Computing Minimal Unsatisfiable Subsets of Clause Sets

Author: Shahriar Robbani
Supervision: Erika Ábrahám

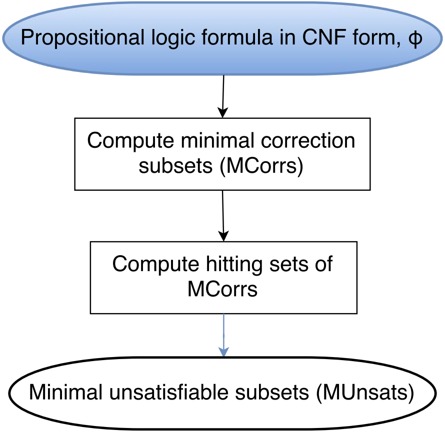
Theory of Hybrid Systems - Informatik 2 - RWTH-Aachen

Satisfiability Checking Seminar, Winter-16/17

Outline

- Propositional Logic Formula
- Minimal Unsatisfiable Subsets and Minimal Correction Subsets
- Duality of Minimal Unsatisfiable and Correction Subset
- Algorithms for Computing all Minimal Unsatisfiable Subsets

FlowChart


Propositional logic formula in CNF form, ϕ
Compute minimal correction subsets (MCorrs)
Compute hitting sets of MCorrs
Minimal unsatisfiable subsets (MUnsats)

Propositional Logic Formula

x

variable

Propositional Logic Formula

x $\neg x$

variable

Propositional Logic Formula

x $\qquad\qquad$ $\neg x$

literals

Propositional Logic Formula

$$\overbrace{x \quad \vee \quad \neg x}^{\textit{clause}}$$

literals

Propositional Logic Formula

$$\varphi := \overbrace{(x \quad \vee \quad \neg x)}^{\textit{clause}} \wedge (x \vee \neg y)$$

literals

$$\underbrace{(x \vee \neg x) \wedge (x \vee \neg y)}_{\text{conjunctive normal form}}$$

Problem Statement

$$\begin{aligned}\varphi &:= (x \vee \neg x) \wedge (\neg x \vee y)\\ &:= (T \vee F) \wedge (F \vee T)\\ &:= T \longrightarrow \text{SAT}\end{aligned}$$

Problem Statement

$$\begin{aligned}\varphi &:= (x \vee \neg x) \wedge (\neg x \vee y)\\ &:= (T \vee F) \wedge (F \vee T)\\ &:= T \longrightarrow \text{SAT}\end{aligned}$$

$$\begin{aligned}\varphi &:= (x) \wedge (\neg x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)\\ &:= (T) \wedge (F) \wedge (F \vee T) \wedge (F \vee F)\\ &:= F \longrightarrow \text{UNSAT}\end{aligned}$$

Problem Statement

$$\begin{aligned}\varphi &:= (x \vee \neg x) \wedge (\neg x \vee y)\\ &:= (T \vee F) \wedge (F \vee T)\\ &:= T \longrightarrow \text{SAT}\end{aligned}$$

$$\begin{aligned}\varphi &:= (x) \wedge (\neg x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)\\ &:= (T) \wedge (F) \wedge (F \vee T) \wedge (F \vee F)\\ &:= F \longrightarrow \text{UNSAT}\end{aligned}$$

unsatisfiable subset = $\{ \{(x), (\neg x), (\neg x \vee \neg y)\}$

Problem Statement

$$\begin{aligned}\varphi &:= (x \vee \neg x) \wedge (\neg x \vee y)\\ &:= (T \vee F) \wedge (F \vee T)\\ &:= T \longrightarrow \text{ SAT}\end{aligned}$$

$$\begin{aligned}\varphi &:= (x) \wedge (\neg x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)\\ &:= (T) \wedge (F) \wedge (F \vee T) \wedge (F \vee F)\\ &:= F \longrightarrow \text{ UNSAT}\end{aligned}$$

unsatisfiable subset = $\{\, \{(x), (\neg x), (\neg x \vee \neg y)\}$
$, \{(x), (\neg x \vee y), (\neg x \vee \neg y)\}$

Problem Statement

$$\begin{aligned}\varphi &:= (x \vee \neg x) \wedge (\neg x \vee y)\\ &:= (T \vee F) \wedge (F \vee T)\\ &:= T \longrightarrow \text{SAT}\end{aligned}$$

$$\begin{aligned}\varphi &:= (x) \wedge (\neg x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)\\ &:= (T) \wedge (F) \wedge (F \vee T) \wedge (F \vee F)\\ &:= F \longrightarrow \text{UNSAT}\end{aligned}$$

$$\begin{aligned}\text{unsatisfiable subset} \quad = \quad &\{\, \{(x), (\neg x), (\neg x \vee \neg y)\}\\ &,\{(x), (\neg x \vee y), (\neg x \vee \neg y)\}\\ &,\{(x), (\neg x)\}\,\}\end{aligned}$$

Minimal Unsatisfiable Subset (MUnsat):

(x)	$(\neg x)$	$(\neg x \vee y)$	$(\neg x \vee \neg y)$	MUnsat	MinimumUnsat
$\lightning$	$\lightning$			$\lightning$	$\lightning$
$\lightning$	$\lightning$		$\lightning$		
$\lightning$		$\lightning$	$\lightning$	$\lightning$	

$$\varphi = \underbrace{(x)}_{C_1} \wedge \underbrace{(\neg x)}_{C_2} \wedge \underbrace{(\neg x \vee y)}_{C_3} \wedge \underbrace{(\neg x \vee \neg y)}_{C_4}$$

Minimal Unsatisfiable Subsets and Minimal Correction Subsets

Minimal Unsatisfiable Subset (MUnsat):

(x)	$(\neg x)$	$(\neg x \vee y)$	$(\neg x \vee \neg y)$	MUnsat	MinimumUnsat
$\lightning$	$\lightning$			$\lightning$	$\lightning$
$\lightning$	$\lightning$		$\lightning$		
$\lightning$		$\lightning$	$\lightning$	$\lightning$	

Minimal Correction Subset (MCorr):

(x)	$(\neg x)$	$(\neg x \vee y)$	$(\neg x \vee \neg y)$
$\checkmark$			
	$\checkmark$	$\checkmark$	
	$\checkmark$		$\checkmark$

$$\varphi = \underbrace{(x)}_{C_1} \wedge \underbrace{(\neg x)}_{C_2} \wedge \underbrace{(\neg x \vee y)}_{C_3} \wedge \underbrace{(\neg x \vee \neg y)}_{C_4}$$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

Variable Set:	$\mathrm{D} = \{w, x, y, z\}$
Collection Set:	$\Omega = \{\{w, x\}, \{x, y, z\}\}$
Hitting Set:	$H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: $\mathrm{D} = \{w, x, y, z\}$

  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$

  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \dots\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

Variable Set:	$D = \{w, x, y, z\}$
Collection Set:	$\Omega = \{\{w, x\}, \{x, y, z\}\}$
Hitting Set:	$H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: $\mathrm{D} = \{w, x, y, z\}$

  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$

  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: D = $\{w, x, y, z\}$

  Collection Set: Ω = $\{\{w, x\}, \{x, y, z\}\}$

  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$

  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: $D = \{w, x, y, z\}$
  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$
  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$
  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: $D = \{w, x, y, z\}$
  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$
  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$
  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

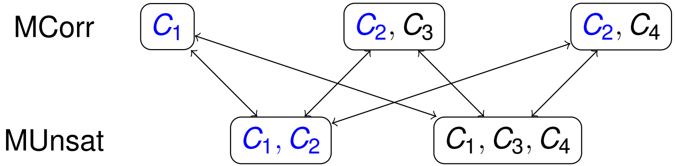
  Variable Set: D = $\{w, x, y, z\}$
  Collection Set: Ω = $\{\{w, x\}, \{x, y, z\}\}$
  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$
  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: $D = \{w, x, y, z\}$
  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$
  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$
  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

- Minimal hitting sets of the set of MCorr:



$$\varphi = \underbrace{(x)}_{C_1} \wedge \underbrace{(\neg x)}_{C_2} \wedge \underbrace{(\neg x \vee y)}_{C_3} \wedge \underbrace{(\neg x \vee \neg y)}_{C_4}$$

- **Hitting Sets:**

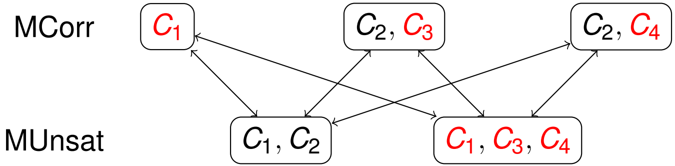
  Variable Set: $D = \{w, x, y, z\}$

  Collection Set: $\Omega = \{\{w, x\}, \{x, y, z\}\}$

  Hitting Set: $H = \{\{w, z\}, \{x, y\}, \{x\}, \ldots\}$

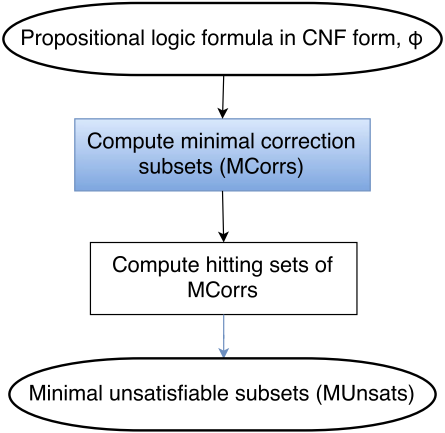
  Minimal Hitting Set: $MinH = \{\{w, y\}, \{w, z\}, \{x\}\}$

- Minimal hitting sets of the set of MCorr:



$$\varphi = \underbrace{(x)}_{C_1} \wedge \underbrace{(\neg x)}_{C_2} \wedge \underbrace{(\neg x \vee y)}_{C_3} \wedge \underbrace{(\neg x \vee \neg y)}_{C_4}$$

FlowChart


Propositional logic formula in CNF form, ϕ
Compute minimal correction subsets (MCorrs)
Compute hitting sets of MCorrs
Minimal unsatisfiable subsets (MUnsats)

Algorithm: Computing all MCorrs

1. Augment CNF with clause selector variables

$$\begin{array}{lcccccccc} \varphi = & (x) & \wedge & (\neg x) & \wedge & (\neg x \vee y) & \wedge & (\neg x \vee \neg y) \\ \varphi' = & (\neg w_1 \vee x) & \wedge & (\neg w_2 \vee \neg x) & \wedge & (\neg w_3 \vee \neg x \vee y) & \wedge & (\neg w_4 \vee \neg x \vee \neg y) \end{array}$$

Algorithm: Computing all MCorrs

1. Augment CNF with clause selector variables

$$\begin{array}{lccccccc} \varphi = & (x) & \wedge & (\neg x) & \wedge & (\neg x \vee y) & \wedge & (\neg x \vee \neg y) \\ \varphi' = & (\neg w_1 \vee x) & \wedge & (\neg w_2 \vee \neg x) & \wedge & (\neg w_3 \vee \neg x \vee y) & \wedge & (\neg w_4 \vee \neg x \vee \neg y) \end{array}$$

2. All MCorrs are found incrementally

$$\varphi' = (\neg \textbf{false} \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y)$$

Algorithm: Computing all MCorrs

1. Augment CNF with clause selector variables

$$\begin{array}{lccccccc} \varphi = & (x) & \wedge & (\neg x) & \wedge & (\neg x \vee y) & \wedge & (\neg x \vee \neg y) \\ \varphi' = & (\neg w_1 \vee x) & \wedge & (\neg w_2 \vee \neg x) & \wedge & (\neg w_3 \vee \neg x \vee y) & \wedge & (\neg w_4 \vee \neg x \vee \neg y) \end{array}$$

2. All MCorrs are found incrementally

$$\varphi' = (\neg \textbf{false} \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y)$$

3. Add blocking clauses to block old solutions

$$\varphi' = \varphi' \wedge w_1$$

Example: Computing all MCorrs

$$\varphi = (x) \wedge (\neg x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y)$$

Example: Computing all MCorrs

$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y)$$

1. Add clause-selector variables.

Example: Computing all MCorrs

$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y)$$

1. Add clause-selector variables.
2. Add AtMost constraint.

$$AtMost(\{w_1, w_2, w_3, w_4\}, 1)$$

Example: Computing all MCorrs

$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCorr.

MCorrs

1. $\{(x)\}$

$$AtMost(\{w_1, w_2, w_3, w_4\}, 1)$$

Example: Computing all MCorrs

$$\varphi \quad = \quad (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCorr.
4. No further solutions, increment AtMost.

MCorrs

1. $\{(x)\}$

$$AtMost(\{w_1, w_2, w_3, w_4\}, 2)$$

Example: Computing all MCorrs

$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1) \wedge (w_2 \vee w_3)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCorr.
4. No further solutions, increment AtMost.
5. Second solution : w_2 and w_3 are **false**. Add blocking clause and another MCorrs.

MCorrs

1. $\{(x)\}$
2. $\{(\neg x), (\neg x \vee y)\}$

$$AtMost(\{w_1, w_2, w_3, w_4\}, 2)$$

Example: Computing all MCorrs

$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1) \wedge (w_2 \vee w_3) \wedge (w_2 \vee w_4)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCorr.
4. No further solutions, increment AtMost.
5. Second solution : w_2 and w_3 are **false**. Add blocking clause and another MCorrs.
6. Third solution : w_2 and w_4 are **false**. Add blocking clause and another MCorrs.

MCorrs

1. $\{(x)\}$
2. $\{(\neg x), (\neg x \vee y)\}$
3. $\{(\neg x), (\neg x \vee \neg y)\}$

$$AtMost(\{w_1, w_2, w_3, w_4\}, 2)$$

Example: Computing all MCorrs

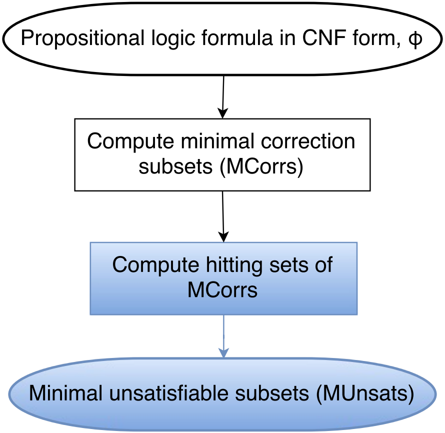
$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1) \wedge (w_2 \vee w_3) \wedge (w_2 \vee w_4)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCorr.
4. No further solutions, increment AtMost.
5. Second solution : w_2 and w_3 are **false**. Add blocking clause and another MCorrs.
6. Third solution : w_2 and w_4 are **false**. Add blocking clause and another MCorrs.
7. No further solutions, even without AtMost constraint.

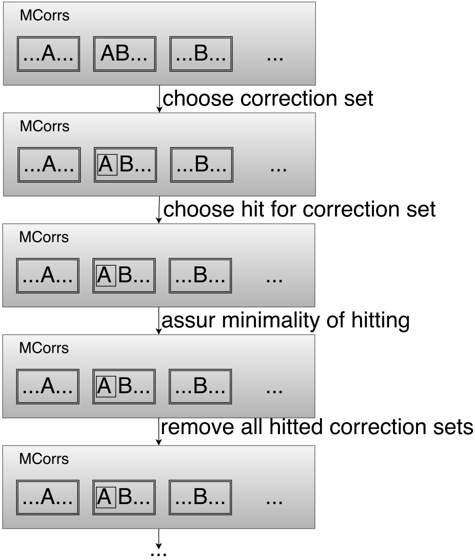
MCorrs

1. $\{(x)\}$
2. $\{(\neg x), (\neg x \vee y)\}$
3. $\{(\neg x), (\neg x \vee \neg y)\}$

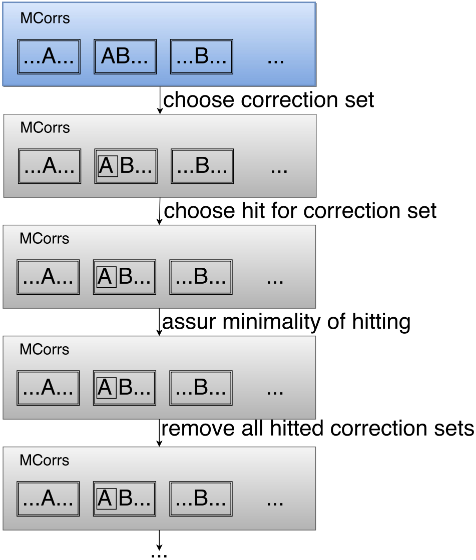
FlowChart


Propositional logic formula in CNF form, ϕ
Compute minimal correction subsets (MCorrs)
Compute hitting sets of MCorrs
Minimal unsatisfiable subsets (MUnsats)

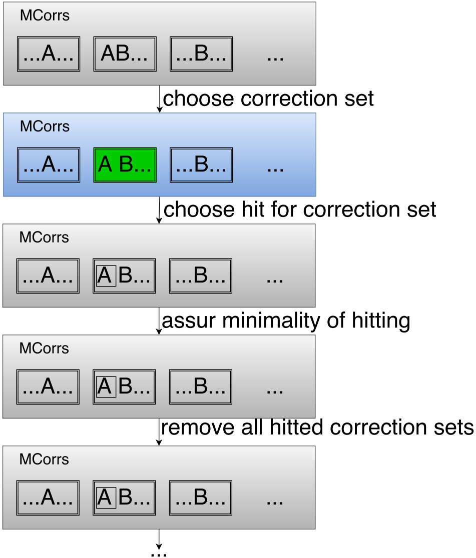
Example: Computing All Minimal Hitting Sets of MUnsats



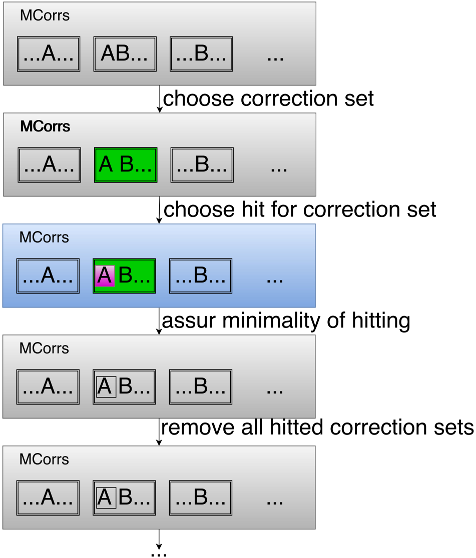
Example: Computing All Minimal Hitting Sets of MUnsats



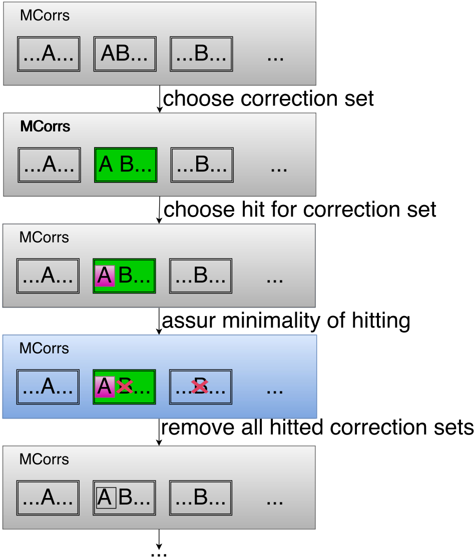
Example: Computing All Minimal Hitting Sets of MUnsats



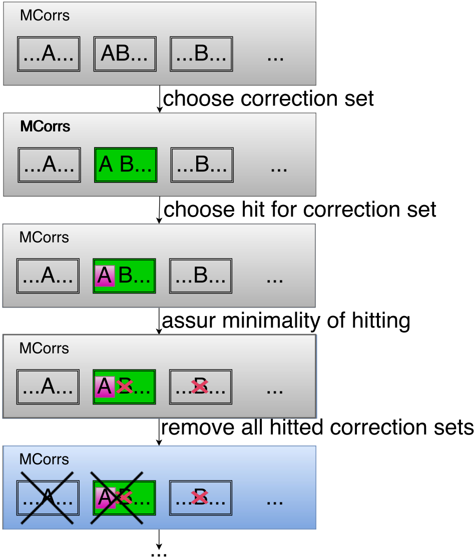
Example: Computing All Minimal Hitting Sets of MUnsats



Example: Computing All Minimal Hitting Sets of MUnsats



Example: Computing All Minimal Hitting Sets of MUnsats



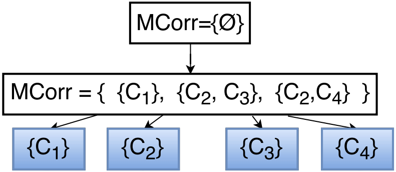
Example: Computing All Hitting Sets of MUnsats

$$\mathrm{MCorr}=\{\emptyset\}$$

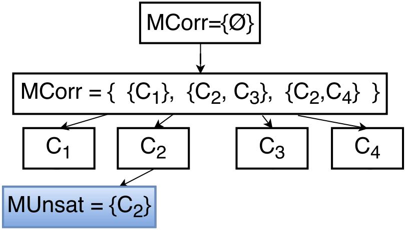
$$\downarrow$$

$$\mathrm{MCorr} = \{\ \{C_1\},\ \{C_2, C_3\},\ \{C_2, C_4\}\ \}$$

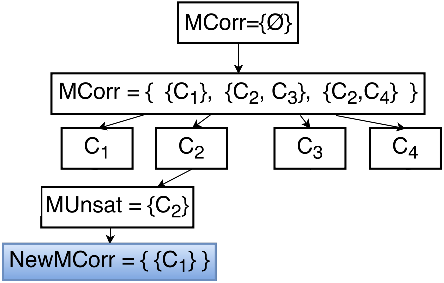
Example: Computing All Hitting Sets of MUnsats



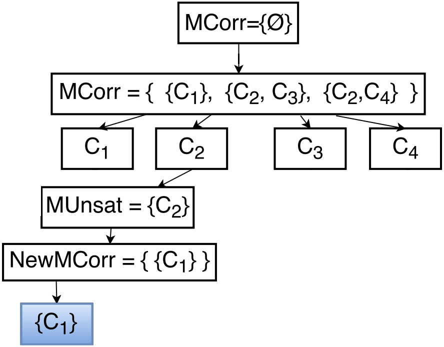
Example: Computing All Hitting Sets of MUnsats



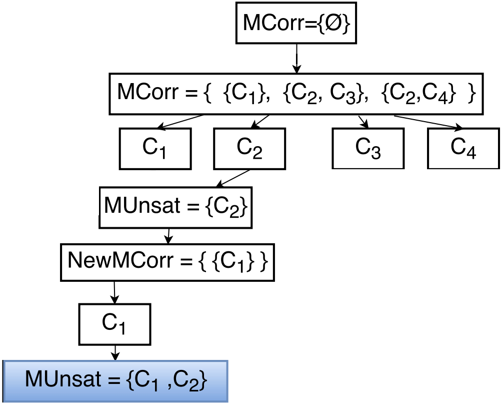
Example: Computing All Hitting Sets of MUnsats



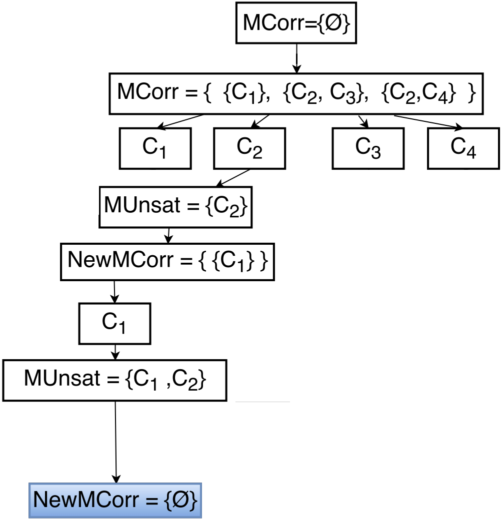
Example: Computing All Hitting Sets of MUnsats



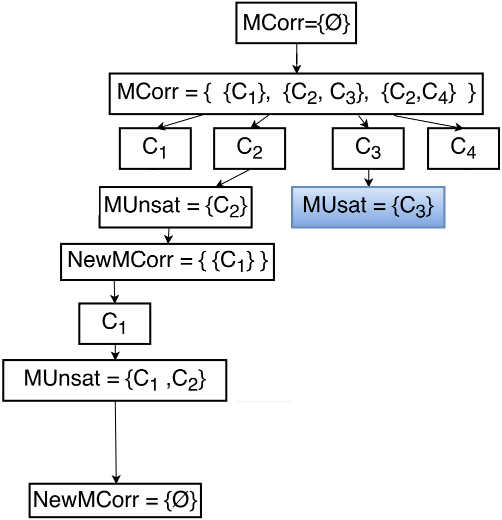
Example: Computing All Hitting Sets of MUnsats



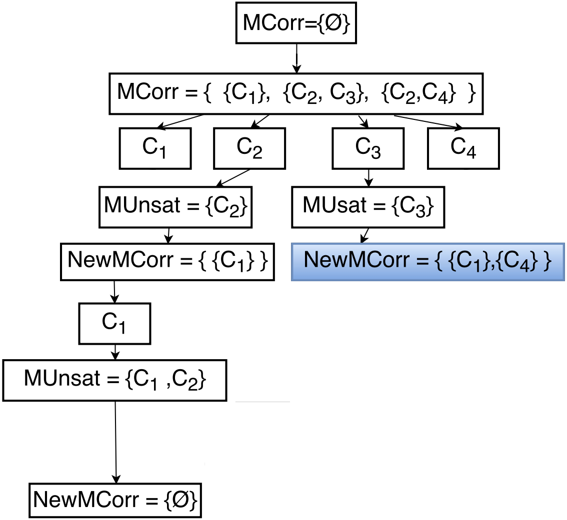
Example: Computing All Hitting Sets of MUnsats



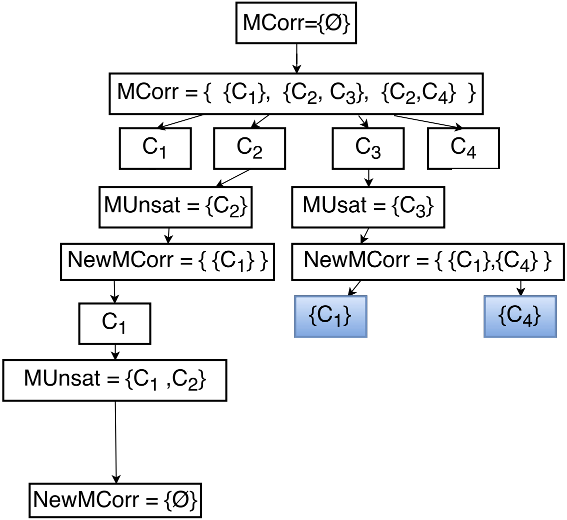
Example: Computing All Hitting Sets of MUnsats



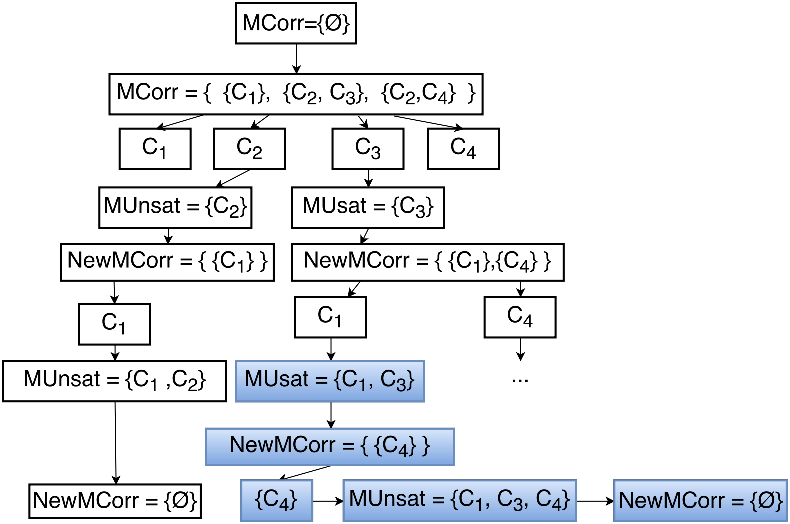
Example: Computing All Hitting Sets of MUnsats



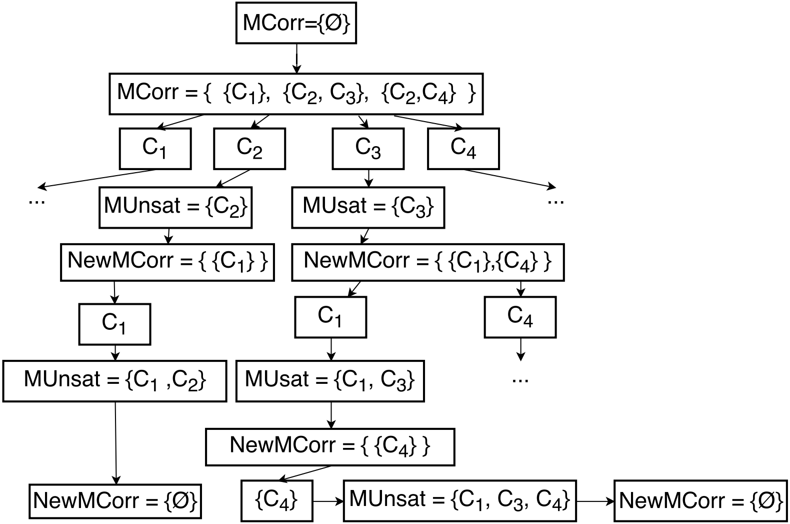
Example: Computing All Hitting Sets of MUnsats



Example: Computing All Hitting Sets of MUnsats



Example: Computing All Hitting Sets of MUnsats



Outlook

- Perform better than the existing algorithms
- Application:
  - A design debugging / fault diagnosis system
  - Reveal system

Reference

Mark H. Liffiton, Karem A. Sakallah, *Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints*. Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor 48109-2121.

Alexander Nadel, Vadim Ryvchin, Ofer Strichman *Ultimately Incremental SAT*. https://ie.technion.ac.il/~ofers/publications/sat14.pdf