Computing Minimal Unsatisfiable Subsets of Constraints

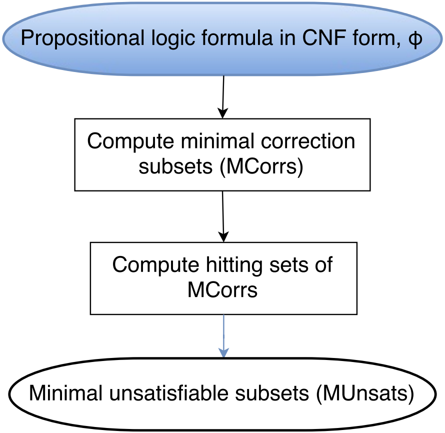
Author: Shahriar Robbani
Supervision: Erika Ábrahám

Theory of Hybrid Systems - Informatik 2 - RWTH-Aachen

Satisfiability Checking Seminar, Winter-16/17

Outline

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 $\neg x$

literals

Propositional Logic Formula

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

literals

Propositional Logic Formula

$$\varphi := \underbrace{\overbrace{(x \vee \neg x)}^{\textit{clause}} \wedge (x \vee \neg y)}_{\text{Conjunctive Normal form}}$$

literals

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

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

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

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

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 Subsets and Minimal Correction Subset

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 Subset

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

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

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

  Collection Set: $\Omega = \{(w,x),(x,c,z)\}$

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

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

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

  Collection Set: $\Omega = \{(w,x),(x,c,z)\}$

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

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

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

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

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

  Collection Set: $\Omega = \{(w,x),(x,c,z)\}$

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

  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: $\Omega = \{(w,x),(x,c,z)\}$

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

  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: $\Omega = \{(w,x),(x,c,z)\}$
  Hitting Set: $H = \{(w,x), (x,y), x, \ldots\}$
  Minimal Hitting Set: $MinH = \{(w,y), (w,z), x\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

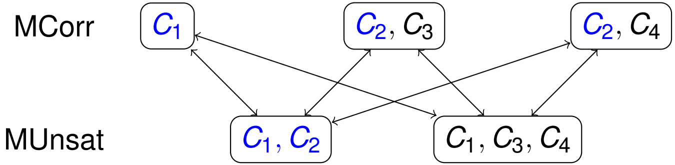
  Variable Set: $D = \{w, x, y, z\}$
  Collection Set: $\Omega = \{(w,x),(x,c,z)\}$
  Hitting Set: $H = \{(w,x), (x,y), x, \ldots\}$
  Minimal Hitting Set: $MinH = \{(w,y), (w,z), x\}$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

  Variable Set: D = {w, x, y, z}
  Collection Set: Ω = {(w,x),(x,c,z)}
  Hitting Set: H = {(w,x), (x,y), x, . . .}
  Minimal Hitting Set: MinH = {(w,y), (w,z), x}

- Minimal hitting sets of the set of MCorr:



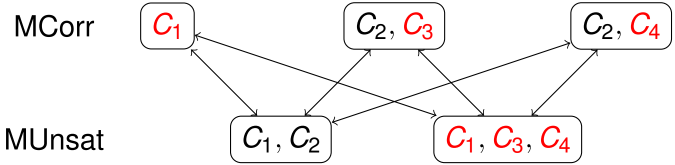
$$\varphi = \underbrace{(x)}_{C_1} \wedge \underbrace{(\neg x)}_{C_2} \wedge \underbrace{(\neg x \vee y)}_{C_3} \wedge \underbrace{(\neg x \vee \neg y)}_{C_4}$$

Duality of Minimal Unsatisfiable and Correction Subset

- **Hitting Sets:**

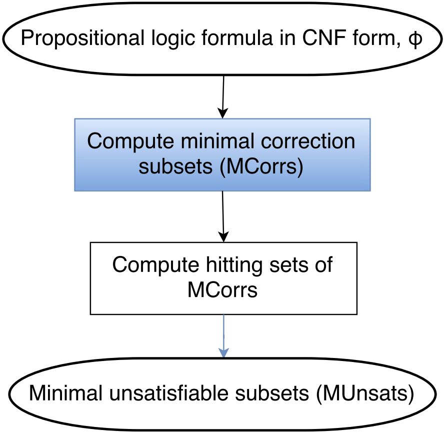
  Variable Set: D = {w, x, y, z}
  Collection Set: Ω = {(w,x),(x,c,z)}
  Hitting Set: H = {(w,x), (x,y), x, . . .}
  Minimal Hitting Set: MinH = {(w,y), (w,z), x}

- 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

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

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 **false**

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

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 **false**

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

3. Add blocking clauses to block old solutions

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

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 **false**

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

3. Add blocking clauses to block old solutions

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

4. Find MCorrs incrementally until all are found.

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 MCS.

MCSs

1. $\{x\}$

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

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 MCS.
4. No further solutions, increment AtMost.

MCSs

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 MCS.
4. No further solutions, increment AtMost.
5. Second solution : w2 and w3 are **false**. Add blocking clause and another MCSs.

MCSs

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 MCS.
4. No further solutions, increment AtMost.
5. Second solution : w2 and w3 are **false**. Add blocking clause and another MCSs.
6. Third solution : w2 and w4 are **false**. Add blocking clause and another MCSs.

MCSs

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

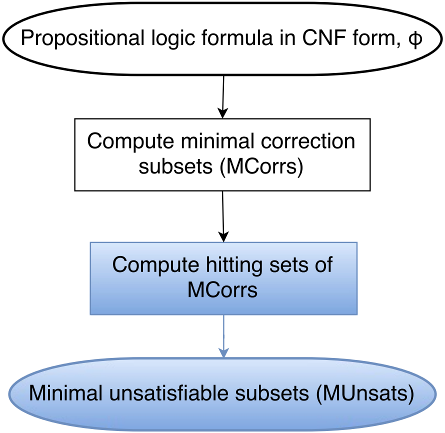
$$\varphi = (\neg w_1 \vee x) \wedge (\neg w_2 \vee \neg x) \wedge (\neg w_3 \vee \neg x \vee y) \wedge (\neg w_4 \vee \neg x \vee \neg y) \wedge (w_1) \wedge (w_2 \vee w_3) \wedge (w_2 \vee w_4)$$

1. Add clause-selector variables.
2. Add AtMost constraint.
3. First solution : w_1 is **false**. Add blocking clause and a MCS.
4. No further solutions, increment AtMost.
5. Second solution : w2 and w3 are **false**. Add blocking clause and another MCSs.
6. Third solution : w2 and w4 are **false**. Add blocking clause and another MCSs.
7. No further solutions, even without AtMost constraint.

MCSs

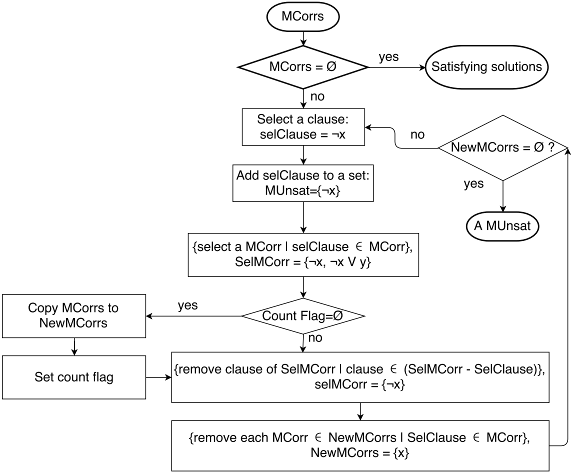
1. $\{x\}$
2. $\{\neg x, \neg x \vee y\}$
3. $\{\neg x, \neg x \vee \neg y\}$

FlowChart


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

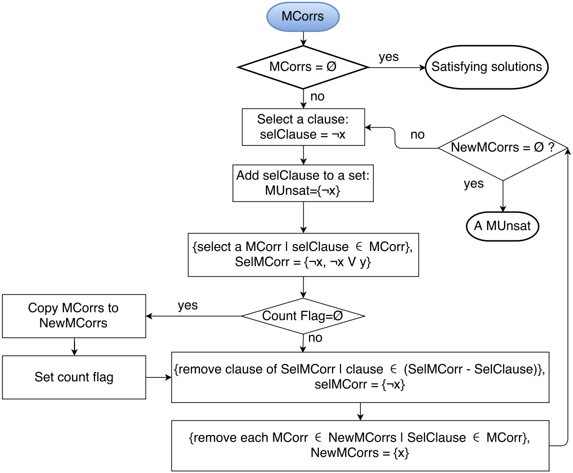
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \lor y), (\neg x, \neg x \lor \neg y)\}$$



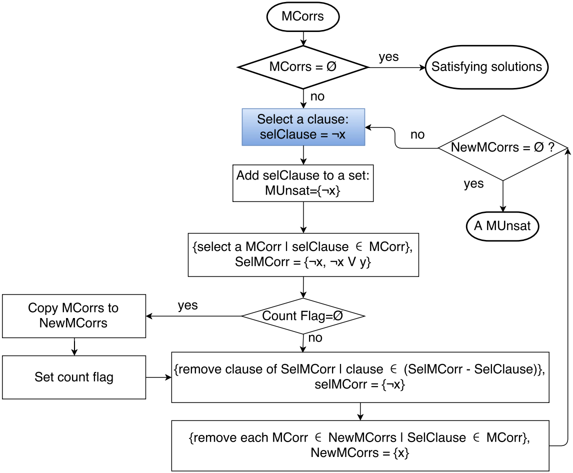
Computing a MUnsat: Search Tree

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



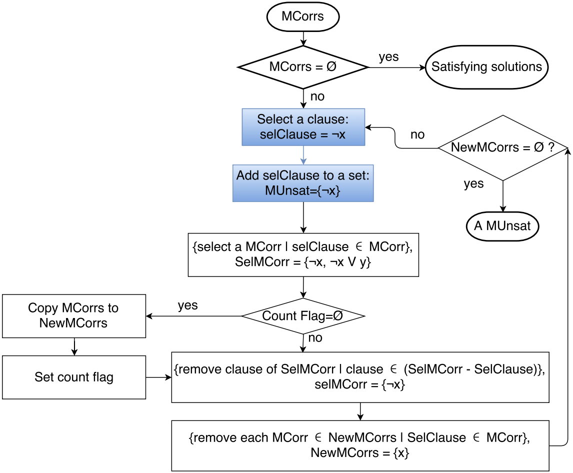
Computing a MUnsat: Search Tree

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



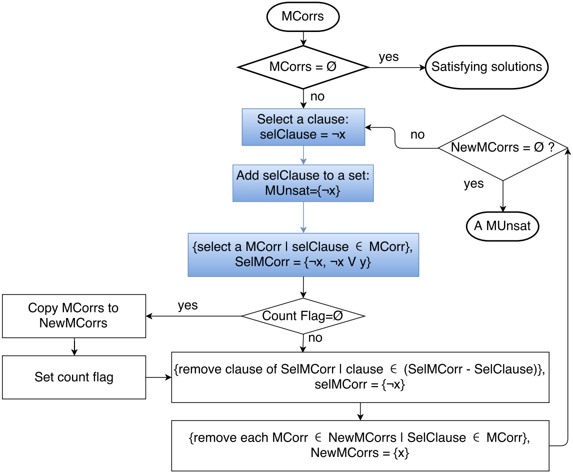
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$MUnsat = \{\neg x\}$$



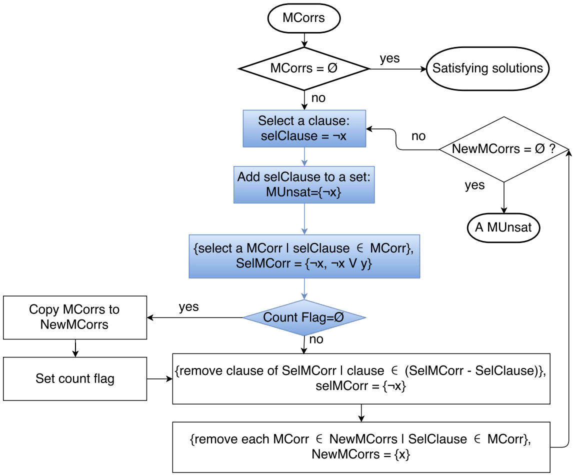
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$SelMCorr = \{\neg x, \neg x \vee y\}$$



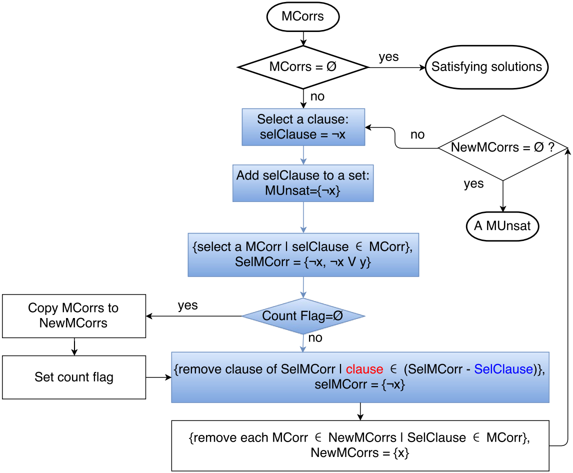
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$SelMCorr = \{\neg x, \neg x \vee y\}$$



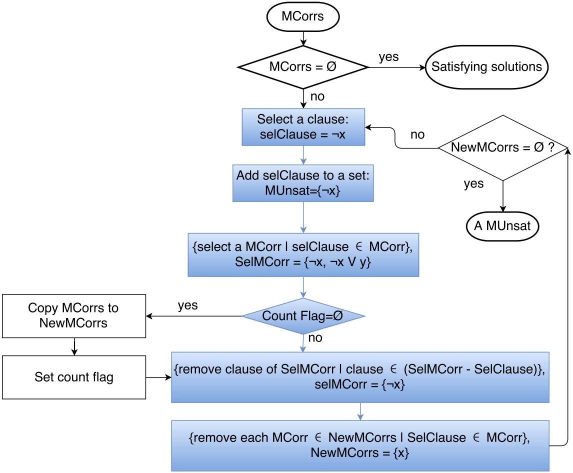
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$SelMCorr = \{\neg x, \neg x \vee y\}$$



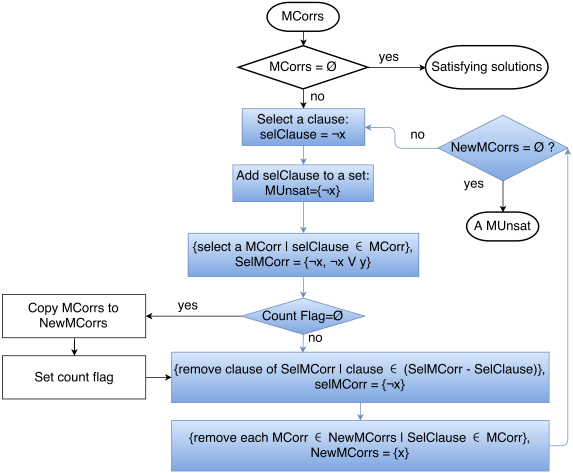
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$NewMCorrs = \{x\}$$



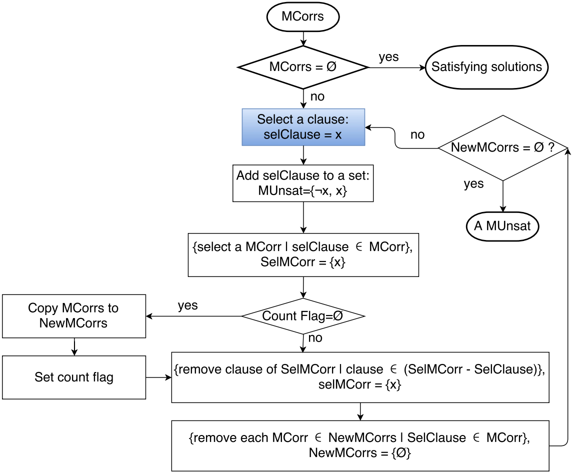
Computing a MUnsat: Search Tree

$$MCorrs = \{x, (\neg x, \neg x \vee y), (\neg x, \neg x \vee \neg y)\}$$
$$NewMCorrs = \{x\}$$



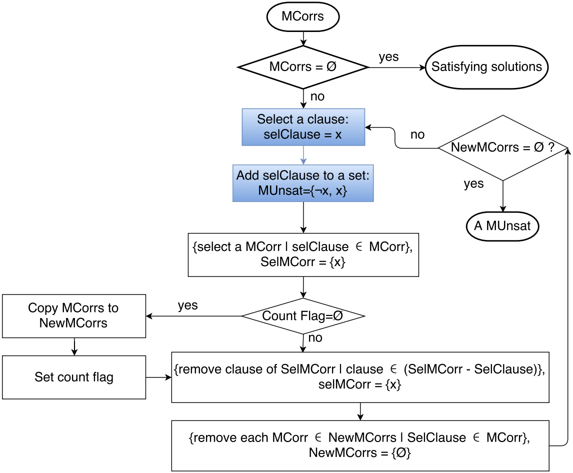
Computing a MUnsat: Search Tree

$$NewMCorrs = \{x\}$$



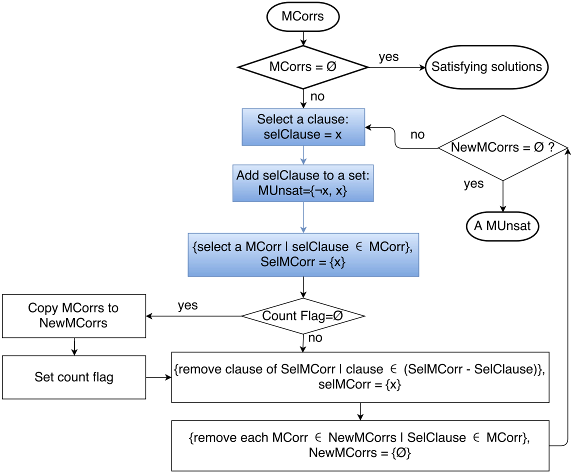
Computing a MUnsat: Search Tree

$$NewMCorrs = \{x\}$$
$$MUnsat = \{\neg x, x\}$$



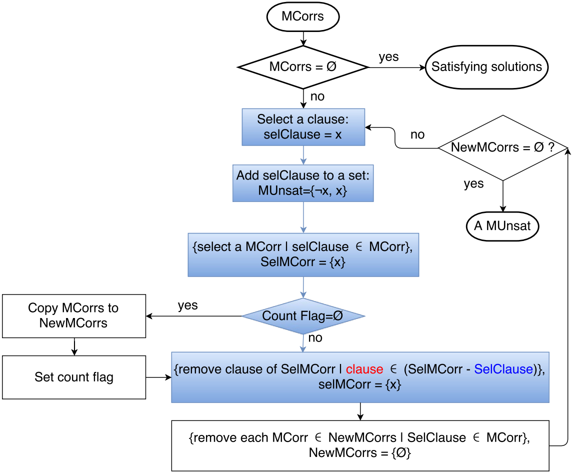
Computing a MUnsat: Search Tree

$$NewMCorrs = \{x\}$$
$$SelMCorr = \{x\}$$



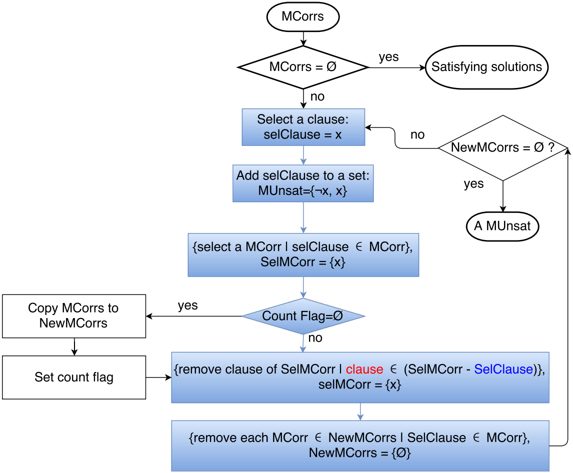
Computing a MUnsat: Search Tree

$$NewMCorrs = \{x\}$$
$$SelMCorr = \{x\}$$



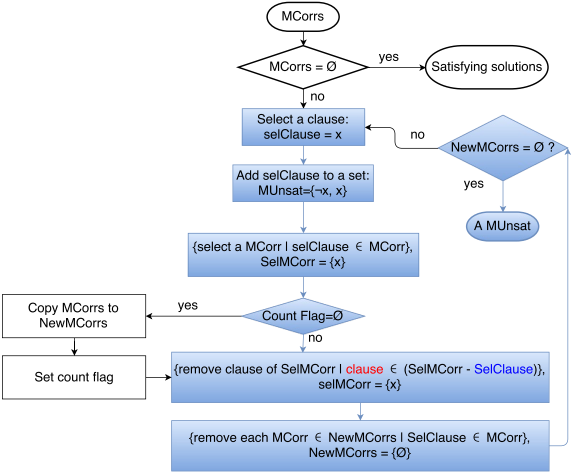
Computing a MUnsat: Search Tree

$$NewMCorrs = \{\emptyset\}$$

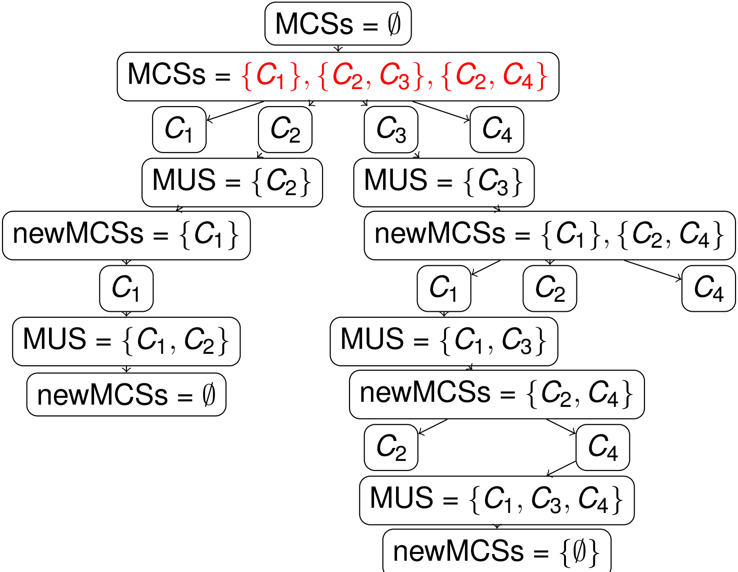


Computing a MUnsat: Search Tree

$$NewMCorrs = \{\emptyset\}$$
$$MUnsat = \{\neg x, x\}$$



Example: Computing All Hitting Sets of MUnsats



Example: Computing All Hitting Sets of MUnsats