MUSes and MCSes

George Katsirelos

February 13, 2016

SAT

- Boolean satisfiability in Conjunctive Normal Form (CNF)
  - True/False variables x, y, $z, \ldots$
  - Literals $x, \overline{x}, \ldots$
  - Clauses: disjunctions (sets) of literals $x \vee \overline{y} \vee z$
  - Formulas: conjunctions (sets) of clauses

Minimal Correction Sets

- $F \setminus C$ is satisfiable, no larger subset of F is

Minimal Correction Sets

- $F \setminus C$ is satisfiable, no larger subset of F is

$$
\begin{array}{ccc}
(x_1) & (x_2) & (x_3) \\
(\overline{x}_1 \vee \overline{x}_2) & (\overline{x}_1 \vee \overline{x}_3) & (\overline{x}_2 \vee \overline{x}_3)
\end{array}
$$

Minimal Unsatisfiable Sets

- $U \subseteq F$ is unsatisfiable, no smaller subset of F is

Also called minimal cores

Minimal Unsatisfiable Sets

- $U \subseteq F$ is unsatisfiable, no smaller subset of F is

Also called minimal cores

$$\begin{array}{ccc} (x_1) & (x_2) & (x_3) \\ (\overline{x}_1 \vee \overline{x}_2) & (\overline{x}_1 \vee \overline{x}_3) & (\overline{x}_2 \vee \overline{x}_3) \end{array}$$

MUS applications

- Analysis of over-constrained systems [Junker, 2004]
- Diagnosis [Han and Lee, 1999]
- Type error Debugging [Stuckey et al, 2003]
- Function Decomposition [Belov et al, 2012]
- Approximate model counting [Ivrii et al, 2016]
- MaxSAT [Morgado et al, 2013; Davies and Bacchus, 2011]

MUS applications – over-constrained systems

- Help user find the bug in their model

MUS applications – over-constrained systems

- Help user find the bug in their model
- Also for indirectly generated models
    - Compiler-generated model for type-checking

MUS applications – Approximate model counting

- Add a system of large XOR constraints and do complete model counting
- But SAT solvers generally poor at handling large XOR constraints

MUS applications – Approximate model counting

- Add a system of large XOR constraints and do complete model counting
- But SAT solvers generally poor at handling large XOR constraints
- Find “input” variables, restrict XORs to these

MUS applications – Approximate model counting

- Add a system of large XOR constraints and do complete model counting
- But SAT solvers generally poor at handling large XOR constraints
- Find "input" variables, restrict XORs to these
- MUSes of

$$F(\mathbf{X}) \wedge F(\mathbf{Y}) \wedge \bigwedge (x_i = y_i) \wedge \bigvee (x_i \neq y_i)$$

MCS applications

- MaxSAT [Mencía et al, 2015; Bjorner and Narodytska, 2015]
- Finding MUSes! [Bacchus and K, 2015]

Hitting set duality

$(x_1) \quad (x_2) \quad (x_3)$

$(\overline{x}_1 \vee \overline{x}_2) \quad (\overline{x}_1 \vee \overline{x}_3) \quad (\overline{x}_2 \vee \overline{x}_3)$

$(x_1) \quad (x_2) \quad (x_3)$

$(\overline{x}_1 \vee \overline{x}_2) \quad (\overline{x}_1 \vee \overline{x}_3) \quad (\overline{x}_2 \vee \overline{x}_3)$

$(x_1) \quad (x_2) \quad (x_3)$

$(\overline{x}_1 \vee \overline{x}_2) \quad (\overline{x}_1 \vee \overline{x}_3) \quad (\overline{x}_2 \vee \overline{x}_3)$

Hitting set duality

- Every (minimal) CS is a hitting set of all (minimal) USes
- Every (minimal) US is a hitting set of all (minimal) CSes

[Reiter, 1987]

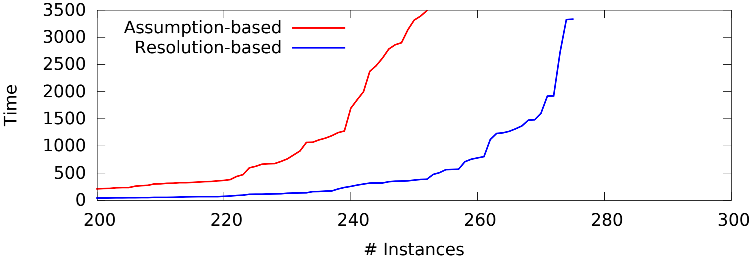
Interesting Queries

- Discover a single MCS/MCS
  - D^P-complete [Liberatore, 2005]
- Enumerate all MCSes/MUSes
  - Hypergraph transversal
- Find minimum MCS/MUS
  - MUS: Σ_2^P-complete [Gupta, 2006]
  - MCS: D^P-complete (MaxSAT)

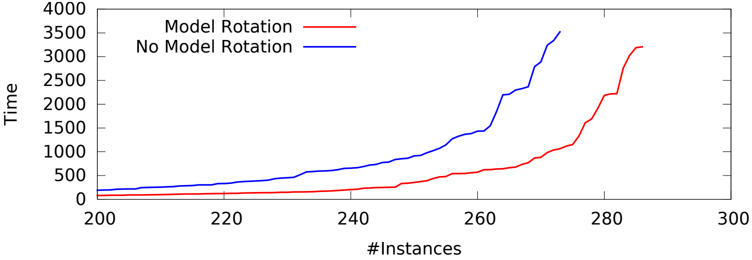
Discovering MUSes

- Algorithm
- Incremental solver technology
- Avoiding calling the SAT solver
  - Trimming [Belov et al, 2014]
  - Model Rotation [Belov and Marques-Silva, 2011]
  - Clause set refinement [Nadel et al, 2014]

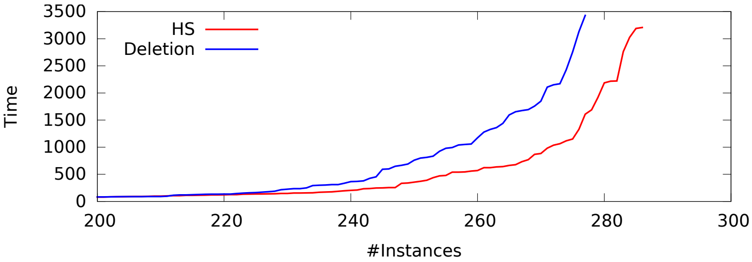
Impact of incremental solver technology



Impact of fewer SAT calls



Impact of algorithm



Algorithms

- Deletion-based [Belov and Marques-Silva, 2012]
- Addition-based [Gregoire et al, 2008; van Maaren and Wieringa, 2008]
- Dichotomic [Junker, 2004]
- Exploit duality [Bacchus and K, 2015]

Algorithms

- Deletion-based [Belov and Marques-Silva, 2012]
- Addition-based [Gregoire et al, 2008; van Maaren and Wieringa, 2008]
- Dichotomic [Junker, 2004]
- Exploit duality [Bacchus and K, 2015]

$\Rightarrow$ All applicable to any set minimization problem

Deletion based

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

Deletion based

$(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$

Deletion based

$(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ ✓ UNSAT

Deletion based

$(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$

Deletion based

$(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$

Deletion based

$(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$ ✓ UNSAT

Deletion based

$(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$

Deletion based

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Deletion based

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

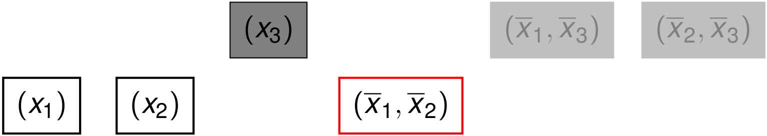
(x_1) (x_2) (x_3) ✗ SAT

Deletion based

$(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3) $(\overline{x}_1, \overline{x}_2)$

Deletion based



Deletion based

(x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) $(\overline{x}_1, \overline{x}_2)$ ✓ UNSAT

Deletion based

(x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) $(\overline{x}_1, \overline{x}_2)$

Deletion based

(x_2) (x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) $(\overline{x}_1, \overline{x}_2)$

Deletion based

(x_2) (x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) $(\overline{x}_1, \overline{x}_2)$ ✗ SAT

Deletion based

(x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) $(\overline{x}_1, \overline{x}_2)$

Deletion based

(x_1) (x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_2) $(\overline{x}_1, \overline{x}_2)$

Deletion based

(x_1) (x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_2) $(\overline{x}_1, \overline{x}_2)$ ✗ SAT

Deletion based

(x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) $(\overline{x}_1, \overline{x}_2)$

Deletion based

(x_3) $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) $(\overline{x}_1, \overline{x}_2)$

MUS!

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

Exploiting Hitting Set duality

$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

✗ No MCS

Exploiting Hitting Set duality

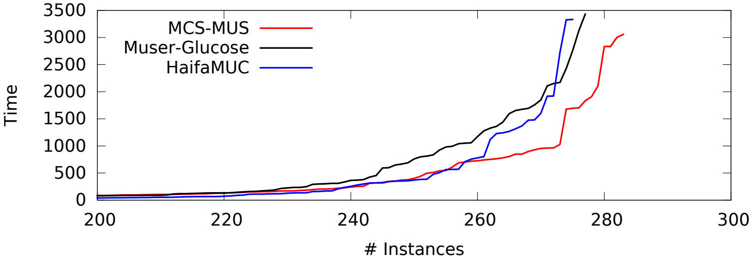
$(\overline{x}_1, \overline{x}_2)$ $(\overline{x}_1, \overline{x}_3)$ $(\overline{x}_2, \overline{x}_3)$

(x_1) (x_2) (x_3)

MUS

Is it faster to

- do m SAT checks?
  – or –
- discover k MCSes?


MCS-MUS
Muser-Glucose
HaifaMUC
Time
Instances
0
500
1000
1500
2000
2500
3000
3500
200
220
240
260
280
300

The way forward

- Faster SAT solving 😄
  - Better use of existing SAT solvers
  - Extract more information from the solver
- Improved MCS extraction
- A new algorithm?

More problems

- Enumeration [Liffiton and Sakallah, 2008; Liffiton et al, 2015; Bacchus and K, 2016]
- Minimization [Ignatiev et al, 2013;Ignatiev et al, 2015;Liffiton et al, 2009; Marques-Silva and Previti, 2014]
- MCS extraction and enumeration [O'Callaghan et al, 2005; Bailey and Stuckey, 2005; Felfernig et al, 2014; Nöhrer et al, 2012; Marques-Silva et al, 2013; Bacchus et al, 2014;Bacchus and K, 2015;Mencía et al, 2015]

Q?