Algorithms for Computational Logic

Minimal Unsatisfiable Subformulas

Emmanuel Hebrard (adapted from) João Marques Silva

Outline

Outline

Notation

- $\mathcal{A} \vDash \mathcal{B}$ iff any model of $\mathcal{A}$ is a model of $\mathcal{B}$
  - $(x) \vDash (x \vee y)$
  - $(x \vee y) \wedge (\neg x \vee z) \vDash (y \vee z)$
- $\mathcal{F} \vDash \perp$ iff $\mathcal{F}$ is unsatisfiable
  - $\mathcal{F}$ is said to be inconsistent
- $\mathcal{F} \nvDash \perp$ iff $\mathcal{F}$ is satisfiable

Analyzing inconsistent problems – explanations

- Let $\mathcal{F}$ be inconsistent, i.e. $\mathcal{F} \vDash \perp$
- $\mathcal{G} \subseteq \mathcal{F}$ is an explanation of $\mathcal{F}$ being inconsistent if $\mathcal{G} \vDash \perp$
- Often we are interested in minimal or minimum explanations
- Example 1:

$$\mathcal{F} \triangleq (x_1) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2) \land (x_3 \lor \neg x_4) \land (\neg x_3 \lor x_4)$$

  - An explanation for $\mathcal{F}$ being inconsistent is:

$$\mathcal{G} \triangleq (x_1) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2)$$

- Example 2:

$$\mathcal{F} \triangleq (\neg x_1) \land (\neg x_2) \land (x_1 \lor x_2) \land (\neg x_2 \lor x_3) \land (\neg x_3 \lor x_2)$$

  - An explanation for $\mathcal{F}$ being inconsistent is:

$$\mathcal{G} \triangleq (\neg x_1) \land (\neg x_2) \land (x_1 \lor x_2)$$

Analyzing inconsistent problems – relaxations

- Let $\mathcal{F}$ be inconsistent, i.e. $\mathcal{F} \vDash \perp$
- $\mathcal{G} \subseteq \mathcal{F}$ is a relaxation of $\mathcal{F}$ if $\mathcal{G} \nvDash \perp$
- Often we are interested in maximal or maximum relaxations
- Example:

$$\mathcal{F} \triangleq (x_1) \land (\neg x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2) \land (x_3 \lor \neg x_4) \land (\neg x_3 \lor x_4)$$

  - A relaxation of $\mathcal{F}$ is:

$$\mathcal{G} \triangleq (x_1) \land (\neg x_1 \lor x_2) \land (x_3 \lor \neg x_4) \land (\neg x_3 \lor x_4)$$

Outline

MUSes, MCSes & MSSes

- Given $\mathcal{F}$ unsatisfiable, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M}$ is unsatisfiable and $\forall_{c \in \mathcal{M}}, \mathcal{M} \setminus \{c\}$ is satisfiable

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- $\mathcal{C} \subseteq \mathcal{F}$ is a Minimal Correction Subset (MCS) iff $\mathcal{F} \setminus \mathcal{C}$ is satisfiable and $\forall_{c \in \mathcal{C}}, \mathcal{F} \setminus (\mathcal{C} \setminus \{c\})$ is unsatisfiable

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- $\mathcal{S} \subseteq \mathcal{F}$ is a Maximal Satisfiable Subset (MSS) iff $\mathcal{S}$ is satisfiable and $\forall_{c \in \mathcal{F} \setminus \mathcal{S}}, \mathcal{S} \cup \{c\}$ is unsatisfiable
  - $\mathcal{S} \subseteq \mathcal{F}$ is an MSS iff $\mathcal{C} = \mathcal{F} \setminus \mathcal{S}$ is an MCS

Example – MCSes as minimal sets

$(\bar{x}_1 \vee \bar{x}_2)$ (x_1) $(x_5 \vee x_6)$ $(\bar{x}_3 \vee \bar{x}_4)$ (x_2) (x_3) (x_4)

- Formula is unsatisfiable with satisfiable subformulas
- Can remove clauses such that remaining clauses are satisfiable

- **Minimal Correction Subset** (**MCS**):
  - Irreducible subformula such that the complement is satisfiable
    - MCSes are minimal sets

Another example – find MCS

$$\begin{array}{cc} (x_1) & (\neg x_1) \\ \dots & \dots \\ (x_n) & (\neg x_n) \end{array}$$

- E.g. for each $i = 1, \dots, n$, pick (x_i)

How many MCSes?

$$\begin{array}{cc} (x_1) & (\neg x_1) \\ \dots & \dots \\ (x_n) & (\neg x_n) \end{array}$$

- For each $i = 1, \dots, n$ either pick (x_i) or $(\neg x_i)$, i.e. 2 cases
- Thus, 2^n MCSes

Another example – find MCS

$$\begin{array}{cc} (\neg x_1) \wedge (x_1 \vee z_1) & \\ (\neg y_1) \wedge (y_1 \vee z_1) & \\ \dots & (\neg z_1 \vee \neg z_2 \vee \dots \vee \neg z_n) \\ (\neg x_n) \wedge (x_n \vee z_n) & \\ (\neg y_n) \wedge (y_n \vee z_n) & \end{array}$$

- E.g. remove $(\neg z_1 \vee \neg z_2 \vee \dots \vee \neg z_n)$

How many MUSes?

$$(\neg x_1) \wedge (x_1 \vee z_1)$$
$$(\neg y_1) \wedge (y_1 \vee z_1)$$
$$\dots \qquad\qquad (\neg z_1 \vee \neg z_2 \vee \dots \vee \neg z_n)$$
$$(\neg x_n) \wedge (x_n \vee z_n)$$
$$(\neg y_n) \wedge (y_n \vee z_n)$$

- For each $i = 1, \dots, n$ either resolve away x_i or y_i, i.e. 2 cases
- Thus, 2^n MUSes
- But, there exist formulas with more MUSes. **How?**

Increasing the number of MUSes

$$(\neg x_1) \wedge (\neg x_2) \wedge \dots \wedge (\neg x_r)$$
$$(x_1 \vee z_1) \wedge (x_2 \vee z_1) \wedge \dots \wedge (x_r \vee z_1)$$
$$(x_1 \vee z_2) \wedge (x_2 \vee z_2) \wedge \dots \wedge (x_r \vee z_2)$$
$$\dots$$
$$(x_1 \vee z_n) \wedge (x_2 \vee z_n) \wedge \dots \wedge (x_r \vee z_n)$$
$$(\neg z_1 \vee \neg z_2 \vee \dots \vee \neg z_n)$$

- There are r^n MUSes
- Upper bound by Sperner's theorem: $C(m, \lfloor \frac{m}{2} \rfloor)$

Outline

Overview of query complexity results

- Set $\mathcal{R}$ with m elements and k the size of largest minimal subset
- Worst-case number of oracle calls:
  - Insertion-based: $\mathcal{O}(k\, m)$ [Siqueira&Puget'88; van Maaren&Wieringa'08]
  - Deletion-based: $\mathcal{O}(m)$ [Chinneck&Dravnieks'91; Bakker et al.'93]
  - Linear insertion: $\mathcal{O}(m)$ [MSL'11,BLMS'12]
  - Dichotomic: $\mathcal{O}(k \log(m))$ [Lecoutre et al.'06]
  - QuickXplain: $\mathcal{O}(k + k \log(\frac{m}{k}))$ [Junker'01; Junker'04]
  - Progression: $\mathcal{O}(k \log(1 + \frac{m}{k}))$ [MSJB'13]
- For MUS extraction:
  - Oracle calls correspond to testing unsatisfiability with SAT solver
    - ★ $p(\cdot) \triangleq \text{UNSAT}(\cdot)$

Input : Set $\mathcal{R}$
Output: Minimal subset $\mathcal{M}$
begin
  $\mathcal{M} \leftarrow \mathcal{R}$
  foreach $c \in \mathcal{M}$ **do**
    if $UNSAT(\mathcal{M} \setminus \{c\})$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \setminus \{c\}$ // If $UNSAT(\mathcal{M} \setminus \{c\})$, then $c \notin$ MUS Remove c from $\mathcal{M}$
  return $\mathcal{M}$ // Final $\mathcal{M}$ is minimal set
end

- Number of predicate tests: $\mathcal{O}(m)$ [Chinneck&Dravnieks'91; Bakker et al.'93]

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	UNSAT$(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_8$	$c_2..c_8$	1	Drop c_1
$c_2..c_8$	$c_3..c_8$	1	Drop c_2
$c_3..c_8$	$c_4..c_8$	1	Drop c_3
$c_4..c_8$	$c_5..c_8$	0	Keep c_4
$c_4..c_8$	$c_4 c_6..c_8$	0	Keep c_5
$c_4..c_8$	$c_4 c_5 c_7 c_8$	1	Drop c_6
$c_4 c_5 c_7 c_8$	$c_4 c_5 c_8$	1	Drop c_7
$c_4 c_5 c_8$	$c_4 c_5$	0	Keep c_8

- MUS: $\{c_4, c_5, c_8\}$

Insertion-based MUS computation

Input : Set $\mathcal{R}$
Output: Minimal subset $\mathcal{M}$
begin

```
    M ← ∅
    while R ≠ ∅ do
        S ← ∅                                   // Subset of R
        c_r ← ∅
        while ¬UNSAT(M ∪ S) do
            c_i ← SelectRemoveElement(R)
            S ← S ∪ {c_i}
            c_r ← c_i
        M ← M ∪ {c_r}      // c_r is in minimal subset c_r is transition element
        R ← S \ {c_r}
    return M                                    // Final M is minimal subset
end
```

Number of predicate tests: $\mathcal{O}(k\,m)$

[Siqueira&Puget'88; van Maaren&Wieringa'08]

Insertion – MUS example

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{M}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{M} \cup \mathcal{S}$	UNSAT($\mathcal{M} \cup \mathcal{S}$)	Outcome
$\emptyset$	$c_1..c_8$	$\emptyset$	$\emptyset$	0	–
$\emptyset$	$c_2..c_8$	c_1	c_1	0	–
$\emptyset$	$c_3..c_8$	$c_1 c_2$	$c_1 c_2$	0	–
$\emptyset$	$c_4..c_8$	$c_1..c_3$	$c_1..c_3$	0	–
$\emptyset$	$c_5..c_8$	$c_1..c_4$	$c_1..c_4$	0	–
$\emptyset$	$c_6..c_8$	$c_1..c_5$	$c_1..c_5$	0	–
$\emptyset$	$c_7 c_8$	$c_1..c_6$	$c_1..c_6$	0	–
$\emptyset$	c_8	$c_1..c_7$	$c_1..c_7$	0	–
$\emptyset$	$\emptyset$	$c_1..c_8$	$c_1..c_8$	1	Add c_8 to $\mathcal{M}$
c_8	$c_1..c_7$	$\emptyset$	c_8	0	–
c_8	$c_2..c_7$	c_1	$c_1 c_8$	0	–
...					

Insertion – MUS example (Cont'd)

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{M}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{M} \cup \mathcal{S}$	UNSAT$(\mathcal{M} \cup \mathcal{S})$	Outcome
c_8	$c_3..c_7$	$c_1 c_2$	$c_1 c_2 c_8$	0	–
c_8	$c_4..c_7$	$c_1..c_3$	$c_1..c_3 c_8$	0	–
c_8	$c_5..c_7$	$c_1..c_4$	$c_1..c_4 c_8$	0	–
c_8	$c_6..c_7$	$c_1..c_5$	$c_1..c_5 c_8$	1	Add c_5 to $\mathcal{M}$
$c_5 c_8$	$c_1..c_4$	$\emptyset$	$c_5 c_8$	0	–
$c_5 c_8$	$c_2..c_4$	c_1	$c_1 c_5 c_8$	0	–
$c_5 c_8$	$c_3..c_4$	$c_1 c_2$	$c_1 c_2 c_5 c_8$	0	–
$c_5 c_8$	c_4	$c_1..c_3$	$c_1..c_3 c_5 c_8$	0	–
$c_5 c_8$	$\emptyset$	$c_1..c_4$	$c_1..c_5 c_8$	1	Add c_4 to $\mathcal{M}$
...					

- MUS: $\{c_4, c_5, c_8\}$

Dichotomic

Input : Set $\mathcal{R} = \{c_1, \ldots, c_m\}$
Output: Minimal subset $\mathcal{M}$
begin
    $\mathcal{M} \leftarrow \emptyset$
    while $\neg UNSAT(\mathcal{M})$ **do**
        min $\leftarrow 1$
        max $\leftarrow |\mathcal{R}|$
        while min $\neq$ max **do**
            mid $= \lfloor(\text{min} + \text{max})/2\rfloor$ // Execute binary search
            $\mathcal{S} \leftarrow \{c_1, \ldots, c_{\text{mid}}\}$ // Extract sub-sequence of $\mathcal{R}$
            if $\neg UNSAT(\mathcal{M} \cup \mathcal{S})$ **then**
                min $\leftarrow$ mid $+ 1$
            else
                max $\leftarrow$ mid
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{c_{\text{min}}\}$ // c_{min} is in minimal subsetc_{min} is transition element
        $\mathcal{R} \leftarrow \{c_1, \ldots, c_{\text{min}-1}\}$
    return $\mathcal{M}$ // Final $\mathcal{M}$ is minimal subset
end

- Number of predicate tests: $\mathcal{O}(k \log m)$

[Lecoutre et al. 06]

Dichotomic – MUS example

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{M}$	UNSAT($\mathcal{M}$)	$\mathcal{R}$	c_{min}
$\emptyset$	0	$c_1..c_8$	c_8
c_8	0	$c_1..c_7$	c_5
$c_5\,c_8$	0	$c_1..c_4$	c_4
$c_4\,c_5\,c_8$	1	$c_1..c_3$	–

- MUS: $\{c_4, c_5, c_8\}$
- Note: additional predicate tests required for computing c_{min}

Outline

QuickXplain

Input: $\mathcal{B}$; $\mathcal{T}$; *has_set*

Output: Elements in the minimal set

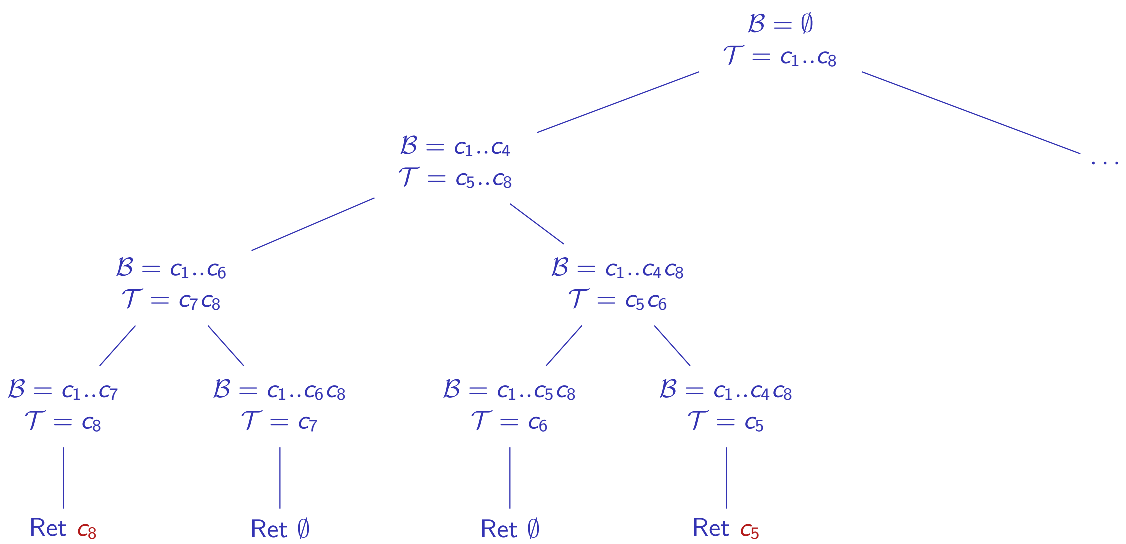
if *has_set* $\wedge\ p(\mathcal{B})$ **then**
  return $\emptyset$
if $|\mathcal{T}| = 1$ **then**
  return $\mathcal{T}$
$m \leftarrow \lfloor \frac{|\mathcal{T}|}{2} \rfloor$
$(\mathcal{T}_1, \mathcal{T}_2) \leftarrow (\mathcal{T}_{1..m}, \mathcal{T}_{m+1..|\mathcal{T}|})$
$\mathcal{M}_2 \leftarrow \text{QuickXplain}(\mathcal{B} \cup \mathcal{T}_1, \mathcal{T}_2, |\mathcal{T}_1| > 0)$
$\mathcal{M}_1 \leftarrow \text{QuickXplain}(\mathcal{B} \cup \mathcal{M}_2, \mathcal{T}_1, |\mathcal{M}_2| > 0)$
return $\mathcal{M}_1 \cup \mathcal{M}_2$

- Number of predicate tests: $\mathcal{O}(k + k\ \log(\frac{m}{k}))$

[Junker'01; Junker'04]

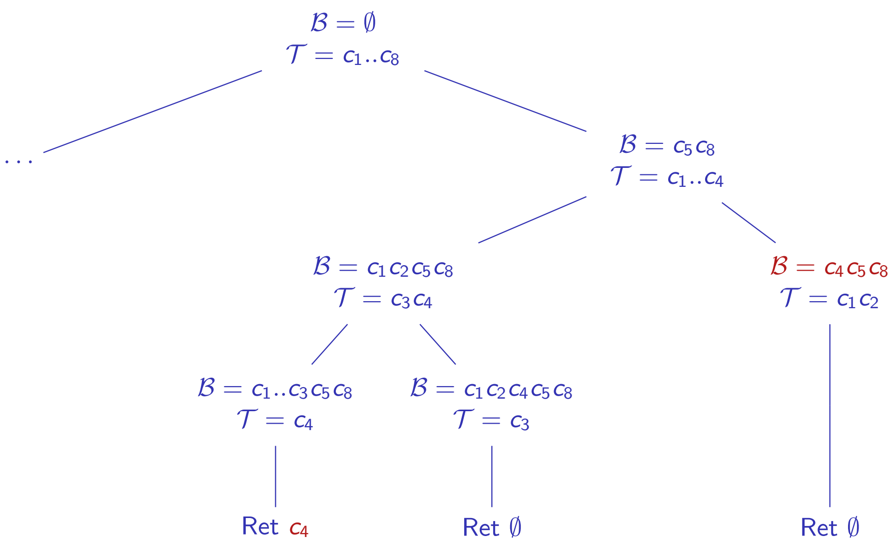
QuickXplain – MUS example

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)



QuickXplain – MUS example (Cont'd)

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)



Outline

1. Inconsistent Problems
2. Defining MCSes & MSSes
3. Basic Algorithms
   - Deletion-Based
   - Insertion-Based
   - Dichotomic
4. Advanced Algorithms
   - QuickXplain
5. **Basic MCS Algorithms**
   - Linear Search
   - Clause D
6. Duality Between MUSes and MCSes

Overview of query complexity results

- Set $\mathcal{R}$ with m elements, k the size of largest minimal subset, and r the size of the smallest minimal subset

- Worst-case number of oracle calls:
  - MaxSAT-based: $\mathcal{O}(\log m)$
  - Linear search: $\mathcal{O}(m)$ [e.g. Bailey&Stuckey'05]
  - Clause D: $\mathcal{O}(m - r)$ [MSHJPB'13]
  - Dichotomic: $\mathcal{O}(k \log(m))$ [Lecoutre et al.'06]
  - FastDiag / QuickXplain: $\mathcal{O}(k + k \log(\frac{m}{k}))$ [Felfernig et al.'12]
  - Progression: $\mathcal{O}(k \log(1 + \frac{m}{k}))$ [MSJB'13]

- For MCS extraction:
  - Oracle calls correspond to testing satisfiability with SAT solver
    - ★ $p(\cdot) \triangleq \mathrm{SAT}(\cdot)$

Basic linear search

- Let $\mathcal{S} \subseteq \mathcal{F}$, such that $\mathcal{S} \nvDash \bot$, initially $\mathcal{S} = \emptyset$

- Let $\mathcal{C} \subseteq \mathcal{F}$, such that $\forall_{c \in \mathcal{C}} \mathcal{S} \cup \{c\} \vDash \bot$, initially $\mathcal{C} = \emptyset$

- At each iteration, analyze one clause $c \in \mathcal{F} \setminus (\mathcal{S} \cup \mathcal{C})$:
  - If $\mathcal{S} \cup \{c\} \vDash \bot$, then add c to $\mathcal{C}$, i.e. c is part of MCS
  - If $\mathcal{S} \cup \{c\} \nvDash \bot$, then add c to $\mathcal{S}$, i.e. c is part of MSS

- Number of calls to oracle: $\mathcal{O}(m)$

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{C}$	$\mathcal{S}$	c	$\mathcal{S} \cup \{c\}$	SAT$(\cdot)$	Outcome
$\emptyset$	$\emptyset$	c_1	c_1	1	Update $\mathcal{S}$
$\emptyset$	c_1	c_2	$c_1 c_2$	1	Update $\mathcal{S}$
$\emptyset$	$c_1 c_2$	c_3	$c_1..c_3$	1	Update $\mathcal{S}$
$\emptyset$	$c_1..c_3$	c_4	$c_1..c_4$	1	Update $\mathcal{S}$
$\emptyset$	$c_1..c_4$	c_5	$c_1..c_5$	1	Update $\mathcal{S}$
$\emptyset$	$c_1..c_5$	c_6	$c_1..c_6$	1	Update $\mathcal{S}$
$\emptyset$	$c_1..c_6$	c_7	$c_1..c_7$	1	Update $\mathcal{S}$
$\emptyset$	$c_1..c_7$	c_8	$c_1..c_8$	0	Update $\mathcal{C}$

- MCS: $\{c_8\}$

- Example 1:

$$(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

- Example 2:

$$(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_4) \wedge (\neg x_4 \vee x_3) \wedge (\neg x_4 \vee \neg x_3)$$

Clause D algorithm

- Pick an assignment and let $\mathcal{S} \subseteq \mathcal{F}$ be the satisfied clauses and $\mathcal{U} \subseteq \mathcal{F}$ be the falsified clauses, with $\mathcal{F} = \mathcal{S} \cup \mathcal{U}$
  - **Claim**: there are **no** complemented literals in $\mathcal{U}$! (**Why?**)
- Repeat:
  - Create clause $D = \cup_{l \in c, c \in \mathcal{U}} l$ (**Obs:** D is not tautologous)
  - If $\mathcal{S} \cup \{D\} \vDash \bot$, then $\mathcal{U}$ is MCS (**Why?**)
    - Report MCS & terminate
  - If $\mathcal{S} \cup \{D\} \nvDash \bot$, then add to $\mathcal{S}$ the satisfied clauses in $\mathcal{U}$, remove from $\mathcal{U}$ the satisfied clauses and loop
- Number of calls to oracle: $\mathcal{O}(m - r)$

Clause D – MCS example

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
$(x_1 \vee x_2)$	$(x_3 \vee x_4)$	$(\neg x_3 \vee \neg x_4)$	$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_5)	$(\neg x_5 \vee x_6)$	(x_2)

$\mathcal{S}$	$\mathcal{U}$	D	SAT$(\cdot)$	Variables $= 1$
$\emptyset$	$\emptyset$	—	1	$\emptyset$
$c_3 c_4 c_7$	$c_1 c_2 c_5 c_6 c_8$	$\{x_1, \ldots, x_5\}$	1	$\{x_1, x_3\}$
$c_1 .. c_5 c_7$	$c_6 c_8$	$\{x_2, x_5\}$	1	$\{x_1, x_3, x_5\}$
$c_1 .. c_7$	c_8	$\{x_2\}$	0	$\emptyset$

- MCS: $\{c_8\}$

Additional examples

- Example 1:

$$(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

- Example 2:

$$(x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_4) \wedge (\neg x_4 \vee x_3) \wedge (\neg x_4 \vee \neg x_3)$$

Outline

Hitting sets

- Let $\mathcal{S}$ be a finite set
- Let $\mathcal{F}$ be a set of subsets of $\mathcal{S}$, $\mathcal{F} \subseteq 2^{\mathcal{S}}$
- A hitting set $\mathcal{H} \subseteq \mathcal{S}$ is such that $\forall_{\mathcal{G} \in \mathcal{F}} \mathcal{H} \cap \mathcal{G} \neq \emptyset$
- $\mathcal{H}$ is subset minimal (or minimal) if none of its subsets is a hitting set of $\mathcal{F}$
- $\mathcal{H}$ is cardinality minimal (or of minimum size) if there are no hitting sets of $\mathcal{F}$ with fewer elements
- An Example

$$
\begin{aligned}
\mathcal{S} &= \{1, 2, 3, 4, 5, 6, 7\} \\
\mathcal{F} &= \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6, 7\}\} \\
\mathcal{H}_1 &= \{1, 2, 4, 6, 7\} \\
\mathcal{H}_2 &= \{2, 4, 6\} \\
\mathcal{H}_3 &= \{3, 7\}
\end{aligned}
$$

MUSes vs. MCSes

- Claim:
  MUSes are minimal hitting sets of MCSes, and
  MCSes are minimal hitting sets of MUSes
- Example:

$$
\begin{array}{ccccccc}
c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \\
\hline
(x_1) & (\neg x_1) & (\neg x_2) & (x_2 \vee x_3) & (x_2 \vee \neg x_3) & (x_2 \vee x_4) & (x_2 \vee \neg x_4)
\end{array}
$$

MUS	$\{\{c_1, c_2\}, \{c_3, c_4, c_5\}, \{c_3, c_6, c_7\}\}$
MCS	$\{\{c_1, c_3\}, \{c_2, c_3\}, \{c_1, c_4, c_6\}, \{c_1, c_4, c_7\}, \{c_1, c_5, c_6\},$ $\{c_1, c_5, c_7\}, \{c_2, c_4, c_6\}, \{c_2, c_4, c_7\}, \{c_2, c_5, c_6\}, \{c_2, c_5, c_7\}\}$

- Any correction set $\mathcal{C}$ must hit all unsatisfiable sets $\mathcal{U}$. **Why?**
  - Otherwise, $\mathcal{C}$ would not hit **any** clause of some unsatisfiable set $\mathcal{V}$, and so $\mathcal{F} \setminus \mathcal{C}$ would be unsatisfiable since it would be contain $\mathcal{V}$; a contradiction
- Any unsatisfiable set $\mathcal{U}$ must hit all correction sets $\mathcal{C}$. **Why?**
  - Otherwise, $\mathcal{U}$ would not hit **any** clause of some correction set $\mathcal{T}$, and so $\mathcal{U}$ would be satisfiable since $\mathcal{F} \setminus \mathcal{U}$ would contain correction set $\mathcal{T}$; a contradiction