Tracking MUSes and Strict Inconsistent Covers

Éric Grégoire Bertrand Mazure Cédric Piette

November 13, 2006

centre de recherche en informatique de lens

Formal Methods in Computer Aided Design (FMCAD'2006)

1. MUSES & INCONSISTENT COVERS
   - Definitions and properties
   - Motivations

2. *(A)OMUS*: A MUS EXTRACTOR
   - Deciding which clauses belong to a MUS
   - Taking the neighborhood of the current interpretation into account
   - Algorithm and Experimental Results

3. COMPUTING ONE STRICT INCONSISTENT COVER
   - Algorithm and Experimental Results

4. CONCLUSIONS AND FUTURE WORK

Definitions and properties

Definition: *CNF formula*

We call:

- *literal*: propositional atom or its negation(l,$\neg l$)
- *clause*: finite disjunction of literals ($l_1 \vee l_2 \vee ... \vee l_n$)
- *CNF formula*: finite conjunction of clauses ($c_1 \wedge c_2 \wedge ... \wedge c_m$)

Definitions and properties

Definition: *CNF formula*

We call:

- *literal*: propositional atom or its negation(l,$\neg l$)
- *clause*: finite disjunction of literals ($l_1 \vee l_2 \vee ... \vee l_n$)
- *CNF formula*: finite conjunction of clauses ($c_1 \wedge c_2 \wedge ... \wedge c_m$)

Definition: *Interpretation*

→ Let ϕ be a CNF formula. An *interpretation* is an application from Var(ϕ) to $\{0, 1\}$.
→ A *model* of ϕ is an interpretation that satisfies ϕ.

DEFINITIONS AND PROPERTIES

DEFINITION: *CNF formula*

We call:

- *literal*: propositional atom or its negation(l,$\neg l$)
- *clause*: finite disjunction of literals ($l_1 \vee l_2 \vee \ldots \vee l_n$)
- *CNF formula*: finite conjunction of clauses ($c_1 \wedge c_2 \wedge \ldots \wedge c_m$)

DEFINITION: *Interpretation*

→ Let ϕ be a CNF formula. An *interpretation* is an application from Var(ϕ) to $\{0, 1\}$.
→ A *model* of ϕ is an interpretation that satisfies ϕ.

DEFINITION: *SAT*

The *SAT* problem consists in deciding whether a CNF formula admits a model, or not. When a model exists, the CNF is said *satisfiable*, otherwise is said *unsatisfiable*.

Definitions and properties

Definition: *CNF formula*

We call:

- ⋆ *literal*: propositional atom or its negation(l,$\neg l$)
- ⋆ *clause*: finite disjunction of literals ($l_1 \vee l_2 \vee ... \vee l_n$)
- ⋆ *CNF formula*: finite conjunction of clauses ($c_1 \wedge c_2 \wedge ... \wedge c_m$)

Definition: *Interpretation*

→ Let ϕ be a CNF formula. An *interpretation* is an application from Var(ϕ) to $\{0, 1\}$.
→ A *model* of ϕ is an interpretation that satisfies ϕ.

Definition: *SAT*

The *SAT* problem consists in deciding whether a CNF formula admits a model, or not. When a model exists, the CNF is said *satisfiable*, otherwise is said *unsatisfiable*.

Property

If a CNF formula is unsatisfiable, then its exhibits at least one Minimal Unsatisfiable Subformula (MUS).

Definitions and properties

Definition: *Minimal Unsatisfiable Subformula (MUS)*

A Minimal Unsatisfiable Subformula or MUS K of a CNF formula ϕ is a set of clauses s.t.

- $K \subseteq \phi$
- K is unsatisfiable
- Each proper subset of K is satisfiable

DEFINITIONS AND PROPERTIES

DEFINITION: *Minimal Unsatisfiable Subformula (MUS)*

A Minimal Unsatisfiable Subformula or MUS K of a CNF formula ϕ is a set of clauses s.t.

- $K \subseteq \phi$
- K is unsatisfiable
- Each proper subset of K is satisfiable

DEFINITION: *The set of MUSes*

The set of MUSes is defined by:

$$KS_{\phi} = \{K \mid K \text{ is a MUS and } K \in \phi\}$$

Definitions and properties

Definition: *Minimal Unsatisfiable Subformula (MUS)*

A Minimal Unsatisfiable Subformula or MUS K of a CNF formula ϕ is a set of clauses s.t.

- $K \subseteq \phi$
- K is unsatisfiable
- Each proper subset of K is satisfiable

Definition: *The set of MUSes*

The set of MUSes is defined by:

$$KS_\phi = \{K \mid K \text{ is a MUS and } K \in \phi\}$$

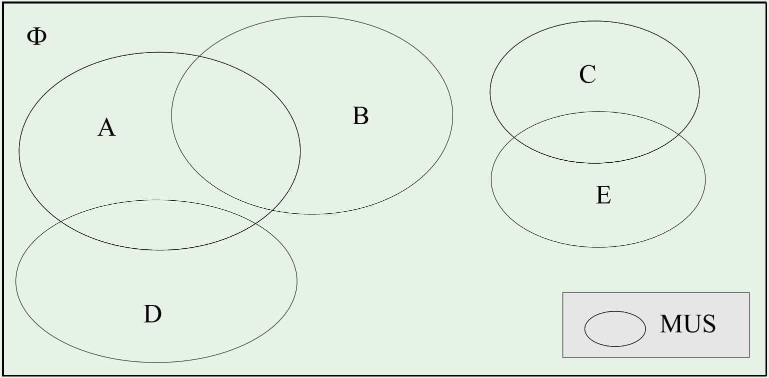
Definition: *Inconsistent cover*

An inconsistent cover of a unsatisfiable CNF formula ϕ is a subset of KS_ϕ such that its removal restores the satisfiability of ϕ.

A *strict* insconsistent cover is composed of independent MUSes.

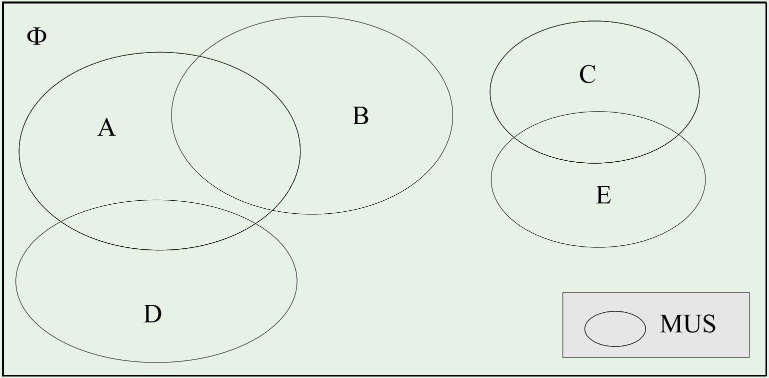
DEFINITIONS AND PROPERTIES

EXAMPLE



DEFINITIONS AND PROPERTIES

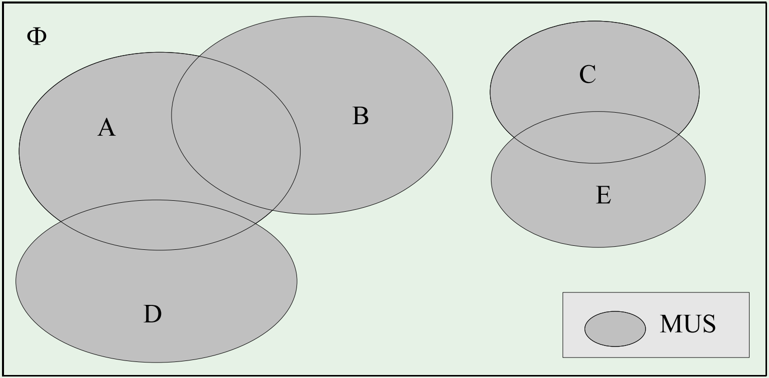
EXAMPLE



- MUSes: A, B, C, D, E

Definitions and properties

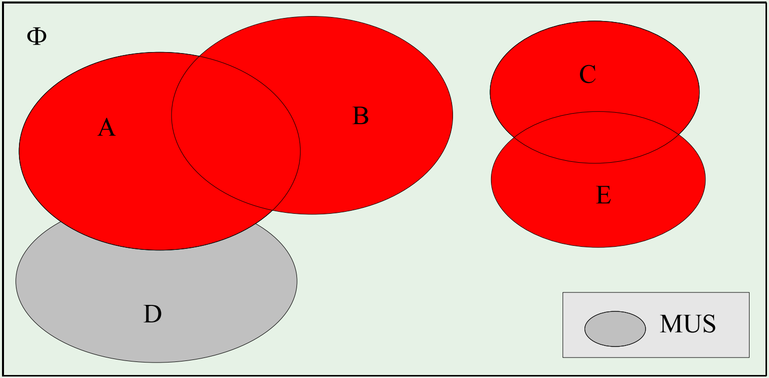
Example



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}

DEFINITIONS AND PROPERTIES

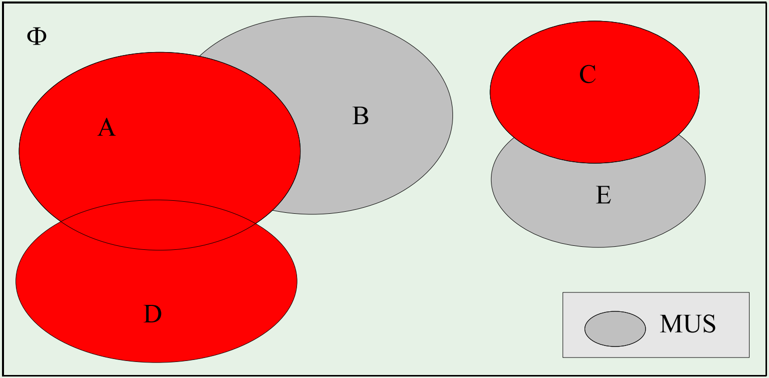
EXAMPLE



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}

DEFINITIONS AND PROPERTIES

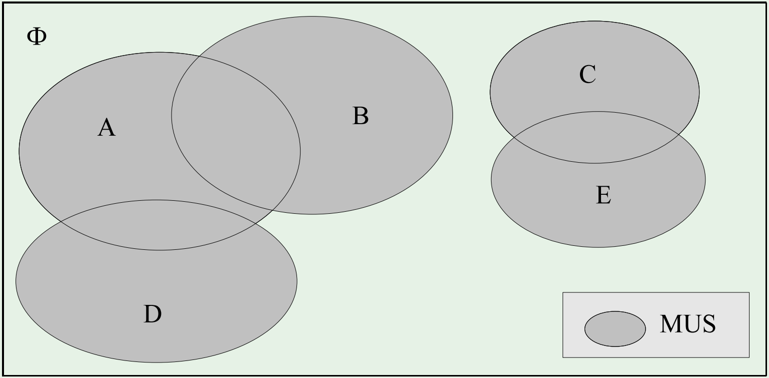
EXAMPLE



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D}

DEFINITIONS AND PROPERTIES

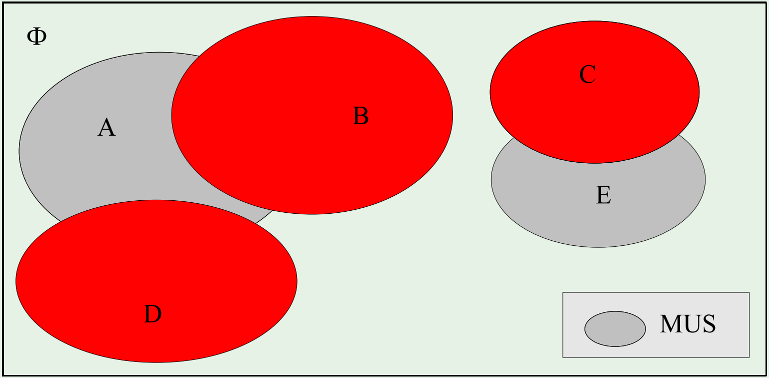
EXAMPLE



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D} , ...
- Strict inconsistent covers:

Definitions and properties

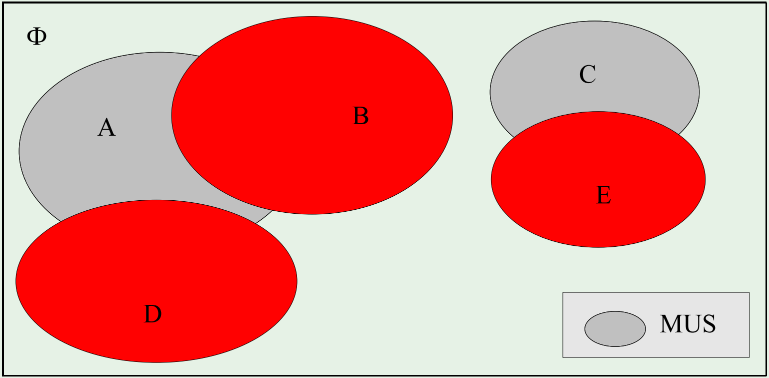
Example



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D} , ...
- Strict inconsistent covers: {B, D, C}

Definitions and properties

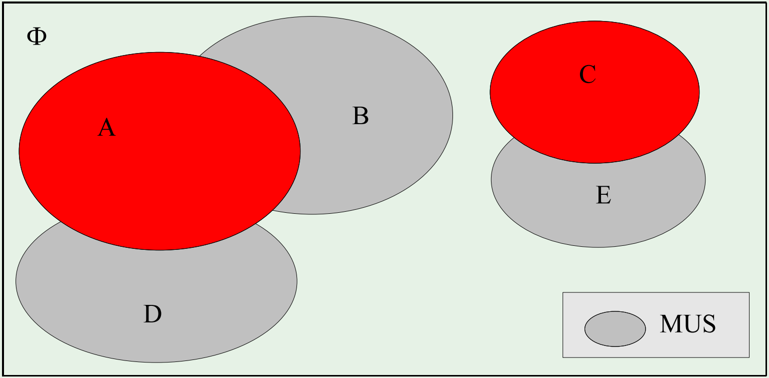
Example



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D} , ...
- Strict inconsistent covers: {B, D, C}, {B, D, E}

Definitions and properties

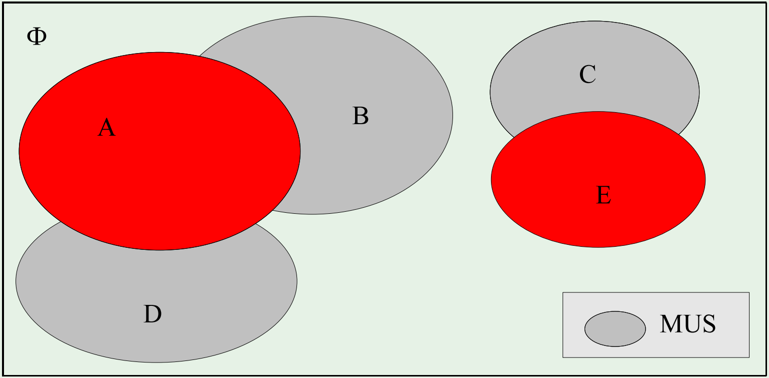
Example



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D} , ...
- Strict inconsistent covers: {B, D, C}, {B, D, E}, {A, C}

Definitions and properties

Example



- MUSes: A, B, C, D, E
- The set of MUSes: {A, B, C, D, E}
- Inconsistent covers: {A, B, C, E}, {A, C, D} , ...
- Strict inconsistent covers: {B, D, C}, {B, D, E}, {A, C}, {A, E}

Definitions and properties

Corollary

Let K be a MUS, and c be a clause. $\forall c \in K$, $K \setminus \{c\}$ is satisfiable.

Definitions and properties

Corollary

Let K be a MUS, and c be a clause. $\forall c \in K$, $K \setminus \{c\}$ is satisfiable.

Property

Let ϕ be an inconsistent n-clauses CNF formula and SIC_ϕ be a strict inconsistent cover of ϕ. Then we have:

$$MaxSat(\phi) \;\leq\; n - |SIC_\phi|$$

DEFINITIONS AND PROPERTIES

COROLLARY

Let K be a MUS, and c be a clause. $\forall c \in K$, $K \backslash \{c\}$ is satisfiable.

PROPERTY

Let ϕ be an inconsistent n-clauses CNF formula and SIC_ϕ be a strict inconsistent cover of ϕ. Then we have:

$$MaxSat(\phi) \leq n - |SIC_\phi|$$

RELATION BETWEEN MAXSAT AND MUSES

Let ω be an optimal interpretation for MaxSat, any falsified clause w.r.t. ω belongs to at least one MUS of the CNF formula.

Motivations

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

Motivations

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

MOTIVATIONS

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

Motivations

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

MOTIVATIONS

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

MOTIVATIONS

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

Motivations

- A MUS represents one smallest explanation for the inconsistency (certificate)
- It can help in finding new technics for SAT practical resolution
- It can provide a way to restore satisfiability
- Lots of potential applications (VLSI correctness checking, non-monotonic logics, etc.)

Complexity

- Deciding whether a CNF formula is a MUS or not is **DP-complete**

  [Papadimitriou & Wolfe 85]

- Deciding whether a CNF formula belongs to the set of MUSes or not is in Σ_2^p

  [Eiter & Gottlob 92]

1. MUSes & Inconsistent Covers
   - Definitions and properties
   - Motivations

2. *(A)OMUS*: a MUS extractor
   - Deciding which clauses belong to a MUS
   - Taking the neighborhood of the current interpretation into account
   - Algorithm and Experimental Results

3. Computing one strict inconsistent cover
   - Algorithm and Experimental Results

4. Conclusions and future work

Deciding the clauses contained in a MUS

Property [Mazure-Sais-Grégoire 97]

Let ϕ be a CNF formula, K a MUS of ϕ, and c a clause.
For all interpretations ω, $\exists c \in K$ s.t. $\omega \not\models c$

DECIDING THE CLAUSES CONTAINED IN A MUS

PROPERTY [MAZURE-SAIS-GRÉGOIRE 97]

Let ϕ be a CNF formula, K a MUS of ϕ, and c a clause.
For all interpretations ω, $\exists c \in K$ s.t. $\omega \not\models c$

CANDIDATE HEURISTIC

During a local search run, the most often falsified clauses belong to MUSes.

Deciding the clauses contained in a MUS

Property [Mazure-Sais-Grégoire 97]

Let ϕ be a CNF formula, K a MUS of ϕ, and c a clause.
For all interpretations ω, $\exists c \in K$ s.t. $\omega \not\models c$

Candidate Heuristic

During a local search run, the most often falsified clauses belong to MUSes.

Problem: Some clauses can be often falsified without belonging to MUSes.

Deciding the clauses contained in a MUS

Property [Mazure-Sais-Grégoire 97]

Let ϕ be a CNF formula, K a MUS of ϕ, and c a clause.
For all interpretations ω, $\exists c \in K$ s.t. $\omega \not\models c$

Candidate Heuristic

During a local search run, the most often falsified clauses belong to MUSes.

Problem: Some clauses can be often falsified without belonging to MUSes.

$\Rightarrow$ A **more discriminating criterion** is needed to identify clauses of MUSes.

Taking the neighborhood of the current interpretation into account

Definition: *once-satisfied clause*

A clause c is said *once-satisfied clause* w.r.t. an interpretation ω iff ω satisfies exactly one literal of c.

Definition: *critical clause*

A clause c falsified w.r.t. an interpretation ω is said critical iff the opposite of each literal of c appears in at least one once-satisfied clause.

These once-satisfied clauses are said linked to the critical clause c

EXAMPLE

$$
\begin{aligned}
&(a \vee b \vee c) \\
\wedge\ &(\neg b \vee e) \\
\wedge\ &(\neg a \vee b \vee c) \\
\wedge\ &(\neg a \vee \neg b) \\
\wedge\ &(a \vee d) \\
\wedge\ &(b \vee \neg c) \\
\wedge\ &(\neg d \vee e) \\
\wedge\ &(a \vee \neg b) \\
\wedge\ &(\neg e \vee \neg f)
\end{aligned}
$$

EXAMPLE

clauses belonging to MUS: $\leftarrow$

$$
\begin{array}{ll}
(a \lor b \lor c) & \leftarrow \\
\land (\neg b \lor e) & \\
\land (\neg a \lor b \lor c) & \leftarrow \\
\land (\neg a \lor \neg b) & \leftarrow \\
\land (a \lor d) & \\
\land (b \lor \neg c) & \leftarrow \\
\land (\neg d \lor e) & \\
\land (a \lor \neg b) & \leftarrow \\
\land (\neg e \lor \neg f) &
\end{array}
$$

EXAMPLE

$$\omega = \{\neg a, \neg b, c, d, e, f\}$$

clauses belonging to MUS: $\leftarrow$

$$
\begin{array}{ll}
(a \lor b \lor c) & \leftarrow \\
\land (\neg b \lor e) & \\
\land (\neg a \lor b \lor c) & \leftarrow \\
\land (\neg a \lor \neg b) & \leftarrow \\
\land (a \lor d) & \\
\land (b \lor \neg c) & \leftarrow \\
\land (\neg d \lor e) & \\
\land (a \lor \neg b) & \leftarrow \\
\land (\neg e \lor \neg f) &
\end{array}
$$

EXAMPLE

$$\omega = \{\neg a, \neg b, c, d, e, f\}$$

PROPERTY

Let c be a critical clause w.r.t. an interpretation ω.

Any flip on ω in order to satisfy c leads to falsify another clause previously satisfied w.r.t. ω.

clauses belonging to MUS: $\leftarrow$

$$
\begin{array}{ll}
(a \vee b \vee c) & \leftarrow \\
\wedge\ (\neg b \vee e) & \\
\wedge\ (\neg a \vee b \vee c) & \leftarrow \\
\wedge\ (\neg a \vee \neg b) & \leftarrow \\
\wedge\ (a \vee d) & \\
\wedge\ (b \vee \neg c) & \leftarrow \\
\wedge\ (\neg d \vee e) & \\
\wedge\ (a \vee \neg b) & \leftarrow \\
\wedge\ (\neg e \vee \neg f) &
\end{array}
$$

EXAMPLE

$$\omega = \{\neg a, \neg b, c, d, e, f\}$$

PROPOSED HEURISTIC

Performing a local search that counts for each clause the number of times it has been critical.

clauses belonging to MUS: $\leftarrow$

$$
\begin{array}{ll}
\quad (a \vee b \vee c) & \leftarrow \\
\wedge\ (\neg b \vee e) & \\
\wedge\ (\neg a \vee b \vee c) & \leftarrow \\
\wedge\ (\neg a \vee \neg b) & \leftarrow \\
\wedge\ (a \vee d) & \\
\wedge\ (b \vee \neg c) & \leftarrow \\
\wedge\ (\neg d \vee e) & \\
\wedge\ (a \vee \neg b) & \leftarrow \\
\wedge\ (\neg e \vee \neg f) &
\end{array}
$$

WHY COUNTING CRITICAL CLAUSES ?

WHY COUNTING CRITICAL CLAUSES ?

Let K be a MUS, and c be a clause s.t. $c \in K$

WHY COUNTING CRITICAL CLAUSES ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$$\Downarrow$$

$K \setminus \{c\}$ is SAT.

Let ω be a model of $K \setminus \{c\}$

WHY COUNTING CRITICAL CLAUSES ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$$\Downarrow$$

$K \setminus \{c\}$ is SAT.

Let ω be a model of $K \setminus \{c\}$

$$\Downarrow$$

$$\omega \not\models c$$

WHY COUNTING CRITICAL CLAUSES ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$$\Downarrow$$

$K \backslash \{c\}$ is SAT.

Let ω be a model of $K \backslash \{c\}$

$$\Downarrow$$

$$\omega \not\models c$$

$$\Downarrow$$

Let ω' s.t. $d_H(\omega, \omega') = 1$ and $\omega' \models c$

Why counting critical clauses ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$$\Downarrow$$

$K \backslash \{c\}$ is SAT.

Let ω be a model of $K \backslash \{c\}$

$$\Downarrow$$

$$\omega \not\models c$$

$$\Downarrow$$

Let ω' s.t. $d_H(\omega, \omega') = 1$ and $\omega' \models c$

$$\Downarrow$$

$\exists c' \in K$ s.t. $\omega' \not\models c'$

WHY COUNTING CRITICAL CLAUSES ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$$\Downarrow$$

$K \backslash \{c\}$ is SAT.

Let ω be a model of $K \backslash \{c\}$

$$\Downarrow$$

$$\omega \not\models c$$

$$\Downarrow$$

Let ω' s.t. $d_H(\omega, \omega') = 1$ and $\omega' \models c$

$$\Downarrow$$

$\exists c' \in K$ s.t. $\omega' \not\models c'$

$$\Downarrow$$

Then, we have c is critical (w.r.t. ω)

Why counting critical clauses ?

Let K be a MUS, and c be a clause s.t. $c \in K$

$\Downarrow$

$K \backslash \{c\}$ is SAT.

Let ω be a model of $K \backslash \{c\}$

$\Downarrow$

$\omega \not\models c$

$\Downarrow$

Let ω' s.t. $d_H(\omega, \omega') = 1$ and $\omega' \models c$

$\Downarrow$

$\exists c' \in K$ s.t. $\omega' \not\models c'$

$\Downarrow$

Then, we have c is critical (w.r.t. ω)

Property

For each clause c in a MUS, there exists an interpretation ω s.t. c is critical.

Why counting critical clauses ?

Property

For each clause c in a MUS, there exists an interpretation ω s.t. c is critical.

Extension of the relationship between MaxSat and MUSes

Let ω be an optimal interpretation for MaxSat, any falsified clause c w.r.t. ω:

- belongs to at least one MUS of the CNF formula
- is critical w.r.t. ω
- at least one once-satified clause linked to c belongs to the same MUS

(A)OMUS ALGORITHM

```
Function (A)OMUS(φ:  CNF formula) :  CNF formula
    stack = ∅;
    While ((LS+score(φ) does not find a model of φ)) do
        push(φ);
        φ ← φ − φ_LowestScore ;
    done
    Repeat
        φ = pop();
    until (UNSAT(φ))
    [For OMUS]
    Fine-Tune(φ);
    Return φ;
End
```

EXPERIMENTAL RESULTS

Instance	zCore [Zhang & Malik 03]	[Lynce & M.-Silva 04]	[Bruni 03] [1]	AOMUS (falsified clauses)	AOMUS
aim-50-2_0-no-2	30 (1,88)	30 (0,90)	31	30 (1,79)	30 (2,61)
aim-50-2_0-no-4	21 (1,29)	21 (3,49)	21	21 (2,97)	21 (2,85)
aim-100-1_6-no-1	47 (1,45)	47 (284)	47	47 (2,62)	47 (2,67)
aim-100-1_6-no-2	54 (1,12)	53 (224)	54	53 (2,37)	53 (2,82)
aim-100-1_6-no-3	57 (1,23)	time out	57	57 (1,87)	57 (3,20)
aim-100-1_6-no-4	48 (0,95)	48 (241)	48	48 (1,86)	48 (2,84)
aim-200-1_6-no-2	81 (1,52)	time out	82	80 (1,79)	80 (2,94)
jnh11	121 (2,46)	time out	129	225 (13)	167 (29)
jnh13	57 (1,90)	time out	106	90 (41)	66 (77)
jnh14	91 (1,85)	time out	124	111 (45)	90 (89)
jnh2	45 (1,95)	time out	60	117 (56)	74 (50)
jnh5	86 (1,79)	time out	125	143 (39)	114 (61)
jnh8	90 (2,28)	time out	91	118 (65)	76 (102)
fpga10_11_uns	561 (27)	time out	-	565 (15)	561 (26)
fpga10_12_uns	672 (65)	time out	-	568 (66)	561 (57)
homer10.shuffled	940 (624)	time out	-	518 (818)	415 (496)
homer11.shuffled	561 (25)	time out	-	564 (16)	561 (26)
homer14.shuffled	1065 (714)	time out	-	561 (536)	561 (449)
homer15.shuffled	time out	time out	-	677 (1299)	561 (1104)

[1] extracted from [Bruni 03]

1. MUSes & Inconsistent Covers
   - Definitions and properties
   - Motivations

2. *(A)OMUS*: a MUS extractor
   - Deciding which clauses belong to a MUS
   - Taking the neighborhood of the current interpretation into account
   - Algorithm and Experimental Results

3. Computing one strict inconsistent cover
   - Algorithm and Experimental Results

4. Conclusions and future work

Strict Inconsistent Cover

Motivations

- Goal :
  - delivering the source(s) of inconsistency
  - helping in satisfiability restoring

Strict Inconsistent Cover

Motivations

- Goal :
  - delivering the source(s) of inconsistency
  - helping in satisfiability restoring
- Is computing all MUSes of the formula tractable ?

Strict Inconsistent Cover

Motivations

- Goal :
  - delivering the source(s) of inconsistency
  - helping in satisfiability restoring
- Is computing all MUSes of the formula tractable ?
- **Problem :** A *n*-clauses formula can exhibit $C_n^{n/2}$ MUSes in the worst case

Strict Inconsistent Cover

Motivations

- Goal :
  - delivering the source(s) of inconsistency
  - helping in satisfiability restoring
- Is computing all MUSes of the formula tractable ?
- **Problem :** A n-clauses formula can exhibit $C_n^{n/2}$ MUSes in the worst case
- $\rightarrow$ Intractable computation

Strict Inconsistent Cover

Motivations

- Goal :
  - delivering the source(s) of inconsistency
  - helping in satisfiability restoring
- Is computing all MUSes of the formula tractable ?
- **Problem :** A *n*-clauses formula can exhibit $C_n^{n/2}$ MUSes in the worst case
- $\rightarrow$ Intractable computation
- We need to compute independent causes of unsatisfiability $\Rightarrow$ concept of Strict Inconsistent Cover

ICMUS ALGORITHM

Function ICMUS(ϕ: **CNF formula**) : **a strict Inconsistent Cover**

$IC \longleftarrow \emptyset$;

While ((Σ is unsatisfiable)) **do**

$MUS \longleftarrow$ `OMUS`(Σ) ;

$IC \longleftarrow IC \cup MUS$;

$\Sigma \longleftarrow \Sigma \setminus MUS$;

done

return IC ;

End

Algorithm 1: `ICMUS` algorithm

Experimental results

TABLE: Inconsistent covers for various classes of formulas

Instance	#var	#cla	Time	**#MUSes** in the IC	
dp02u01	213	376	1.19	**1**	(47,51)
dp03u02	478	1007	362	**1**	(327,760)
fpga10_11_uns_rcr	220	1122	56	**2**	(110,561) (110,561)
fpga11_12_uns_rcr	264	1476	128	**2**	(132,738) (132,738)
ca002	26	70	0.61	**1**	(20,39)
ca004	60	168	1.11	**1**	(49,108)
ca008	130	370	5.26	**1**	(110,255)
term1_gr_rcs_w3	606	2518	6180	**11**	(12,22) (21,33) (30,58) (12,22) (12,22) (12,22) (12,22) (12,22) (12,22) (24,39) (21,33)
C220_FV_RZ_14	1728	4508	28	**1**	(10,14)
C220_FV_RZ_13	1728	4508	46	**1**	(9,13)
C170_FR_SZ_96	1659	4955	18	**1**	(81,233)
C208_FA_SZ_121	1608	5278	21	**1**	(18,32)
C168_FW_UT_851	1909	7491	83	**1**	(7,9)
C202_FW_UT_2814	2038	11352	304	**1**	(15,18)
jnh208	100	800	14	**1**	(76,119)
jnh302	100	900	63	**2**	(27,28) (98,208)
jnh310	100	900	184	**2**	(12,13) (90,188)
3col40_5_3	80	346	4.64	**1**	(64,136)
fphp-012-010	120	1212	57	**1**	(120,670)

1. MUSes & Inconsistent Covers
   - Definitions and properties
   - Motivations

2. *(A)OMUS*: a MUS extractor
   - Deciding which clauses belong to a MUS
   - Taking the neighborhood of the current interpretation into account
   - Algorithm and Experimental Results

3. Computing one strict inconsistent cover
   - Algorithm and Experimental Results

4. **Conclusions and future work**

Conclusions and future work

Contributions

Theoretical and practical applications of the new notion of critical clause

- **Theoretical:** For each clause belonging to a MUS, there exists an interpretation s.t. it can be critical.
- **Practical:** Exploitation of this property in order to extract:
  - An approximation or an exact MUS
  - An inconsistent cover

Conclusions and future work

Contributions

Theoretical and practical applications of the new notion of critical clause

- **Theoretical:** For each clause belonging to a MUS, there exists an interpretation s.t. it can be critical.
- **Practical:** Exploitation of this property in order to extract:
  - An approximation or an exact MUS
  - An inconsistent cover

Future Work

- Specific treatment of long clauses
- Certificates for:
  - The smallest inconsistent cover(s)
  - The set of MUSes
- Apply this work for *MaxSAT* practical resolution.
- ...