Tracking MUSes and Strict Inconsistent Covers

Éric Grégoire  Bertrand Mazure  Cédric Piette

November 13, 2006

Formal Methods in Computer Aided Design (FMCAD’2006)
1 MUSés & 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
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 ($l_1 \lor l_2 \lor \ldots \lor l_n$)
- **CNF formula**: finite conjunction of clauses ($c_1 \land c_2 \land \ldots \land c_m$)
DEFINITIONS AND PROPERTIES

DEFINITION: CNF formula

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

DEFINITION: Interpretation

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

FMCAD’2006
Tracking MUSes and Strict Inconsistent Covers
DEFINITIONS AND PROPERTIES

DEFINITION: CNF formula

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

DEFINITION: Interpretation

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

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 ($l_1 \lor l_2 \lor ... \lor l_n$)
- **CNF formula**: finite conjunction of clauses ($c_1 \land c_2 \land ... \land c_m$)

**Definition: Interpretation**

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

**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 $\phi$ 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 $\phi$ 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 \( \phi \) 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 \( \phi \) is a subset of \( KS_\phi \) such that its removal restores the satisfiability of \( \phi \).

A strict inconsistent cover is composed of independent MUSes.
DEFINITIONS AND PROPERTIES

Example

\[ \Phi \]

\begin{align*}
A & \cap B \\
& \cap D \\
& \cap C \\
& \cap E
\end{align*}

MUS
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\}
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 \( \phi \) be an inconsistent \( n \)-clauses CNF formula and \( SIC_\phi \) be a strict inconsistent cover of \( \phi \). Then we have:

\[
\text{MaxSat}(\phi) \leq n - |SIC_\phi|
\]
**Definitions and Properties**

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

**Property**
Let $\phi$ be an inconsistent $n$-clauses CNF formula and $SIC_\phi$ be a strict inconsistent cover of $\phi$. Then we have:

$$\text{MaxSat}(\phi) \leq n - |SIC_\phi|$$

**Relation Between MaxSat and MUSes**
Let $\omega$ be an optimal interpretation for MaxSat, any falsified clause w.r.t. $\omega$ 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 \( \Sigma_{2}^{P} \)

  [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 $\phi$ be a CNF formula, $K$ a MUS of $\phi$, and $c$ a clause. For all interpretations $\omega$, $\exists c \in K$ s.t. $\omega \not\models c$
Let $\phi$ be a CNF formula, $K$ a MUS of $\phi$, and $c$ a clause. For all interpretations $\omega$, $\exists c \in K$ 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 $\phi$ be a CNF formula, $K$ a MUS of $\phi$, and $c$ a clause. For all interpretations $\omega$, $\exists c \in K$ 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 $\phi$ be a CNF formula, $K$ a MUS of $\phi$, and $c$ a clause. For all interpretations $\omega$, $\exists c \in K$ 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.
Definition: *once-satisfied clause*

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

Definition: *critical clause*

A clause $c$ falsified w.r.t. an interpretation $\omega$ 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

\[(a \lor b \lor c) \land (\neg b \lor e) \land (\neg a \lor b \lor c) \land (\neg a \lor \neg b) \land (a \lor d) \land (b \lor \neg c) \land (\neg d \lor e) \land (a \lor \neg b) \land (\neg e \lor \neg f)\]
Example

clauses belonging to MUS:

\[(a \lor b \lor c) \leftarrow\]
\[\land (\neg b \lor e) \leftarrow\]
\[\land (\neg a \lor b \lor c) \leftarrow\]
\[\land (\neg a \lor \neg b) \leftarrow\]
\[\land (a \lor d) \leftarrow\]
\[\land (b \lor \neg c) \leftarrow\]
\[\land (\neg d \lor e) \leftarrow\]
\[\land (a \lor \neg b) \leftarrow\]
\[\land (\neg e \lor \neg f) \leftarrow\]
Example

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

Clauses belonging to MUS:

\[ (a \lor b \lor c) \]
\[ \land (\neg b \lor e) \]
\[ \land (\neg a \lor b \lor c) \]
\[ \land (\neg a \lor \neg b) \]
\[ \land (a \lor d) \]
\[ \land (b \lor \neg c) \]
\[ \land (\neg d \lor e) \]
\[ \land (a \lor \neg b) \]
\[ \land (\neg e \lor \neg f) \]
$\omega = \{\neg a, \neg b, c, d, e, f\}$

**Example**

**Property**

Let $c$ be a critical clause w.r.t. an interpretation $\omega$.

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

clauses belonging to MUS:

- $(a \lor b \lor c)$
- $\land (\neg b \lor e)$
- $\land (\neg a \lor b \lor c)$
- $\land (\neg a \lor \neg b)$
- $\land (a \lor d)$
- $\land (b \lor \neg c)$
- $\land (\neg d \lor e)$
- $\land (a \lor \neg b)$
- $\land (\neg e \lor \neg f)$
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.
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 $\omega$ 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 $\omega$ 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. $c \in K$

$$\Downarrow$$

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

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

$$\Downarrow$$

$\omega \not\models c$

$$\Downarrow$$

Let $\omega'$ 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 $\omega$ be a model of $K\backslash\{c\}$
\[\Downarrow\]
$\omega \not\models c$
\[\Downarrow\]
Let $\omega'$ 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. $c \in K$

$\Downarrow$

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

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

$\Downarrow$

$\omega \not\models c$

$\Downarrow$

Let $\omega'$ 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. $\omega$)
**Why counting critical clauses?**

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

$\Downarrow$

$K\{c\}$ is SAT.

Let $\omega$ be a model of $K\{c\}$

$\Downarrow$

$\omega \not\models c$

$\Downarrow$

Let $\omega'$ 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. $\omega$)

**Property**

For each clause $c$ in a MUS, there exists an interpretation $\omega$ s.t. $c$ is critical.
**WHY COUNTING CRITICAL CLAUSES?**

**PROPERTY**
For each clause $c$ in a MUS, there exists an interpretation $\omega$ s.t. $c$ is critical.

**EXTENSION OF THE RELATIONSHIP BETWEEN MAXSAT AND MUSes**
Let $\omega$ be an optimal interpretation for MaxSat, any falsified clause $c$ w.r.t. $\omega$:
- belongs to at least one MUS of the CNF formula
- is critical w.r.t. $\omega$
- at least one once-satified clause linked to $c$ belongs to the same MUS
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

<table>
<thead>
<tr>
<th>Instance</th>
<th>zCore [Zhang &amp; Malik 03]</th>
<th>[Lynce &amp; M.-Silva 04]</th>
<th>[Bruni 03] (^1)</th>
<th>AOMUS (falsified clauses)</th>
<th>AOMUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>aim-50-2_0-no-2</td>
<td>30 (1.88)</td>
<td>30 (0.90)</td>
<td>31</td>
<td>30 (1.79)</td>
<td>30 (2.61)</td>
</tr>
<tr>
<td>aim-50-2_0-no-4</td>
<td>21 (1.29)</td>
<td>21 (3.49)</td>
<td>21</td>
<td>21 (2.97)</td>
<td>21 (2.85)</td>
</tr>
<tr>
<td>aim-100-1_6-no-1</td>
<td>47 (1.45)</td>
<td>47 (284)</td>
<td>47</td>
<td>47 (2.62)</td>
<td>47 (2.67)</td>
</tr>
<tr>
<td>aim-100-1_6-no-2</td>
<td>54 (1.12)</td>
<td>53 (224)</td>
<td>54</td>
<td>53 (2.37)</td>
<td>53 (2.82)</td>
</tr>
<tr>
<td>aim-100-1_6-no-3</td>
<td>57 (1.23) time out</td>
<td>57</td>
<td>57 (1.87)</td>
<td>57 (3.20)</td>
<td></td>
</tr>
<tr>
<td>aim-100-1_6-no-4</td>
<td>48 (0.95)</td>
<td>48 (241)</td>
<td>48</td>
<td>48 (1.86)</td>
<td>48 (2.84)</td>
</tr>
<tr>
<td>aim-200-1_6-no-2</td>
<td>81 (1.52) time out</td>
<td>82</td>
<td>80 (1.79)</td>
<td>80 (2.94)</td>
<td></td>
</tr>
<tr>
<td>jnh11</td>
<td>121 (2.46) time out</td>
<td>129</td>
<td>225 (13)</td>
<td>167 (29)</td>
<td></td>
</tr>
<tr>
<td>jnh13</td>
<td>57 (1.90) time out</td>
<td>106</td>
<td>90 (41)</td>
<td>66 (77)</td>
<td></td>
</tr>
<tr>
<td>jnh14</td>
<td>91 (1.85) time out</td>
<td>124</td>
<td>111 (45)</td>
<td>90 (89)</td>
<td></td>
</tr>
<tr>
<td>jnh2</td>
<td>45 (1.95) time out</td>
<td>60</td>
<td>117 (56)</td>
<td>74 (50)</td>
<td></td>
</tr>
<tr>
<td>jnh5</td>
<td>86 (1.79) time out</td>
<td>125</td>
<td>143 (39)</td>
<td>114 (61)</td>
<td></td>
</tr>
<tr>
<td>jnh8</td>
<td>90 (2.28) time out</td>
<td>91</td>
<td>118 (65)</td>
<td>76 (102)</td>
<td></td>
</tr>
<tr>
<td>fpga10_11_uns</td>
<td>561 (27) time out</td>
<td>-</td>
<td>565 (15)</td>
<td>561 (26)</td>
<td></td>
</tr>
<tr>
<td>fpga10_12_uns</td>
<td>672 (65) time out</td>
<td>-</td>
<td>568 (66)</td>
<td>561 (57)</td>
<td></td>
</tr>
<tr>
<td>homer10.shuffled</td>
<td>940 (624) time out</td>
<td>-</td>
<td>518 (818)</td>
<td>415 (496)</td>
<td></td>
</tr>
<tr>
<td>homer11.shuffled</td>
<td>561 (25) time out</td>
<td>-</td>
<td>564 (16)</td>
<td>561 (26)</td>
<td></td>
</tr>
<tr>
<td>homer14.shuffled</td>
<td>1065 (714) time out</td>
<td>-</td>
<td>561 (536)</td>
<td>561 (449)</td>
<td></td>
</tr>
<tr>
<td>homer15.shuffled</td>
<td>time out time out</td>
<td>-</td>
<td>677 (1299)</td>
<td>561 (1104)</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) 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 $\binom{n}{n/2}$ MUSes in the worst case
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 $C_{n/2}^n$ MUSes in the worst case
  - $\rightarrow$ Intractable computation

- We need to compute independent causes of unsatisfiability $\Rightarrow$ concept of Strict Inconsistent Cover
Function \( \text{ICMUS}(\phi: \text{CNF formula}) : \text{a strict Inconsistent Cover} \)

\[
\begin{align*}
\text{IC} & \leftarrow \emptyset ; \\
\text{While} \ ((\Sigma \text{ is unsatisfiable})) \text{ do} & \\
& \quad \text{MUS} \leftarrow \text{OMUS}(\Sigma) ; \\
& \quad \text{IC} \leftarrow \text{IC} \cup \text{MUS} ; \\
& \quad \Sigma \leftarrow \Sigma \setminus \text{MUS} ; \\
\text{done} & \\
\text{return IC} ; \\
\end{align*}
\]

End

Algorithm 1: ICMUS algorithm
### Experimental Results

**Table**: Inconsistent covers for various classes of formulas

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

FMCAD’2006  Tracking MUSes and Strict Inconsistent Covers  22 / 22