On Computing Preferred MUSes and MCSes

J. Marques-Silva[1,2] and A. Previti[1]

[1]University College Dublin, Ireland

[2]IST/INESC-ID, Lisbon, Portugal

SAT Conference @ FLoC 2014

Vienna, Austria, July 2014

Recap MUSes

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

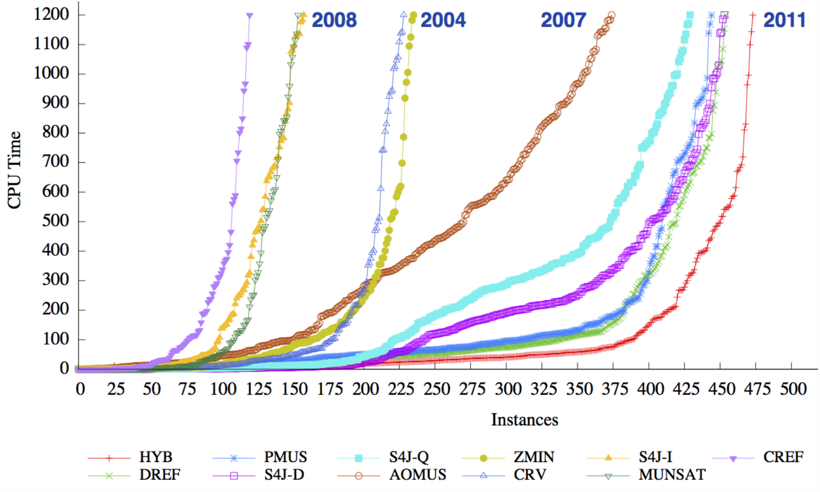
- Formula is unsatisfiable but **not** irreducible
- Can remove clauses, and formula still unsatisfiable
- **Minimal Unsatisfiable Subset** (**MUS**):
  - Irreducible subformula that is unsatisfiable
- Complexity results:
  - Decision problem: D^p-complete [PW88]
  - Function problem: in FP^{NP} with lower bound in $FP^{NP}_{||}$ [CT95]
- Many applications: abstraction in software verification; debugging declarative models; pinpointing in DLs; type error debugging; etc.

Recap MCSes

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

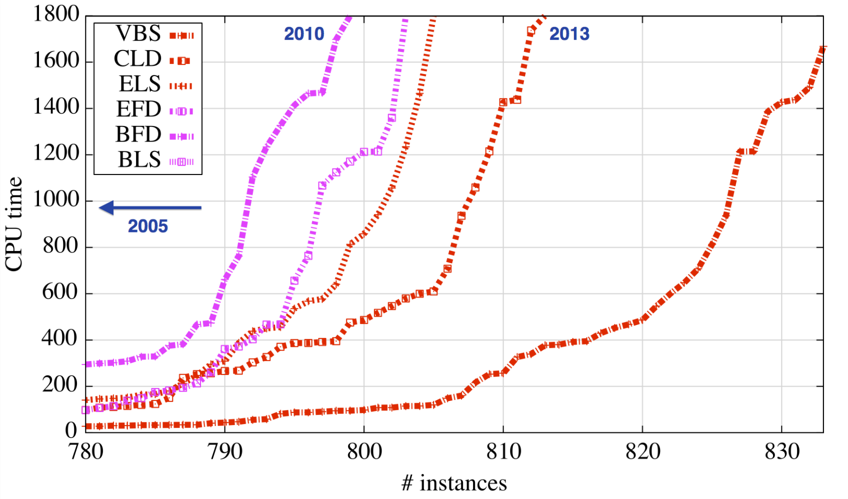
- Formula is unsatisfiable with satisfiable subformulas
- Can remove clauses such that remaining clauses are satisfiable
- **Minimal Correction Subset** (**MCS**):
  - Irreducible subformula such that the complement is satisfiable
  - An **MSS** is the complement of an MCS
- Complexity results:
  - Function problem: can be solved with $\mathcal{O}(\log n)$ calls to a SAT oracle
- Many applications: restore consistency; smallest MCSes are MaxSAT solutions; MUS enumeration; Minimal models; etc.

Recent improvements in MUS extraction



[MFM04,vMW08,GMP07,MSL11]

Recent improvements in MCS extraction



[BS05,FSMFT10,MSHJPB13]

How to compute preferred MUSes & MCSes?

- Preferred explanations in constraints [J01,J04]
  - Uses **non**-intuitive definition of preferred set. **Why**?
- Preferences in MUS/MCS/MSS extraction?
  - Clause selected using some ranking
- Complexity consequences?
  - Are there easy cases with preferences?
- Consequences for existing algorithms?
- Consequences for existing optimizations?
- Consequences significant in practice?

Outline

Outline

Preferred MUSes, MCSes & MSSes

Membership Problems

Complexity Results

Practical Impact

Preferred MUSes, MCSes & MSSes

- Precedence operator, $\prec$ [J01,J04]
  - Given $c_1, c_2 \in \mathcal{F}$, $c_1 \prec c_2$ denotes that c_1 is preferred to c_2
- Strict total order $<$ on set of clauses $\mathcal{F}$, $c_1 < c_2 < \ldots < c_m$

- Lexicographic preference, given $<$ on $\mathcal{F}$:
  - $\mathcal{A} \subseteq \mathcal{F}$ is lexicographically preferred to $\mathcal{B} \subseteq \mathcal{F}$, $\mathcal{A} <_{\text{lex}} \mathcal{B}$, iff

$$\exists_{1 \leq k \leq m},\ c_k \in \mathcal{A} \setminus \mathcal{B}\ \wedge\ \mathcal{A} \cap \{c_1, \ldots, c_{k-1}\} = \mathcal{B} \cap \{c_1, \ldots, c_{k-1}\}$$

- Anti-lexicographic preference, given $<$ on $\mathcal{F}$:
  - $\mathcal{A} \subseteq \mathcal{F}$ is anti-lexicographically preferred to $\mathcal{B} \subseteq \mathcal{F}$, $\mathcal{A} <_{\text{antilex}} \mathcal{B}$, iff

$$\exists_{1 \leq k \leq m},\ c_k \in \mathcal{B} \setminus \mathcal{A}\ \wedge\ \mathcal{A} \cap \{c_{k+1}, \ldots, c_m\} = \mathcal{B} \cap \{c_{k+1}, \ldots, c_m\}$$

- L-preferred/A-preferred MUSes/MCSes/MSSes:
  - MUS/MCS/MSS $\mathcal{A}$ is L-preferred (resp. A-preferred) if for all MUS/MCS/MSS $\mathcal{B} \neq \mathcal{A}$, $\mathcal{A} <_{\text{lex}} \mathcal{B}$ (resp. $\mathcal{A} <_{\text{antilex}} \mathcal{B}$)

Example – preferred MUSes

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\bar{x}_1 \vee \bar{x}_2)$	(x_1)	$(x_5 \vee x_6)$	$(\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$	(x_3)	(x_4)	(x_2)

Set	Type	L or A -preferred?
$\{c_1, c_2, c_7\}$	MUS	✓(L)
$\{c_2, c_4, c_5, c_6\}$	MUS	✓(A)

Example – preferred MCSes

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\bar{x}_1 \vee \bar{x}_2)$	(x_1)	$(x_5 \vee x_6)$	$(\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$	(x_3)	(x_4)	(x_2)

Set	Type	L or A -preferred?
$\{c_2\}$	MCS	✓(A)
$\{c_6, c_7\}$	MCS	✗
$\{c_5, c_7\}$	MCS	✗
$\{c_4, c_7\}$	MCS	✗
$\{c_1, c_6\}$	MCS	✗
$\{c_1, c_5\}$	MCS	✗
$\{c_1, c_4\}$	MCS	✓(L)

Example – preferred MSSes

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\bar{x}_1 \vee \bar{x}_2)$	(x_1)	$(x_5 \vee x_6)$	$(\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$	(x_3)	(x_4)	(x_2)

Set	Type	L or A -preferred?
$\{c_1, c_3, c_4, c_5, c_6, c_7\}$	MSS	✗
$\{c_1, c_2, c_3, c_4, c_5\}$	MSS	✓(L & A)
$\{c_1, c_2, c_3, c_4, c_6\}$	MSS	✗
$\{c_1, c_2, c_3, c_5, c_6\}$	MSS	✗
$\{c_2, c_3, c_4, c_5, c_7\}$	MSS	✗
$\{c_2, c_3, c_4, c_6, c_7\}$	MSS	✗
$\{c_2, c_3, c_5, c_6, c_7\}$	MSS	✗

Outline

Recap complexity classes

- Decision problems:
  - $P^{NP} = \Delta_2^p$: decision problems solvable in deterministic polynomial time using polynomial number of calls to an NP oracle
  - $NP^{NP} = \Sigma_2^p$: decision problems solvable in non-deterministic polynomial time using polynomial number of calls to an NP oracle

- Function problems:
  - FP: function problems solvable in deterministic polynomial time
  - FP^{NP}: function problems solvable in deterministic polynomial time using polynomial number of calls to an NP oracle
  - $FP^{\Sigma_2^p}$: function problems solvable in deterministic polynomial time using polynomial number of calls to a Σ_2^p oracle

MUS & MCS membership

- Given $c \in \mathcal{F}$, does there exist MUS of $\mathcal{F}$ containing c?
  - Problem well-known to be Σ_2^p-complete [L05]

- Given $c \in \mathcal{F}$, does there exist MCS of $\mathcal{F}$ containing c?
  - Claim: problem is Σ_2^p-complete
  - Membership:
    - Guess set $\mathcal{C}$ containing c
    - Check $\mathcal{C}$ irreducible
    - Check $\mathcal{F} \setminus \mathcal{C}$ satisfiable
  - Hardness:
    - Reduce MUS membership testing to MCS membership testing
    - MCSes are minimal hitting sets of MUSes and vice-versa [R87,BL03]
    - Thus, c is in some MUS iff c is in some MCS

MSS non-membership & generalizations

- Given $c \in \mathcal{F}$, does there exist MSS of $\mathcal{F}$ not containing c?
  - Claim: problem is Σ_2^p-complete
  - Membership:
    - Guess set $\mathcal{S}$ not containing c
    - Check $\mathcal{S}$ is satisfiable and maximal
  - Hardness:
    - Reduce MCS membership testing to MSS non-membership testing
    - Any MCS $\mathcal{C}$ is the complement of some MSS $\mathcal{S}$, $\mathcal{C} = \mathcal{F} \setminus \mathcal{S}$
    - c is in some MCS of $\mathcal{F}$ iff c is not in some MSS of $\mathcal{F}$

- Results can be extended to testing membership/non-membership of a set of target clauses $\mathcal{T} \subsetneq \mathcal{F}$

Outline

Preferred MUSes, MCSes & MSSes

Membership Problems

Complexity Results

Practical Impact

Overview of results

	Membership			Hardness		
	MUS	MCS	MSS	MUS	MCS	MSS
L-preferred	$FP^{\Sigma_2^p}$	$FP^{\Sigma_2^p}$	FP^{NP}	Σ_2^p	Σ_2^p	D^p
A-preferred	FP^{NP}	FP^{NP}	$FP^{\Sigma_2^p}$	D^p	D^p	Σ_2^p

- Given the hardness results:
  - Some preferred sets easier to compute than others
  - Explains earlier work on finding A-preferred explanations [J01,J04]

Selected hardness results

- Computing an L-preferred MUS/MCS is Σ_2^p-hard
  - Reduce MUS (resp. MCS) membership to computing an L-preferred MUS (resp. MCS)
    - Let $c \in \mathcal{F}$ be the target clause for MUS (resp. MCS) membership
    - Make c most preferred over all other clauses in $\mathcal{F}$
    - c is in an MUS (resp. MCS) iff the L-preferred MUS (resp. MCS) contains c

- Computing an A-preferred MSS is Σ_2^p-hard
  - Reduce MSS non-membership to computing an A-preferred MSS
    - Let $c \in \mathcal{F}$ be the target clause for MSS non-membership
    - Make c the least preferred clause over all clauses in $\mathcal{F}$
    - c is not in an MSS iff the A-preferred MSS does not contain c

Selected membership results

- Computing an L-preferred MUS/MCS is in $\text{FP}^{\Sigma_2^p}$
  - Describe a polynomial time algorithm that uses an oracle for Σ_2^p to construct a preferred MUS (resp. MCS)
    1. Construct set $\mathcal{S}$, initially set to $\emptyset$
    2. Analyze all clauses of $\mathcal{F}$, by decreasing preference given a total order on the clauses of $\mathcal{F}$
    3. For $c \in \mathcal{F}$ check whether $\mathcal{S} \cup \{c\}$ is in some MUS (resp. MCS)
       Note: Can use a Σ_2^p oracle
    4. If the outcome is true, then add c to $\mathcal{S}$
  - Algorithm executes a linear number of iterations, one for each clause of $\mathcal{F}$
    - A Σ_2^p oracle is invoked each time
  - Thus, computing an L-preferred MUS (resp. MCS) is in $\text{FP}^{\Sigma_2^p}$
- Computing an A-preferred MSS is in $\text{FP}^{\Sigma_2^p}$
  - Similar argument

Outline

Practical consequences

- In practical settings, with large formulas, and given current performance of QBF solvers:
  - Easier to compute A-preferred MUSes and MCSes
  - Easier to compute L-preferred MSSes

- Practical impact for computing A-preferred MUSes/MCSes & L-preferred MSSes?
  - Consequences on algorithms
  - Consequences on optimizations
  - Example of practical impact

Outline

Algorithms for MUSes, MCSes and MSSes

- Can formulate MUS/MCS extraction as computing a minimal set subject to a monotone predicate (MSMP) [MSJB13]
- MSMP algorithms:
  - Insertion (INS): $\mathcal{O}(m\,k)$ calls [PdS88]
  - Deletion (DEL): $\mathcal{O}(m)$ calls [CD91]
  - Dichotomic (Dicho): $\mathcal{O}(k \log m)$ calls [HLSB06]
  - QuickXplain (QXP): $\mathcal{O}(k + k \log(\frac{m}{k}))$ calls [J04]
  - Progression (PRG): $\mathcal{O}(k \log(1 + \frac{m}{k}))$ calls [MSJB13]
- MCS/MSS-specific algorithms:
  - Grow ($\approx$ DEL): $\mathcal{O}(m)$ calls [BS05]
  - ClauseD (CLD): $\mathcal{O}(m - k)$ calls [MSHJPB13]
  - MaxSAT (MxSAT): $\mathcal{O}(\log m)$ calls [P94]

Overview of results

	✗	✓, $\langle c_1, \ldots, c_m \rangle$	✓, $\langle c_m, \ldots, c_1 \rangle$
A-preferred MUS	–	INS, Dicho, QXP	DEL, PRG
A-preferred MCS	MxSAT, CLD	QXP	Grow, PRG
L-preferred MSS	MxSAT, CLD	Grow, PRG	QXP

Cannot compute A-preferred MCSes with CLD

- Recall CLD algorithm: [MSHJPB13]
  - Pick any assignment
  - Split $\mathcal{F}$
    - $\mathcal{S}$: satisfied clauses
    - $\mathcal{U}$: falsified clauses
  - Clauses in $\mathcal{S}$ must be satisfied
  - Create clause D with clauses in $\mathcal{U}$
  - While formula satisfiable
    - Satisfied clauses removed from D and added to $\mathcal{S}$

- By complementing initial assigment, can satisfy less preferred clauses, **but**
- Can satisfy clauses such that less preferred clause is added to MCS

Can compute A-preferred MUSes with Dicho

- Recall dichotomic algorithm on $\langle c_1, c_2, \ldots, c_m \rangle$ (simple version):
  - Working set $\mathcal{U} = \emptyset$
  - While $\mathcal{U}$ satisfiable
  - Use binary search to pick leftmost transition clause c
    - A transition clause is a necessary clause for any MUS given working set and other (preferred) clauses
  - Add c to $\mathcal{U}$

- Always picks most preferred transition clause
- Other less preferred clauses no longer relevant
- Other MUSes will include less preferred clauses

Outline

Optimizations used in MUS/MCS extraction

- Optimizations for MUS extraction:
  - Clause set refinement [BDTW93,MFM04,DHN06,MSL11]
  - (Recursive) model rotation [MSL11,BLMS12]
  - Clause redundancy condition [vMW08]

- Optimizations for MCS extraction:
  - Initial satisfied clauses [NEB12,MSHJPB13]
  - Disjoint unsatisfiable cores [MSHJPB13]
  - Backbone literals [MSHJPB13]

Deletion/addition safety

- Deletion-safe: clauses removed are not in preferred set
  - When relevant?
    - Clause set refinement
    - Disjoint unsatisfiable cores
    - Initial satisfied clauses (for MCSes)
    - Backbone literals ($\rightarrow$ satisfied clauses for MCSes)
- Addition-safe: classes added are in preferred set
  - When relevant?
    - (Recursive) model rotation
    - Initial satisfied clauses (for MSSes)
    - Backbone literals ($\rightarrow$ falsified clauses for MSSes)

Overview of results

- Focus on MUSes & MCSes:

	Optimization	Del/Add	Safe?
MUS	Clause set refinement	Deletion	no
	(Recursive) model rotation	Addition	yes
MCS	Disjoint unsatisfiable cores	Deletion	no
	Initial satisfied clauses	Deletion	no
	Backbone literals	Addition	yes
	Backbone literals	Deletion	yes

Analysis

- Clause set refinement is not deletion safe
  - SAT solver may return unsatisfiable core with less preferred clauses
  - Deleting clauses not in core not guaranteed to yield A-preferred MUS
  - Example: $c_1 < c_2 < c_3 < c_4 < c_5 < c_6$

c_1	c_2	c_3	c_4	c_5	c_6
$(x_1 \vee x_2)$	$(x_1 \vee \bar{x}_2)$	$(\bar{x}_1 \vee x_2)$	$(\bar{x}_1 \vee \bar{x}_2)$	(x_3)	$(\bar{x}_3)$

- Model rotation is addition safe
  - (Recursive) model rotation identifies necessary clauses for all MUSes that include already selected clauses
  - This includes the A-preferred MUS

Outline

Preferred MUSes, MCSes & MSSes

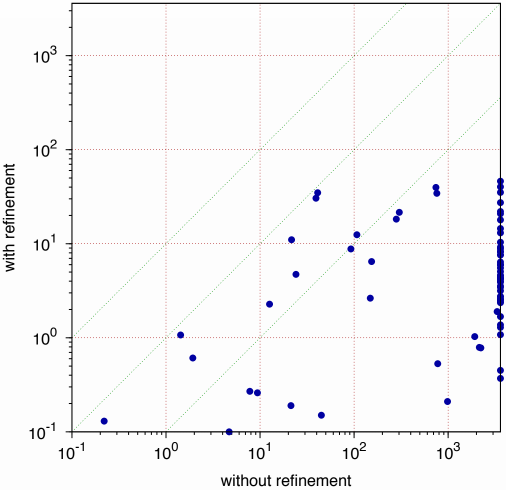
Membership Problems

Complexity Results

Practical Impact

- Algorithms
- Optimizations
- Preliminary Assessment

Example: MUS without clause set refinement



- Most recent release of MUSer2, w/ and w/o clause set refinement

Conclusions & Research Directions

- Computation of preferred MUSes/MCSes/MSSes
  - For "natural" notions of preferred sets, computation of MUSes and MCSes is hard for the second level of the polynomial hierarchy
- Computation of MUSes/MCSes given "less" natural definitions of preferred sets [J01,J04]
  - Some algorithms cannot be used
  - Several optimizations cannot be used
  - Practical impact significant
- Refine results in the paper
  - Membership & hardness bounds
  - Completeness results
- How to address negative results?
  - Are there possible compromises to make for A-preferred MUSes/MCSes?
  - Can L-preferred MUSes/MCSes be computed more efficiently for some classes of problems?

Thank You