Accelerating MUS Extraction with Recursive Model Rotation

Anton Belov and Joao Marques-Silva

Complex and Adaptive Systems Laboratory
School of Computer Science and Informatics
University College Dublin, Ireland

FMCAD 2011
October 31, 2011
Austin, TX, USA
Minimal Unsatisfiability

- $F$ is *minimally unsatisfiable* ($F \in MU$), if $F \in UNSAT$ and for any $C \in F$, $F \setminus C \in SAT$. 

Example

$\{C_1, C_2, C_3, C_4\} \in MU$. 

A. Belov, J. Marques-Silva (UCD, Dublin)
Minimal Unsatisfiability

- \( F \) is **minimally unsatisfiable** \((F \in \text{MU})\), if \( F \in \text{UNSAT} \) and for any \( C \in F \), \( F \setminus C \in \text{SAT} \).

- \( F' \) is **minimally unsatisfiable subformula (MUS)** of \( F \) \((F' \in \text{MUS}(F))\) if \( F' \subseteq F \) and \( F' \in \text{MU} \).
Minimal Unsatisfiability

- $F$ is **minimally unsatisfiable** ($F \in \text{MU}$), if $F \in \text{UNSAT}$ and for any $C \in F$, $F \setminus C \in \text{SAT}$.

- $F'$ is **minimally unsatisfiable subformula (MUS)** of $F$ ($F' \in \text{MUS}(F)$) if $F' \subseteq F$ and $F' \in \text{MU}$.

Example

$$C_1 = x \lor y \quad C_3 = x \lor \neg y$$

$$C_2 = \neg x \lor y \quad C_4 = \neg x \lor \neg y$$

- $\{C_1, C_2, C_3, C_4\} \in \text{MU}$.
Introduction

Minimal Unsatisfiability

- $F$ is minimally unsatisfiable ($F \in \text{MU}$), if $F \in \text{UNSAT}$ and for any $C \in F$, $F \setminus C \in \text{SAT}$.

- $F'$ is minimally unsatisfiable subformula (MUS) of $F$ ($F' \in \text{MUS}(F)$) if $F' \subseteq F$ and $F' \in \text{MU}$.

Example

\[
\begin{align*}
C_1 &= x \lor y & C_3 &= x \lor \neg y & C_5 &= y \lor z \\
C_2 &= \neg x \lor y & C_4 &= \neg x \lor \neg y & C_6 &= y \lor \neg z
\end{align*}
\]

- \{ $C_1, C_2, C_3, C_4$ \} $\in \text{MU}$.
- $F = \{ C_1, \ldots, C_6 \} \in \text{UNSAT}$, but $\notin \text{MU}$. 

A. Belov, J. Marques-Silva (UCD, Dublin)
Minimal Unsatisfiability

- $F$ is \textit{minimally unsatisfiable} ($F \in \text{MU}$), if $F \in \text{UNSAT}$ and for any $C \in F$, $F \setminus C \in \text{SAT}$.

- $F'$ is \textit{minimally unsatisfiable subformula (MUS)} of $F$ ($F' \in \text{MUS}(F)$) if $F' \subseteq F$ and $F' \in \text{MU}$.

Example

\[
\begin{align*}
C_1 &= x \lor y & C_3 &= x \lor \neg y \\
C_2 &= \neg x \lor y & C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

- \{ $C_1, C_2, C_3, C_4$ \} $\in \text{MU}$.
- \{ $C_1, C_2, C_3, C_4$ \} $\in \text{MUS}(F)$.
Minimal Unsatisfiability

- $F$ is \textit{minimally unsatisfiable} ($F \in \text{MU}$), if $F \in \text{UNSAT}$ and for any $C \in F$, $F \setminus C \in \text{SAT}$.

- $F'$ is \textit{minimally unsatisfiable subformula (MUS)} of $F$ ($F' \in \text{MUS}(F)$) if $F' \subseteq F$ and $F' \in \text{MU}$.

Example

\[
\begin{align*}
C_1 &= x \lor y \\
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

- $\{C_1, C_2, C_3, C_4\} \in \text{MU}$.
- $\{C_3, C_4, C_5, C_6\} \in \text{MUS}(F)$. 

A. Belov, J. Marques-Silva  (UCD, Dublin)  Recursive Model Rotation  FMCAD 2011  2 / 16
Minimal Unsatisfiability

- $F$ is **minimally unsatisfiable** ($F \in MU$), if $F \in UNSAT$ and for any $C \in F$, $F \setminus C \in SAT$.

- $F'$ is **minimally unsatisfiable subformula (MUS)** of $F$ ($F' \in MUS(F)$) if $F' \subseteq F$ and $F' \in MU$.

Applications of MUSes (in formal methods)

- Abstraction refinement frameworks.
- Decision procedures.
- Design debugging.
Computation of MUSes

- Based on iterative calls to SAT solver (not the only way, but currently the most effective): for each $C \in F$
  - if $F \setminus \{C\} \in \text{UNSAT}$, then there is an MUS of $F$ that does not contain $C$. Remove $C$ from $F$.
  - if $F \setminus \{C\} \in \text{SAT}$ ($C$ is necessary for $F$), then $C$ is in all MUSes of $F$. Keep $C$.

SAT solving is the main bottleneck of the computation, hence reduction in the number of SAT solver calls is the key to efficiency.

- On UNSAT outcomes – clause set refinement: remove $C$ and all clauses outside the unsatisfiable core of $F \setminus \{C\}$.
  - [Dershowitz et al’06]

- On SAT outcomes – model rotation: detect additional necessary clauses without SAT solver calls.
  - [Marques-Silva&Lynce’11]

Recursive model rotation (RMR) – very effective improvement of model rotation.
- [this paper]
Computation of MUSes

- Based on iterative calls to SAT solver (not the only way, but currently the most effective): for each $C \in F$
  - if $F \setminus \{C\} \in$ UNSAT, then there is an MUS of $F$ that does not contain $C \rightarrow$ remove $C$ from $F$.
  - if $F \setminus \{C\} \in$ SAT ($C$ is necessary for $F$), then $C$ is in all MUSes of $F$ \rightarrow$ keep $C$.

- SAT solving is the main bottleneck of the computation, hence reduction in the number of SAT solver calls is the key to efficiency.
Computation of MUSes

- Based on iterative calls to SAT solver (not the only way, but currently the most effective): for each $C \in F$
  - if $F \setminus \{C\} \in \text{UNSAT}$, then there is an MUS of $F$ that does not contain $C$ → remove $C$ from $F$.
  - if $F \setminus \{C\} \in \text{SAT}$ ($C$ is necessary for $F$), then $C$ is in all MUSes of $F$ → keep $C$.

- SAT solving is the main bottleneck of the computation, hence reduction in the number of SAT solver calls is the key to efficiency.

- On UNSAT outcomes – clause set refinement: remove $C$ and all clauses outside the unsatisfiable core of $F \setminus \{C\}$. [Dershowitz et al’06]


Recursive model rotation (RMR) – very effective improvement of model rotation. [this paper]
Computation of MUSes

- Based on iterative calls to SAT solver (not the only way, but currently the most effective): for each $C \in F$
  - if $F \setminus \{C\} \in \text{UNSAT}$, then there is an MUS of $F$ that does not contain $C \rightarrow$ remove $C$ from $F$.
  - if $F \setminus \{C\} \in \text{SAT}$ ($C$ is necessary for $F$), then $C$ is in all MUSes of $F \rightarrow$ keep $C$.

- SAT solving is the main bottleneck of the computation, hence reduction in the number of SAT solver calls is the key to efficiency.

- On UNSAT outcomes – clause set refinement: remove $C$ and all clauses outside the unsatisfiable core of $F \setminus \{C\}$. [Dershowitz et al’06]


  Recursive model rotation (RMR) – very effective improvement of model rotation. [this paper]
Impact of RMR

- 500 benchmarks submitted to MUS track of SAT Competition 2011.
- Time limit 1200 sec, memory limit 4 GB.

MUS computation without RMR (x-axis) vs with RMR (y-axis)
  - Left: number of SAT solver calls (on instances solved in both cases).
Impact of RMR

- 500 benchmarks submitted to MUS track of SAT Competition 2011.
- Time limit 1200 sec, memory limit 4 GB.

- MUS computation without RMR (x-axis) vs with RMR (y-axis)
  - Left: number of SAT solver calls (on instances solved in both cases).
  - Right: CPU time (sec).
Computation of MUSes

Use SAT solver to identify necessary (or, transition) clauses

- $C \in F$ is necessary for $F$, if $F \in \text{UNSAT}$ and $F \setminus \{C\} \in \text{SAT}$. 
Computation of MUSes

Use SAT solver to identify necessary (or, transition) clauses

- $C \in F$ is necessary for $F$, if $F \in \text{UNSAT}$ and $F \setminus \{C\} \in \text{SAT}$.
- $F \in \text{MU}$ iff every clause $C \in F$ is necessary for $F$. 

Deletion-based MUS Computation

Input: $F$ — an unsatisfiable CNF formula

$M \leftarrow F$

// Inv: $M$ is a superset of some MUS of $F$

foreach $C \in F$ do

if $M \setminus \{C\} \in \text{UNSAT}$ then

// is $C$ necessary for $M$?

// no - delete it

$M \leftarrow M \setminus \{C\}$

// yes - keep it

return $M$

// Every $C \in M$ is necessary for $M$.
Computation of MUSes

Use SAT solver to identify necessary (or, transition) clauses

- $C \in F$ is necessary for $F$, if $F \in$ UNSAT and $F \setminus \{C\} \in$ SAT.
- $F \in$ MU iff every clause $C \in F$ is necessary for $F$.
- If $C$ is necessary for $F$ then $C$ is necessary for every unsatisfiable subset of $F$. 

Deletion-based MUS Computation

Input: $F$ — an unsatisfiable CNF formula

```
M ← F // Inv: M is a superset of some MUS of F
foreach C ∈ F do
  if M \{C\} ∈ UNSAT then
    // is C necessary for M? no - delete it
    M ← M \{C\} // yes - keep it
return M // Every C ∈ M is necessary for M
```
Computation of MUSes

Use SAT solver to identify necessary (or, transition) clauses

▶ $C \in F$ is necessary for $F$, if $F \in \text{UNSAT}$ and $F \setminus \{C\} \in \text{SAT}$.
▶ $F \in \text{MU}$ iff every clause $C \in F$ is necessary for $F$.
▶ If $C$ is necessary for $F$ then $C$ is necessary for every unsatisfiable subset of $F$.

Deletion-based MUS Computation

Input: $F$ — an unsatisfiable CNF formula

$M \leftarrow F$ // Inv: $M$ is a superset of some MUS of $F$

foreach $C \in F$ do

if $M \setminus \{C\} \in \text{UNSAT}$ then // is $C$ necessary for $M$?

// no - delete it

$M \leftarrow M \setminus \{C\}$

// yes - keep it

return $M$ // Every $C \in M$ is necessary for $M$
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_1 &= x \lor y \\
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = F \in \text{UNSAT} \)
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_1 &= x \lor y \\
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT}, \) hence \( C_1 \) is not necessary
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[ C_1 = x \lor y \]
\[ C_2 = \neg x \lor y \]
\[ C_3 = x \lor \neg y \]
\[ C_4 = \neg x \lor \neg y \]
\[ C_5 = y \lor z \]
\[ C_6 = y \lor \neg z \]

\[ M = F \in \text{UNSAT} \]
\[ M \setminus \{ C_1 \} \in \text{UNSAT}, \text{ hence } C_1 \text{ is not necessary } \rightarrow M = M \setminus \{ C_1 \} \]
**Example**

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[ C_3 = x \lor \neg y \quad C_5 = y \lor z \]

\[ C_2 = \neg x \lor y \quad C_4 = \neg x \lor \neg y \quad C_6 = y \lor \neg z \]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT}, \) hence \( C_1 \) is not necessary \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT}, \) hence \( C_3 \) is necessary
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_1 &= \neg x \lor y \\
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT}, \) hence \( C_1 \) is not necessary \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT}, \) hence \( C_3 \) is necessary \( \rightarrow \) keep \( C_3 \)
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[ C_1 = x \lor \neg y \quad \quad C_5 = y \lor z \]
\[ C_2 = \neg x \lor y \quad \quad C_4 = \neg x \lor \neg y \quad \quad C_6 = y \lor \neg z \]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT} \), hence \( C_1 \) is not necessary \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary \( \rightarrow \) keep \( C_3 \)

\( M \setminus \{ C_5 \} \in \text{SAT} \), hence \( C_5 \) is necessary
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_5 &= y \lor z \\
C_2 &= \neg x \lor y \\
C_4 &= \neg x \lor \neg y \\
C_6 &= y \lor \neg z
\end{align*}
\]

\[ M = F \in \text{UNSAT} \]

\[ M \setminus \{ C_1 \} \in \text{UNSAT}, \text{ hence } C_1 \text{ is not necessary } \rightarrow M = M \setminus \{ C_1 \} \]

\[ M \setminus \{ C_3 \} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary } \rightarrow \text{keep } C_3 \]

\[ M \setminus \{ C_5 \} \in \text{SAT}, \text{ hence } C_5 \text{ is necessary } \rightarrow \text{keep } C_5 \]
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT} \), hence \( C_1 \) is \underline{not necessary} \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is \underline{necessary} \( \rightarrow \) keep \( C_3 \)

\( M \setminus \{ C_5 \} \in \text{SAT} \), hence \( C_5 \) is \underline{necessary} \( \rightarrow \) keep \( C_5 \)

\( M \setminus \{ C_2 \} \in \text{UNSAT} \), hence \( C_2 \) is \underline{not necessary}
Example

\[ F = \{C_1, \ldots, C_6\} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
C_3 = x \lor \neg y \\
C_5 = y \lor z
\]

\[
C_4 = \neg x \lor \neg y \\
C_6 = y \lor \neg z
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{C_1\} \in \text{UNSAT}, \text{ hence } C_1 \text{ is not necessary } \rightarrow M = M \setminus \{C_1\} \)

\( M \setminus \{C_3\} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary } \rightarrow \text{ keep } C_3 \)

\( M \setminus \{C_5\} \in \text{SAT}, \text{ hence } C_5 \text{ is necessary } \rightarrow \text{ keep } C_5 \)

\( M \setminus \{C_2\} \in \text{UNSAT}, \text{ hence } C_2 \text{ is not necessary } \rightarrow M = M \setminus \{C_2\} \)
Example

\[ F = \{C_1, \ldots, C_6\} \]

\[ M (an \ overapproximation\ of\ some\ \text{MUS} \ of\ F): \]

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_5 &= y \lor z \\
C_4 &= \neg x \lor \neg y \\
C_6 &= y \lor \neg z
\end{align*}
\]

\[ M = F \in \text{UNSAT} \]

\[ M \setminus \{C_1\} \in \text{UNSAT}, \text{ hence } C_1 \text{ is not necessary } \rightarrow M = M \setminus \{C_1\} \]

\[ M \setminus \{C_3\} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary } \rightarrow \text{keep } C_3 \]

\[ M \setminus \{C_5\} \in \text{SAT}, \text{ hence } C_5 \text{ is necessary } \rightarrow \text{keep } C_5 \]

\[ M \setminus \{C_2\} \in \text{UNSAT}, \text{ hence } C_2 \text{ is not necessary } \rightarrow M = M \setminus \{C_2\} \]

\[ M \setminus \{C_4\} \in \text{SAT}, \text{ hence } C_4 \text{ is necessary} \]
\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT} \), hence \( C_1 \) is not necessary \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary \( \rightarrow \) keep \( C_3 \)

\( M \setminus \{ C_5 \} \in \text{SAT} \), hence \( C_5 \) is necessary \( \rightarrow \) keep \( C_5 \)

\( M \setminus \{ C_2 \} \in \text{UNSAT} \), hence \( C_2 \) is not necessary \( \rightarrow M = M \setminus \{ C_2 \} \)

\( M \setminus \{ C_4 \} \in \text{SAT} \), hence \( C_4 \) is necessary \( \rightarrow \) keep \( C_4 \)

Each clause in \( F \setminus M \) costs one SAT solver call.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\[ M = F \in \text{UNSAT} \]
\[ M \setminus \{ C_1 \} \in \text{UNSAT}, \text{ hence } C_1 \text{ is not necessary } \rightarrow M = M \setminus \{ C_1 \} \]
\[ M \setminus \{ C_3 \} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary } \rightarrow \text{ keep } C_3 \]
\[ M \setminus \{ C_5 \} \in \text{SAT}, \text{ hence } C_5 \text{ is necessary } \rightarrow \text{ keep } C_5 \]
\[ M \setminus \{ C_2 \} \in \text{UNSAT}, \text{ hence } C_2 \text{ is not necessary } \rightarrow M = M \setminus \{ C_2 \} \]
\[ M \setminus \{ C_4 \} \in \text{SAT}, \text{ hence } C_4 \text{ is necessary } \rightarrow \text{ keep } C_4 \]
\[ M \setminus \{ C_6 \} \in \text{SAT}, \text{ hence } C_6 \text{ is necessary} \]
**Example**

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
C_3 = x \lor \lnot y \\
C_4 = \lnot x \lor \lnot y \\
C_5 = y \lor z \\
C_6 = y \lor \lnot z
\]

\( M = F \in \text{UNSAT} \)

\( M \setminus \{ C_1 \} \in \text{UNSAT} \), hence \( C_1 \) is not necessary \( \rightarrow M = M \setminus \{ C_1 \} \)

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary \( \rightarrow \) keep \( C_3 \)

\( M \setminus \{ C_5 \} \in \text{SAT} \), hence \( C_5 \) is necessary \( \rightarrow \) keep \( C_5 \)

\( M \setminus \{ C_2 \} \in \text{UNSAT} \), hence \( C_2 \) is not necessary \( \rightarrow M = M \setminus \{ C_2 \} \)

\( M \setminus \{ C_4 \} \in \text{SAT} \), hence \( C_4 \) is necessary \( \rightarrow \) keep \( C_4 \)

\( M \setminus \{ C_6 \} \in \text{SAT} \), hence \( C_6 \) is necessary \( \rightarrow \) keep \( C_6 \)
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = \{ C_3, C_4, C_5, C_6 \} \) is an MUS of \( F \).
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y & C_5 &= y \lor z \\
C_4 &= \neg x \lor \neg y & C_6 &= y \lor \neg z
\end{align*}
\]

\( M = \{ C_3, C_4, C_5, C_6 \} \) is an MUS of \( F \).

- Each clause in \( F \setminus M \) costs one SAT solver call.
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M = \{ C_3, C_4, C_5, C_6 \} \) is an MUS of \( F \).

- Each clause in \( F \setminus M \) costs \( \leq 1 \) SAT solver call – clause set refinement.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\[ M \] (an overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\[ M = \{ C_3, C_4, C_5, C_6 \} \] is an MUS of \( F \).

- Each clause in \( F \setminus M \) costs \( \leq 1 \) SAT solver call – clause set refinement.
- Each clause in \( M \) costs one SAT solver call.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (an overapproximation of some MUS of \( F \)):

\[ C_3 = x \lor \neg y \quad C_5 = y \lor z \]
\[ C_4 = \neg x \lor \neg y \quad C_6 = y \lor \neg z \]

\( M = \{ C_3, C_4, C_5, C_6 \} \) is an MUS of \( F \).

▶ Each clause in \( F \setminus M \) costs \( \leq 1 \) SAT solver call – clause set refinement.

▶ Each clause in \( M \) costs \( \leq 1 \) SAT solver call – model rotation.
Fact: $C$ is necessary for $F$ iff $F \in \text{UNSAT}$ and $\exists \tau$ such that $\text{Unsat}(F, \tau) = \{C\}$. $\tau$ is a witness (of necessity) for $C$. 
Fact: $C$ is necessary for $F$ iff $F \in \text{UNSAT}$ and $\exists \tau$ such that $\text{Unsat}(F, \tau) = \{C\}$. $\tau$ is a witness (of necessity) for $C$.

During MUS extraction: when $M \setminus \{C\} \in \text{SAT}$, the assignment $\tau$ found by the SAT solver is a witness for $C$. 

Fact: $C$ is necessary for $F$ iff $F \in \text{UNSAT}$ and $\exists \tau$ such that $\text{Unsat}(F, \tau) = \{C\}$. $\tau$ is a witness (of necessity) for $C$.

During MUS extraction: when $M \setminus \{C\} \in \text{SAT}$, the assignment $\tau$ found by the SAT solver is a witness for $C$.

Model rotation: given a witness $\tau$ for $C$, try to modify it into a witness $\tau'$ for another clause $C'$. How?
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z \\
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z \\
C_2 &= \neg x \lor y \\
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary.} \)

SAT solver returns \( \tau = \{ \neg x, y, z \} \)
**Example**

Let $F = \{ C_1, \ldots, C_6 \}$

$M$ (the overapproximation of some MUS of $F$):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

$M \setminus \{ C_3 \} \in \text{SAT}$, hence $C_3$ is necessary.

SAT solver returns $\tau = \{ \neg x, y, z \}$, $\text{Unsat}(M, \tau) = \{ C_3 \}$. 
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \)
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z \\
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

\( \text{SAT solver returns} \ \tau = \{ \neg x, y, z \}, \ \text{Unsat}(M, \tau) = \{ C_3 \} \).

\( \text{Flip } x \ \text{in } \tau : \ \tau' = \{ x, y, z \}, \ \text{Unsat}(M, \tau') = \{ C_4 \} \)
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_5 &= y \lor z \\
C_2 &= \neg x \lor y \\
C_4 &= \neg x \lor \neg y \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.
Example

\[ F = \{C_1, \ldots, C_6\} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[ C_2 = \neg x \lor y \quad C_3 = x \lor \neg y \quad C_5 = y \lor z \]

\[ C_4 = \neg x \lor \neg y \quad C_6 = y \lor \neg z \]

\( M \setminus \{C_3\} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{\neg x, y, z\} \), \( \text{Unsat}(M, \tau) = \{C_3\} \).

Flip \( x \) in \( \tau \): \( \tau' = \{x, y, z\} \), \( \text{Unsat}(M, \tau') = \{C_4\} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.

Flip \( y \) in \( \tau' \): \( \tau'' = \{x, \neg y, z\} \)
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.

Flip \( y \) in \( \tau' \): \( \tau'' = \{ x, \neg y, z \} \), \( \text{Unsat}(M, \tau'') = \{ C_2, C_6 \} \).
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.

Flip \( y \) in \( \tau' \): \( \tau'' = \{ x, \neg y, z \} \), \( \text{Unsat}(M, \tau'') = \{ C_2, C_6 \} \).

Tried all variables in \( C_4 \) — stop.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in SAT \), hence \( C_3 \) is necessary.  

SAT solver returns \( \tau = \{ \neg x, y, z \} \),  
\( Unsat(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \),  
\( Unsat(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.

Flip \( y \) in \( \tau' \): \( \tau'' = \{ x, \neg y, z \} \),  
\( Unsat(M, \tau'') = \{ C_2, C_6 \} \).

\( C_4 \) is necessary, \( \text{without SAT solver call} \).
Simple idea: when model rotation stops, backtrack to a necessary clause detected earlier and flip another variable.
Simple idea: when model rotation stops, backtrack to a necessary clause detected earlier and flip another variable.

**Fact:** let $\tau$ be a witness for $C$ in $F$, that is $\text{Unsat}(F, \tau) = \{C\}$. Then, the sets $\text{Unsat}(F, \tau|\neg x)$ for $x \in \text{Var}(C)$ are pairwise disjoint.

- By flipping different variables we are likely to detect new necessary clauses.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_1 &= x \lor y \\
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in SAT \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( Unsat(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( Unsat(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( x \) in \( \tau' \): back to \( \tau \). \( C_3 \) is already known to be necessary.

Flip \( y \) in \( \tau' \): \( \tau'' = \{ x, \neg y, z \} \), \( Unsat(M, \tau'') = \{ C_2, C_6 \} \).

Tried all variables in \( C_4 \) — stop, go back to \( C_3 \) and \( \tau \).
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z \\
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_3 &= x \lor \neg y \\
C_5 &= y \lor z \\
C_2 &= \neg x \lor y \\
C_4 &= \neg x \lor \neg y \\
C_6 &= y \lor \neg z
\end{align*}
\]

\[ M \setminus \{ C_3 \} \in \text{SAT}, \text{ hence } C_3 \text{ is necessary.} \]

\( \text{SAT solver returns } \tau = \{ \neg x, y, z \}, \text{ Unsat}(M, \tau) = \{ C_3 \}. \)

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \text{ is necessary.} \)

Flip \( y \) in \( \tau \): \( \tau' = \{ \neg x, \neg y, z \} \)
Example

\[
F = \{ C_1, \ldots, C_6 \}
\]

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( y \) in \( \tau \): \( \tau' = \{ \neg x, \neg y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_6 \} \rightarrow C_6 \) is necessary.
Example

\[ F = \{ C_1, \ldots, C_6 \} \]

\( M \) (the overapproximation of some MUS of \( F \)):

\[ C_2 = \neg x \lor y \]
\[ C_3 = x \lor \neg y \]
\[ C_4 = \neg x \lor \neg y \]
\[ C_5 = y \lor z \]
\[ C_6 = y \lor \neg z \]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( y \) in \( \tau \): \( \tau' = \{ \neg x, \neg y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_6 \} \rightarrow C_6 \) is necessary.

Flip \( z \) in \( \tau' \): \( \tau'' = \{ \neg x, \neg y, \neg z \} \)
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

\( \text{SAT solver returns } \tau = \{ \neg x, y, z \}, \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \}, \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( y \) in \( \tau \): \( \tau' = \{ \neg x, \neg y, z \}, \text{Unsat}(M, \tau') = \{ C_6 \} \rightarrow C_6 \) is necessary.

Flip \( z \) in \( \tau' \): \( \tau'' = \{ \neg x, \neg y, \neg z \}, \text{Unsat}(M, \tau'') = \{ C_5 \} \rightarrow C_5 \) is necessary.
Example

\( F = \{ C_1, \ldots, C_6 \} \)

\( M \) (the overapproximation of some MUS of \( F \)):

\[
\begin{align*}
C_2 &= \neg x \lor y \\
C_3 &= x \lor \neg y \\
C_4 &= \neg x \lor \neg y \\
C_5 &= y \lor z \\
C_6 &= y \lor \neg z
\end{align*}
\]

\( M \setminus \{ C_3 \} \in \text{SAT} \), hence \( C_3 \) is necessary.

SAT solver returns \( \tau = \{ \neg x, y, z \} \), \( \text{Unsat}(M, \tau) = \{ C_3 \} \).

Flip \( x \) in \( \tau \): \( \tau' = \{ x, y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_4 \} \rightarrow C_4 \) is necessary.

Flip \( y \) in \( \tau \): \( \tau' = \{ \neg x, \neg y, z \} \), \( \text{Unsat}(M, \tau') = \{ C_6 \} \rightarrow C_6 \) is necessary.

Flip \( z \) in \( \tau' \): \( \tau'' = \{ \neg x, \neg y, \neg z \} \), \( \text{Unsat}(M, \tau'') = \{ C_5 \} \rightarrow C_5 \) is necessary.

\( C_4, C_5, C_6 \) are necessary, without SAT solver call.
Recursive Model Rotation (RMR)

**Input:** $M$ — an over-approximation of an MUS
: $C$ — a clause necessary for $M$
: $\tau$ — a witness for $C$ (i.e. $\text{Unsat}(M, \tau) = \{C\}$)

```plaintext
foreach $x \in \text{Var}(C)$ do
    $\tau' \leftarrow \tau|_{\neg x}$ // flip $x$
    if $\text{Unsat}(M, \tau') = \{C'\}$ and $C'$ is not known to be necessary for $M$
        then
            mark $C'$ as necessary
            RMR($M, C', \tau'$)
```

Note: The second condition of if keeps the number of the recursive calls linear in the size of computed MUS.
Recursive Model Rotation (RMR)

**Input:** $M$ — an over-approximation of an MUS  
: $C$ — a clause necessary for $M$  
: $\tau$ — a witness for $C$ (i.e. $Unsat(M, \tau) = \{C\}$)

```
foreach $x \in Var(C)$ do
    $\tau' \leftarrow \tau|_{\neg x}$ // flip $x$
    if $Unsat(M, \tau') = \{C'\}$ and $C'$ is not known to be necessary for $M$
    then
        mark $C'$ as necessary
        RMR($M, C', \tau'$)
```

- The second condition of if keeps the number of the recursive calls linear in the size of computed MUS.
Recursive Model Rotation (RMR)

- 500 benchmarks submitted to MUS track of SAT Competition 2011.
- Time limit 1200 sec, memory limit 4 GB.

- Left: model rotation (x-axis) vs. RMR (y-axis), CPU time (sec).

A. Belov, J. Marques-Silva (UCD, Dublin)
Recursive Model Rotation (RMR)

- 500 benchmarks submitted to MUS track of SAT Competition 2011.
- Time limit 1200 sec, memory limit 4 GB.

- Left: model rotation (x-axis) vs. RMR (y-axis), CPU time (sec).
- Right: % of clauses in the computed MUS detected by RMR (red) and by (non-recursive) model rotation (blue).
MUSer2 — MUS extractor with RMR

- 295 benchmarks used in the MUS track of SAT Competition 2011.
- Time limit 1800 sec, memory limit 4 GB.
Summary

- Recursive Model Rotation (RMR) — simple but powerful technique for acceleration of MUS extraction.
- Clause reordering (see the paper) — gives a slight performance edge.
- MUSer2 — state-of-the-art MUS extractor
Summary

- Recursive Model Rotation (RMR) — simple but powerful technique for acceleration of MUS extraction.
- Clause reordering (see the paper) — gives a slight performance edge.
- MUSer2 — state-of-the-art MUS extractor

Thank you for your attention!
Impact of RMR

- 295 benchmarks used in the MUS track of SAT Competition 2011.
- Time limit 1800 sec, memory limit 4 GB.

- MUS computation without RMR (x-axis) vs with RMR (y-axis)
  - Left: number of SAT solver calls (instances solved in both cases).
Impact of RMR

- 295 benchmarks used in the MUS track of SAT Competition 2011.
- Time limit 1800 sec, memory limit 4 GB.

MUS computation without RMR ($x$-axis) vs with RMR ($y$-axis)
- Left: number of SAT solver calls (instances solved in both cases).
- Right: CPU time (sec).
Model Rotation [Marques-Silva&Lynce, SAT'11]

- 500 benchmarks submitted to MUS track of SAT Competition 2011.
- Time limit 1200 sec, memory limit 4 GB.

- Left: no model rotation (x-axis) vs. model rotation (y-axis).
- Right: % of clauses in computed MUS detected by model rotation.