{b}) \quad \mathcal{F}^1 = \mathcal{F}^0 \land (\neg \boldsymbol{a}) \land (\neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \lor \boldsymbol{b}}$	$\boldsymbol{a} \lor \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \lor \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \lor \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \lor \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \lor \neg \boldsymbol{b})$
$\neg \boldsymbol{a}$		
$\neg \boldsymbol{b}$		

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}$$

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \vee \boldsymbol{b}) \wedge (\neg \boldsymbol{a} \vee \neg \boldsymbol{b}) \quad \mathcal{F}^1 = \mathcal{F}^0 \wedge (\neg \boldsymbol{a}) \wedge (\neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \vee \boldsymbol{b}}$	$\boldsymbol{a} \vee \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \vee \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \vee \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b})$
$\neg \boldsymbol{a}$		
$\neg \boldsymbol{b}$		

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}$$

$$\mathcal{R}(\{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}, (\boldsymbol{a} : (\boldsymbol{a} \vee \boldsymbol{b})) \cdot (\neg \boldsymbol{b} : (\neg \boldsymbol{a} \vee \neg \boldsymbol{b}))) = \{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}$$

Incremental Inprocessing – Problem

$$\mathcal{F}^0 = (\boldsymbol{a} \lor \boldsymbol{b}) \land (\neg \boldsymbol{a} \lor \neg \boldsymbol{b}) \quad \mathcal{F}^1 = \mathcal{F}^0 \land (\neg \boldsymbol{a}) \land (\neg \boldsymbol{b})$$

$\cancel{\boldsymbol{a} \lor \boldsymbol{b}}$	$\boldsymbol{a} \lor \boldsymbol{b}$	$\boldsymbol{a} : (\boldsymbol{a} \lor \boldsymbol{b})$
$\cancel{\neg \boldsymbol{a} \lor \neg \boldsymbol{b}}$	$\neg \boldsymbol{a} \lor \neg \boldsymbol{b}$	$\neg \boldsymbol{b} : (\neg \boldsymbol{a} \lor \neg \boldsymbol{b})$
$\neg \boldsymbol{a}$		
$\neg \boldsymbol{b}$		

$$\boldsymbol{\tau} = \{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}$$

$$\mathcal{R}(\{\boldsymbol{a} = \bot, \boldsymbol{b} = \bot\}, (\boldsymbol{a} : (\boldsymbol{a} \lor \boldsymbol{b})) \cdot (\neg \boldsymbol{b} : (\neg \boldsymbol{a} \lor \neg \boldsymbol{b}))) = \{\boldsymbol{a} = \top, \boldsymbol{b} = \bot\}$$

Incremental Inprocessing – Possible solutions

- Forbid inprocessing partially:
  - Freeze & Melt – 'Don't touch' variables
    [EénSörensson-ENTCS'03, KupferschmidLewisSchubertBecker-FMSD'11]

Incremental Inprocessing – Possible solutions

- Forbid inprocessing partially:
  - Freeze & Melt – 'Don't touch' variables
    [EénSörensson-ENTCS'03, KupferschmidLewisSchubertBecker-FMSD'11]
- Preprocessing in incremental SAT [NadelRyvchinStrichman-SAT'12]
  - Variable elimination, (self-)subsumption

Incremental Inprocessing – Possible solutions

- Forbid inprocessing partially:
  - Freeze & Melt – 'Don't touch' variables
    [EénSörensson-ENTCS'03, KupferschmidLewisSchubertBecker-FMSD'11]
- Preprocessing in incremental SAT [NadelRyvchinStrichman-SAT'12]
  - Variable elimination, (self-)subsumption
- General solution: **Incremental Inprocessing**
  - Adapt and extend inprocessing rules

Constrained Learning

$\boldsymbol{a} \vee \boldsymbol{b}$

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

Constrained Learning

$\boldsymbol{a} \vee \boldsymbol{b}$ | $\neg \boldsymbol{a} \vee \neg \boldsymbol{b}$ |

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

Constrained Learning

$\boldsymbol{a \vee b}$	$\boldsymbol{\neg a \vee \neg b}$	
$\boldsymbol{a}$		
$\boldsymbol{b}$		

$$\frac{\varphi\,[\,\rho\,]\;\sigma}{\varphi\,[\,\rho \wedge C\,]\;\sigma}\;\boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

Constrained Learning

$a \lor b$	$\neg a \lor \neg b$	
a		
b		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \land C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \land \rho \equiv_{sat} \varphi \land \rho \land C$

$\Longrightarrow$

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \land C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN^{-}

where $\boxed{\sharp}$ is $\varphi \land \rho \models C$

Constrained Learning

$a \vee b$	$\neg a \vee \neg b$	
a		
b		

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN

where $\boxed{\sharp}$ is $\varphi \wedge \rho \equiv_{sat} \varphi \wedge \rho \wedge C$

$$\Longrightarrow$$

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma}\ \boxed{\sharp}$$

LEARN^{-}

where $\boxed{\sharp}$ is $\varphi \wedge \rho \models C$

- Weaker than original LEARN
  - No extended resolution (e.g. blocked clause addition)

Stronger Weakening

$$\frac{\varphi \wedge C \; [\, \rho \,] \; \sigma}{\varphi \; [\, \rho \wedge C \,] \; \sigma \cdot (l : C)} \; \boxed{\flat}$$

WEAKEN

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{l}_{sat} \varphi$

Stronger Weakening

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma \cdot (l : C)}\ \boxed{\flat}$$

WEAKEN

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{l}_{sat} \varphi$

$\Nearrow$

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma}\ \boxed{\varnothing}$$

DROP

where $\boxed{\varnothing}$ is $\varphi \models C$

Stronger Weakening

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho \wedge C\,]\ \sigma \cdot (l : C)}\ \boxed{\flat}$$

Weaken

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{l}_{sat} \varphi$

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma}\ \boxed{\varnothing}$$

Drop

where $\boxed{\varnothing}$ is $\varphi \models C$

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma \cdot (\omega : C)}\ \boxed{\flat}$$

Weaken^{+}

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$

Stronger Weakening

$$\frac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma \cdot (\omega : C)}\ \boxed{\flat}$$

$$\textsc{Weaken}^{+}$$

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$

Stronger Weakening

$$\dfrac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma \cdot (\omega : C)}\ \boxed{\flat}$$

$$\textsc{Weaken}^{+}$$

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$

- Syntax: ω is set of literals s.t. $\omega \cap C \neq \emptyset$

Stronger Weakening

$$\frac{\varphi \wedge C \; [\rho] \; \sigma}{\varphi \; [\rho] \; \sigma \cdot (\omega : C)} \; \boxed{\flat} \qquad \textsc{Weaken}^{+}$$

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$

- Syntax: ω is set of literals s.t. $\omega \cap C \neq \emptyset$
- Semantics: Propagation Redundancy [HeuleKieslBiere-CADE'17]

Stronger Weakening

$$\frac{\varphi \wedge C\ [\,\rho\,]\ \sigma}{\varphi\ [\,\rho\,]\ \sigma \cdot (\omega : C)}\ \boxed{\flat}$$

WEAKEN^{+}

where $\boxed{\flat}$ is $\varphi \wedge C \equiv_{sat}^{\omega} \varphi$

- Syntax: ω is set of literals s.t. $\omega \cap C \neq \emptyset$
- Semantics: Propagation Redundancy [HeuleKiesIBiere-CADE'17]
  - Most general **reconstructive** redundancy property

Stronger Weakening

$$\frac{\varphi \wedge C \; [\,\rho\,] \; \sigma}{\varphi \; [\,\rho\,] \; \sigma \cdot (\omega : C)} \; \boxed{\flat} \quad \textsc{Weaken}^{+}$$

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$

- Syntax: ω is set of literals s.t. $\omega \cap C \neq \emptyset$
- Semantics: Propagation Redundancy [HeuleKiesIBiere-CADE'17]
  - Most general **reconstructive** redundancy property
  - Polynomially reconstructible via witness ω:

Proposition: If $\tau(\varphi) = \top$ and $\tau(C) \neq \top$ then $(\tau \circ \omega)(\varphi \wedge C) = \top$

Incremental Clause Addition

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi \wedge \Delta\ [\,\rho\,]\ \sigma}\ \boxed{\mathcal{I}}$$

ADDCLAUSES

where $\boxed{\mathcal{I}}$ is that each clause of Δ is clean w.r.t. σ

Incremental Clause Addition

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi \wedge \Delta\ [\,\rho\,]\ \sigma}\ \boxed{\mathcal{I}}$$

ADDCLAUSES

where $\boxed{\mathcal{I}}$ is that each clause of Δ is **clean** w.r.t. σ

A clause C is **clean** w.r.t. a sequence of witness labelled clauses σ if and only if for all $(\omega : D) \in \sigma$ we have that $\neg C \cap \omega = \emptyset$.

Incremental Clause Addition

$$\frac{\varphi\ [\,\rho\,]\ \sigma}{\varphi \wedge \Delta\ [\,\rho\,]\ \sigma}\ \boxed{\mathcal{I}}$$

ADDCLAUSES

where $\boxed{\mathcal{I}}$ is that each clause of Δ is **clean** w.r.t. σ

A clause C is **clean** w.r.t. a sequence of witness labelled clauses σ if and only if for all $(\omega : D) \in \sigma$ we have that $\neg C \cap \omega = \emptyset$.

Lemma: If a clause C is clean on a sequence of witness labelled clauses σ, then for all truth assignments τ with $\tau(C) = \top$ we have that $\mathcal{R}(\tau, \sigma)(C) = \top$.

Reversing Weakenings

$$\frac{\varphi\ [\,\rho\,]\ \sigma \cdot (\omega : C) \cdot \sigma'}{\varphi \wedge C\ [\,\rho\,]\ \sigma \cdot \sigma'}\ \boxed{\partial}$$

RESTORE

where $\boxed{\partial}$ is C is clean w.r.t. σ'

Incremental Inprocessing Rules

$$\frac{\varphi\ [\rho]\ \sigma}{\varphi\ [\rho \wedge C]\ \sigma}\ \boxed{\sharp}$$

LEARN$^-$

$$\frac{\varphi\ [\rho \wedge C]\ \sigma}{\varphi\ [\rho]\ \sigma}$$

FORGET

$$\frac{\varphi\ [\rho \wedge C]\ \sigma}{\varphi \wedge C\ [\rho]\ \sigma}$$

STRENGTHEN

$$\frac{\varphi\ [\rho]\ \sigma}{\varphi \wedge \Delta\ [\rho]\ \sigma}\ \boxed{\mathcal{I}}$$

ADDCLAUSES

$$\frac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma \cdot (\omega : C)}\ \boxed{\flat}$$

WEAKEN$^+$

$$\frac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma}\ \boxed{\emptyset}$$

DROP

$$\frac{\varphi\ [\rho]\ \sigma \cdot (\omega : C) \cdot \sigma'}{\varphi \wedge C\ [\rho]\ \sigma \cdot \sigma'}\ \boxed{\partial}$$

RESTORE

where $\boxed{\sharp}$ is $\varphi \wedge \rho \models C$, $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$, $\boxed{\emptyset}$ is $\varphi \models C$,

$\boxed{\partial}$ is C is clean w.r.t. σ' and $\boxed{\mathcal{I}}$ is that each clause in Δ is clean w.r.t. σ

Incremental Inprocessing Rules

$$\frac{\varphi\;[\rho]\;\sigma}{\varphi\;[\rho \wedge C]\;\sigma}\;\boxed{\sharp}$$

LEARN^{-}

$$\frac{\varphi\;[\rho \wedge C]\;\sigma}{\varphi\;[\rho]\;\sigma}$$

FORGET

$$\frac{\varphi\;[\rho \wedge C]\;\sigma}{\varphi \wedge C\;[\rho]\;\sigma}$$

STRENGTHEN

$$\frac{\varphi\;[\rho]\;\sigma}{\varphi \wedge \Delta\;[\rho]\;\sigma}\;\boxed{\mathcal{I}}$$

ADDCLAUSES

$$\frac{\varphi \wedge C\;[\rho]\;\sigma}{\varphi\;[\rho]\;\sigma \cdot (\omega : C)}\;\boxed{\flat}$$

WEAKEN^{+}

$$\frac{\varphi \wedge C\;[\rho]\;\sigma}{\varphi\;[\rho]\;\sigma}\;\boxed{\varnothing}$$

DROP

$$\frac{\varphi\;[\rho]\;\sigma \cdot (\omega : C) \cdot \sigma'}{\varphi \wedge C\;[\rho]\;\sigma \cdot \sigma'}\;\boxed{\partial}$$

RESTORE

where $\boxed{\sharp}$ is $\varphi \wedge \rho \models C$, $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$, $\boxed{\varnothing}$ is $\varphi \models C$,

$\boxed{\partial}$ is C is clean w.r.t. σ' and $\boxed{\mathcal{I}}$ is that each clause in Δ is clean w.r.t. σ

Incremental Inprocessing – Formal correctness

- Gathered information (including learned clauses) can be kept

$\boldsymbol{F^{i+1} \models \rho^{i}_{k_i}}$ where $\rho^i_{k_i}$ is ρ at the end of the evaluation of F^i

Incremental Inprocessing – Formal correctness

- Gathered information (including learned clauses) can be kept

$$\boldsymbol{F^{i+1} \models \rho^{i}_{k_i}} \text{ where } \rho^{i}_{k_i} \text{ is } \rho \text{ at the end of the evaluation of } F^{i}$$

- Satisfiability preserving derivation continuation

$$\boldsymbol{F^{i} \equiv_{sat} \varphi^{i}_{j} \wedge \rho^{i}_{j}} \text{ for all } j \text{ with } 0 \leq j \leq k_i$$

Incremental Inprocessing – Formal correctness

- Gathered information (including learned clauses) can be kept

$$\boldsymbol{F^{i+1} \models \rho^i_{k_i}} \text{ where } \rho^i_{k_i} \text{ is } \rho \text{ at the end of the evaluation of } F^i$$

- Satisfiability preserving derivation continuation

$$\boldsymbol{F^i \equiv_{sat} \varphi^i_j \wedge \rho^i_j} \text{ for all } j \text{ with } 0 \leq j \leq k_i$$

- Solution reconstruction in any satisfiable state

$$\boldsymbol{\tau(\varphi^i_{k_i}) = \top \implies \mathcal{R}(\tau, \sigma^i_{k_i})(F^i) = \top} \text{ for all } i \text{ with } 0 \leq i \leq m$$

IMPLEMENTATION

Algorithm to Restore and Add Clauses

AlgorithmRestoreAddClauses (new clauses Δ, reconstruction stack σ)

$(\omega_1 : C_1) \cdots (\omega_n : C_n) \;:=\; \sigma$

for i **from** 1 **to** n

    if exists $\ell \in \omega_i$ where $\neg\ell$ occurs in Δ **then**

        $\Delta \;:=\; \Delta \cup C_i\,, \quad \sigma \;:=\; \sigma \setminus (\omega_i : C_i)$

return $\langle \Delta, \sigma \rangle$

EXPERIMENTS

Experiments

CaDiCaL SAT solver with inprocessing [Biere-SATCompProc'18]

Experiments

CaDiCaL SAT solver with inprocessing [Biere-SATCompProc'18]

- variable elimination [EénBiere-SAT'05]
- vivification [PietteHamadiSais-ECAI'08, LuoLiXiaoManyLü-IJCAI'17]
- equivalent-literal substitution [AspvallPlassTarjan-IPL'79, Brafman-IJCAI'01]
- hyper-binary resolution [BacchusWinter-SAT'03]
- (self-)subsumption [EénBiere-SAT'05]
- blocked clause elimination [JärvisaloBiereHeule-TACAS'10]

Experiments

CaDiCaL SAT solver with inprocessing [Biere-SATCompProc'18]

- variable elimination [EénBiere-SAT'05]
- vivification [PietteHamadiSais-ECAI'08, LuoLiXiaoManyLü-IJCAI'17]
- equivalent-literal substitution [AspvallPlassTarjan-IPL'79, Brafman-IJCAI'01]
- hyper-binary resolution [BacchusWinter-SAT'03]
- (self-)subsumption [EénBiere-SAT'05]
- blocked clause elimination [JärvisaloBiereHeule-TACAS'10]

CaMiCaL: bounded model checker with CaDiCaL as back-end

Experiments

CaDiCaL SAT solver with inprocessing [Biere-SATCompProc'18]

- variable elimination [EénBiere-SAT'05]
- vivification [PietteHamadiSais-ECAI'08, LuoLiXiaoManyLü-IJCAI'17]
- equivalent-literal substitution [AspvallPlassTarjan-IPL'79, Brafman-IJCAI'01]
- hyper-binary resolution [BacchusWinter-SAT'03]
- (self-)subsumption [EénBiere-SAT'05]
- blocked clause elimination [JärvisaloBiereHeule-TACAS'10]

CaMiCaL: bounded model checker with CaDiCaL as back-end

- safety property track of HWMCC'17: 300 AIGER models

Experiments

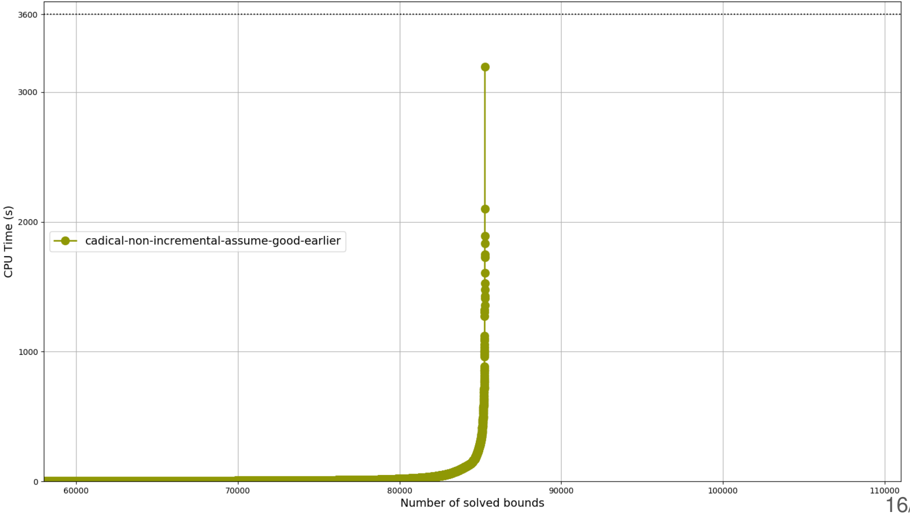
CaDiCaL SAT solver with inprocessing [Biere-SATCompProc'18]

- variable elimination [EénBiere-SAT'05]
- vivification [PietteHamadiSais-ECAI'08, LuoLiXiaoManyLü-IJCAI'17]
- equivalent-literal substitution [AspvallPlassTarjan-IPL'79, Brafman-IJCAI'01]
- hyper-binary resolution [BacchusWinter-SAT'03]
- (self-)subsumption [EénBiere-SAT'05]
- blocked clause elimination [JärvisaloBiereHeule-TACAS'10]

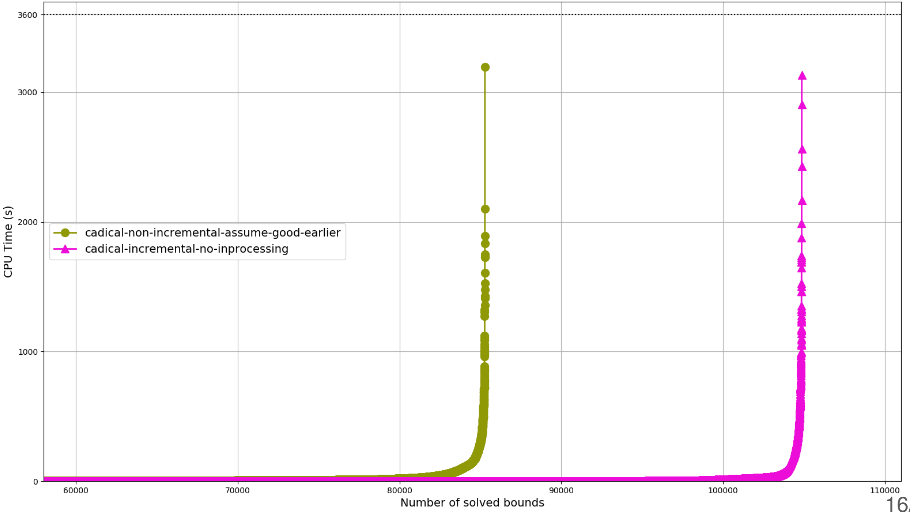
CaMiCaL: bounded model checker with CaDiCaL as back-end

- safety property track of HWMCC'17: 300 AIGER models
- solving a bound = (incremental) SAT call

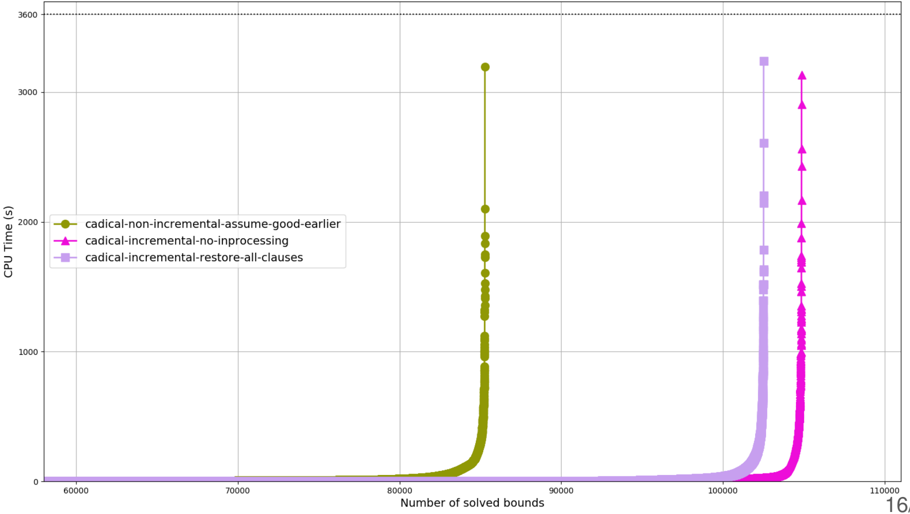
Experimental Results



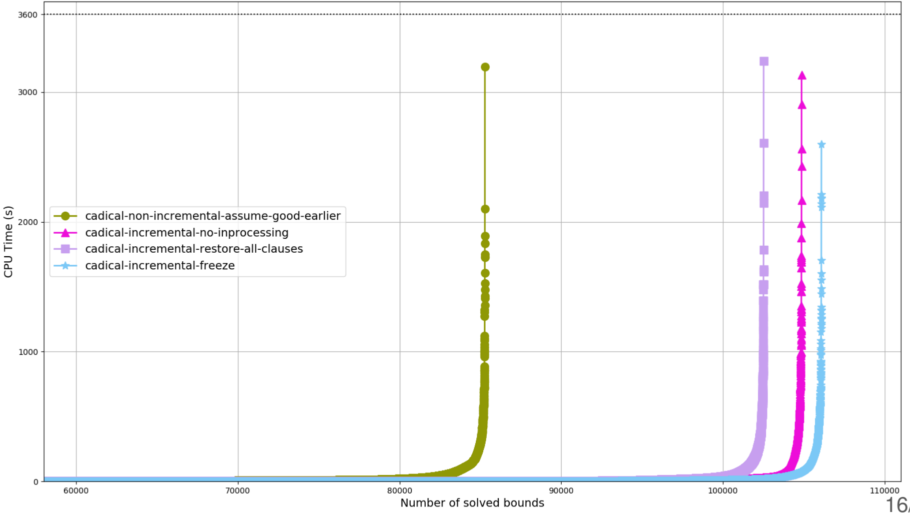
Experimental Results



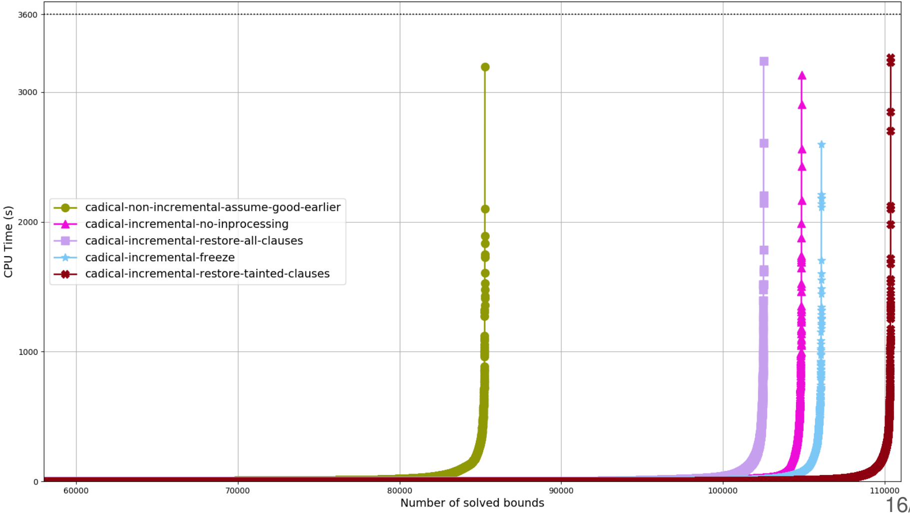
Experimental Results



Experimental Results



Experimental Results



Conclusion & Future Work

- Incremental inprocessing in SAT

Conclusion & Future Work

- Incremental inprocessing in SAT
- Simplified solver use
  - No need for variable freezing

Conclusion & Future Work

- Incremental inprocessing in SAT
- Simplified solver use
  - No need for variable freezing
- Simple and efficient implementation

Conclusion & Future Work

- Incremental inprocessing in SAT
- Simplified solver use
  - No need for variable freezing
- Simple and efficient implementation

Future Work:

- Limitations
  - LEARN^{-} (for e.g. blocked clause addition)

Conclusion & Future Work

- Incremental inprocessing in SAT
- Simplified solver use
  - No need for variable freezing
- Simple and efficient implementation

Future Work:

- Limitations
  - LEARN^{-} (for e.g. blocked clause addition)
  - Clean clause definition

Conclusion & Future Work

- Incremental inprocessing in SAT
- Simplified solver use
  - No need for variable freezing
- Simple and efficient implementation

Future Work:

- Limitations
  - Learn^- (for e.g. blocked clause addition)
  - Clean clause definition
- Inprocessing under assumptions

Thank you for your attention!

INCREMENTAL INPROCESSING IN SAT SOLVING

Katalin Fazekas[1], Armin Biere[1], Christoph Scholl[2]
July 9, 2019, Lisbon
[1]Johannes Kepler University Linz, Austria
[2]Albert–Ludwigs–University, Freiburg, Germany

JOHANNES KEPLER UNIVERSITY LINZ

UNI FREIBURG

logics LOGICAL METHODS IN COMPUTER SCIENCE

FWF

References I

[Aspvall et al., 1979] Aspvall, B., Plass, M. F., and Tarjan, R. E. (1979).
A linear-time algorithm for testing the truth of certain quantified boolean formulas.
Information Processing Letters, 8(3):121–123.

[Bacchus and Winter, 2003] Bacchus, F. and Winter, J. (2003).
Effective preprocessing with hyper-resolution and equality reduction.
In Giunchiglia, E. and Tacchella, A., editors, *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 341–355. Springer.

References II

[Biere, 2018] Biere, A. (2018).
CaDiCaL, Lingeling, Plingeling, Treengeling and YalSAT Entering the SAT Competition 2018.
In Heule, M., Järvisalo, M., and Suda, M., editors, *Proc. of SAT Competition 2018 – Solver and Benchmark Descriptions*, volume B-2018-1 of *Department of Computer Science Series of Publications B*, pages 13–14. University of Helsinki.

[Brafman, 2001] Brafman, R. I. (2001).
A simplifier for propositional formulas with many binary clauses.
In Nebel, B., editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, pages 515–522. Morgan Kaufmann.

References III

[Eén and Biere, 2005] Eén, N. and Biere, A. (2005).
Effective preprocessing in SAT through variable and clause elimination.
In Bacchus, F. and Walsh, T., editors, *Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings*, volume 3569 of *Lecture Notes in Computer Science*, pages 61–75. Springer.

[Eén and Sörensson, 2003] Eén, N. and Sörensson, N. (2003).
Temporal induction by incremental SAT solving.
Electr. Notes Theor. Comput. Sci., 89(4):543–560.

References IV

[Heule et al., 2019] Heule, M. J. H., Kiesl, B., and Biere, A. (2019).
Strong extension-free proof systems.
Journal of Automated Reasoning.
To be published.

[Järvisalo et al., 2010] Järvisalo, M., Biere, A., and Heule, M. (2010).
Blocked clause elimination.
In Esparza, J. and Majumdar, R., editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 129–144. Springer.

References V

[Järvisalo et al., 2012] Järvisalo, M., Heule, M., and Biere, A. (2012).
Inprocessing rules.
In Gramlich, B., Miller, D., and Sattler, U., editors, *Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, volume 7364 of *Lecture Notes in Computer Science*, pages 355–370. Springer.

[Kupferschmid et al., 2011] Kupferschmid, S., Lewis, M. D. T., Schubert, T., and Becker, B. (2011).
Incremental preprocessing methods for use in BMC.
Formal Methods in System Design, 39(2):185–204.

References VI

[Luo et al., 2017] Luo, M., Li, C., Xiao, F., Manyà, F., and Lü, Z. (2017).
An effective learnt clause minimization approach for CDCL SAT solvers.
In Sierra, C., editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 703–711. ijcai.org.

[Nadel et al., 2012] Nadel, A., Ryvchin, V., and Strichman, O. (2012).
Preprocessing in incremental SAT.
In Cimatti, A. and Sebastiani, R., editors, *Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*, volume 7317 of *Lecture Notes in Computer Science*, pages 256–269. Springer.

References VII

[Piette et al., 2008] Piette, C., Hamadi, Y., and Sais, L. (2008).
Vivifying propositional clausal formulae.
In Ghallab, M., Spyropoulos, C. D., Fakotakis, N., and Avouris, N. M., editors, *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 525–529. IOS Press.

Experimental Results