# On Minimal Corrections in ASP

**Mikoláš Janota**[1] Joao Marques-Silva[2]

RCRA 2017, Bari

[1] INESC-ID/IST, University of Lisbon, Portugal
[2] LaSIGE, Faculty of Science, University of Lisbon, Portugal

# Context and History

- Number of interesting problems about propositional formulae:

# Context and History

- Number of interesting problems about propositional formulae:
- Minimally Unsatisfiable Set (MUS), diagnostics, debugging, SMT

# Context and History

- Number of interesting problems about propositional formulae:
- Minimally Unsatisfiable Set (MUS), diagnostics, debugging, SMT
- Minimal Correction Set (MCS), diagnostics, debugging

# Context and History

- Number of interesting problems about propositional formulae:
- Minimally Unsatisfiable Set (MUS), diagnostics, debugging, SMT
- Minimal Correction Set (MCS), diagnostics, debugging
- Prime Implicant/Implicate, model checking

# Context and History

- Number of interesting problems about propositional formulae:
- Minimally Unsatisfiable Set (MUS), diagnostics, debugging, SMT
- Minimal Correction Set (MCS), diagnostics, debugging
- Prime Implicant/Implicate, model checking
- Minimal model, circumscription

# Context and History

- Number of interesting problems about propositional formulae:
- Minimally Unsatisfiable Set (MUS), diagnostics, debugging, SMT
- Minimal Correction Set (MCS), diagnostics, debugging
- Prime Implicant/Implicate, model checking
- Minimal model, circumscription
- Backbone, fault-localization

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\big)$

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\big)$
  - MUS — subset minimum for the UNSAT predicate.

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow (\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi))$
  - $\phi \subseteq \psi \Rightarrow (\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi))$
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\big)$
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.
- $\mathcal{L}_1, \mathcal{L}_2$ sets of literals:

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \left(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\right)$
  - $\phi \subseteq \psi \Rightarrow \left(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\right)$
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.
- $\mathcal{L}_1, \mathcal{L}_2$ sets of literals:
  - $\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow \left(\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi\right)$

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\big)$
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.
- $\mathcal{L}_1, \mathcal{L}_2$ sets of literals:
  - $\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow \big(\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi\big)$
  - prime implicant — subset minimum for the $\cdot \models \varphi$ predicate.

# Monotone Predicates

- These problems are instances of monotone predicates. [Marques-Silva et al., 2013]
- Example for sets of clauses $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{SAT}(\psi) \Rightarrow \mathsf{SAT}(\phi)\big)$
  - $\phi \subseteq \psi \Rightarrow \big(\mathsf{UNSAT}(\phi) \Rightarrow \mathsf{UNSAT}(\psi)\big)$
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.
- $\mathcal{L}_1, \mathcal{L}_2$ sets of literals:
  - $\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow \big(\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi\big)$
  - prime implicant — subset minimum for the $\cdot \models \varphi$ predicate.
- Literal a backbone if $\phi \models l$
  $\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow \big(\bigwedge_{l \in \mathcal{L}_2} \phi \models l \Rightarrow \bigwedge_{l \in \mathcal{L}_1} \phi \models l\big)$

# What about ASP?

- Unlike propositional logic, ASP is not monotone.

# What about ASP?

- Unlike propositional logic, ASP is not monotone.
- How to define minimality?

## What about ASP?

- Unlike propositional logic, ASP is not monotone.
- How to define minimality?
- Can the algorithms from propositional logic be adapted?
  (or at least some)

# What about ASP?

- Unlike propositional logic, ASP is not monotone.
- How to define minimality?
- Can the algorithms from propositional logic be adapted? (or at least some)

**Example**

$$
\begin{array}{rcll}
 & \leftarrow & \mathit{not}\ \mathit{move}(a). & \texttt{\% program} \\
\mathit{move}(a) & \leftarrow & \mathit{stone}(b), \mathit{not}\ \mathit{stone}(c). & \texttt{\% program} \\
\mathit{stone}(c) & \leftarrow & . & \texttt{\% fact (input)}
\end{array}
$$

# What about ASP?

- Unlike propositional logic, ASP is not monotone.
- How to define minimality?
- Can the algorithms from propositional logic be adapted? (or at least some)

**Example**

$$\begin{array}{lcll} & \leftarrow & \mathit{not}\ \mathit{move}(a). & \texttt{\% program} \\ \mathit{move}(a) & \leftarrow & \mathit{stone}(b), \mathit{not}\ \mathit{stone}(c). & \texttt{\% program} \\ \mathit{stone}(c) & \leftarrow & . & \texttt{\% fact (input)} \end{array}$$

Possible fix: add $\mathit{stone}(b)$, remove $\mathit{stone}(c)$

# Maximal Consistent Set in ASP

## Definition

# Maximal Consistent Set in ASP

**Definition**

- Let $P$ be a consistent ASP program and $\mathcal{S}$ be a set of atoms.

# Maximal Consistent Set in ASP

## Definition

- Let $P$ be a consistent ASP program and $\mathcal{S}$ be a set of atoms.
- A set $\mathcal{L} \subseteq \mathcal{S}$ is a maximal consistent subset of $\mathcal{S}$ w.r.t. $P$

# Maximal Consistent Set in ASP

## Definition

- Let $P$ be a consistent ASP program and $\mathcal{S}$ be a set of atoms.
- A set $\mathcal{L} \subseteq \mathcal{S}$ is a maximal consistent subset of $\mathcal{S}$ w.r.t. $P$
  - if the program $P \cup \{s. \mid s \in \mathcal{L}\}$ is consistent

# Maximal Consistent Set in ASP

**Definition**

- Let $P$ be a consistent ASP program and $\mathcal{S}$ be a set of atoms.
- A set $\mathcal{L} \subseteq \mathcal{S}$ is a maximal consistent subset of $\mathcal{S}$ w.r.t. $P$
  - if the program $P \cup \{s. \mid s \in \mathcal{L}\}$ is consistent
  - and for any $\mathcal{L}'$, such that $\mathcal{L} \subsetneq \mathcal{L}' \subseteq \mathcal{S}$, the program $P \cup \{s. \mid s \in \mathcal{L}'\}$ is inconsistent.

**Observe:** *In monotone case $\mathcal{L}$ is maximally consistent iff $\mathcal{L} \cup \{s\}$ is inconsistent for any $s \in \mathcal{S} \setminus \mathcal{L}$. Does not hold in non-monotone.*

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.
- Let `choice`$(\mathcal{S})$ denote the choice rule $0 \leq \{s_1, \ldots, s_k\}$

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.
- Let `choice`$(\mathcal{S})$ denote the choice rule $0 \leq \{s_1, \ldots, s_k\}$
- Let `atleast1`$(\{s_1, \ldots, s_k\})$ denote the choice rule $1 \leq \{s_1, \ldots, s_k\}$.

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.
- Let $\texttt{choice}(\mathcal{S})$ denote the choice rule $0 \leq \{s_1, \ldots, s_k\}$
- Let $\texttt{atleast1}(\{s_1, \ldots, s_k\})$ denote the choice rule $1 \leq \{s_1, \ldots, s_k\}$.

## Idea

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.
- Let $\texttt{choice}(\mathcal{S})$ denote the choice rule $0 \leq \{s_1, \ldots, s_k\}$
- Let $\texttt{atleast1}(\{s_1, \ldots, s_k\})$ denote the choice rule $1 \leq \{s_1, \ldots, s_k\}$.

## Idea

- Define $P' = P \cup \{s. \mid s \in \mathcal{L}\} \cup \{\texttt{choice}(\mathcal{S} \setminus \mathcal{L}).\}$.

# Choice Rules

## Notation

- Consider set of atoms: $\mathcal{S} = \{s_1, \ldots, s_k\}$.
- Let $\texttt{choice}(\mathcal{S})$ denote the choice rule $0 \leq \{s_1, \ldots, s_k\}$
- Let $\texttt{atleast1}(\{s_1, \ldots, s_k\})$ denote the choice rule $1 \leq \{s_1, \ldots, s_k\}$.

## Idea

- Define $P' = P \cup \{s. \mid s \in \mathcal{L}\} \cup \{\texttt{choice}(\mathcal{S} \setminus \mathcal{L}).\}$.
- There exists a consistent set $\mathcal{L}'$ s.t. $\mathcal{L} \subseteq \mathcal{L}' \subseteq \mathcal{S}$ iff $P'$ has an answer set $\mu$ such that $\mathcal{L}' = \mathcal{S} \cap \mu$.

# Algorithm: At-least-1

$\mathcal{L} \leftarrow \emptyset$ // consistency lower bound

**while** true **do**

  $P' \leftarrow P \cup \{s. \mid s \in \mathcal{L}\}$

  $P' \leftarrow P' \cup \{\texttt{atleast1}(\mathcal{S} \setminus \mathcal{L}).\}$

  $(\text{res}, \mu) \leftarrow \texttt{solve}(P')$

  **if** $\neg$res **then return** $\mathcal{L}$

  $\mathcal{L} \leftarrow \mathcal{L} \cup (\mu \cap \mathcal{S})$

# Algorithm: Unit addition

$\mathcal{L} \leftarrow \emptyset$ // consistency lower bound
**while** $\mathcal{S} \neq \emptyset$ **do**
  $s_f \leftarrow$ pick an arbitrary element from $\mathcal{S}$
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_f\}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{s_f\}$
  $P' \leftarrow P' \cup \{s. \mid s \in \mathcal{L}\}$
  $P' \leftarrow P \cup \{\texttt{choice}(\mathcal{S}).\}$
  $(\text{res}, \mu) \leftarrow \texttt{solve}(P')$
  **if** $\neg\text{res}$ **then** $\mathcal{L} \leftarrow \mathcal{L} \setminus \{s_f\}$
  **else** $\mathcal{L} \leftarrow \mathcal{L} \cup (\mu \cap \mathcal{S})$
**return** $\mathcal{L}$

# Algorithm: Progression

$\mathcal{L} \leftarrow \emptyset$ `// consistency lower bound`
$K \leftarrow 1$ `// chunk size`
**while** $\mathcal{S} \neq \emptyset$ **do**
  $\mathcal{C} \leftarrow$ pick $\min(|\mathcal{S}|, K)$ arbitrary elements from $\mathcal{S}$
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{C}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{C}$
  $P' \leftarrow P' \cup \{s. \mid s \in \mathcal{L}\}$
  $P' \leftarrow P \cup \{\texttt{choice}(\mathcal{S}).\}$
  $(\text{res}, \mu) \leftarrow \texttt{solve}(P')$
  **if** $\neg\text{res}$ **then** `// re-analyze chunk more finely`
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \mathcal{C}$
    **if** $K > 1$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{C}$
    $K = 1$ `// reset chunk size`
  **else**
    $K \leftarrow 2K$ `// double chunk size`
    $\mathcal{L} \leftarrow \mathcal{L} \cup (\mu \cap \mathcal{S})$
**return** $\mathcal{L}$

# Minimal Correction Sets in ASP

## Definition

# Minimal Correction Sets in ASP

**Definition**

- $P$ be an inconsistent logic program

# Minimal Correction Sets in ASP

**Definition**

- $P$ be an inconsistent logic program
- $\mathcal{A}$ and $\mathcal{R}$ be sets of rules

# Minimal Correction Sets in ASP

**Definition**

- $P$ be an inconsistent logic program
- $\mathcal{A}$ and $\mathcal{R}$ be sets of rules
- An $(\mathcal{A}, \mathcal{R})$-correction of $P$ is a pair $(M_r, M_a)$ s.t.

# Minimal Correction Sets in ASP

**Definition**

- $P$ be an inconsistent logic program
- $\mathcal{A}$ and $\mathcal{R}$ be sets of rules
- An $(\mathcal{A}, \mathcal{R})$-correction of $P$ is a pair $(M_r, M_a)$ s.t.
  - $M_r \subseteq \mathcal{R}$ and $M_a \subseteq \mathcal{A}$ and the program

### Definition

- $P$ be an inconsistent logic program
- $\mathcal{A}$ and $\mathcal{R}$ be sets of rules
- An $(\mathcal{A}, \mathcal{R})$-correction of $P$ is a pair $(M_r, M_a)$ s.t.
    - $M_r \subseteq \mathcal{R}$ and $M_a \subseteq \mathcal{A}$ and the program
    - $(P \setminus M_r) \cup M_a$ is consistent.

## Definition

- $P$ be an inconsistent logic program
- $\mathcal{A}$ and $\mathcal{R}$ be sets of rules
- An $(\mathcal{A}, \mathcal{R})$-correction of $P$ is a pair $(M_r, M_a)$ s.t.
    - $M_r \subseteq \mathcal{R}$ and $M_a \subseteq \mathcal{A}$ and the program
    - $(P \setminus M_r) \cup M_a$ is consistent.
- An $(\mathcal{A}, \mathcal{R})$-correction $(M_r, M_a)$ is minimal if for any $(\mathcal{A}, \mathcal{R})$-correction $(M'_r, M'_a)$ such that $M'_r \subseteq M_r$ and $M'_a \subseteq M_a$, it holds that $M_a = M'_a$ and $M_r = M'_r$.

# Minimal Correction Sets and Maximal Consistency

To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s_r^{\mathrm{r}}$ for each $r \in \mathcal{R}$.

# Minimal Correction Sets and Maximal Consistency

To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s_r^{\mathrm{r}}$ for each $r \in \mathcal{R}$.
- Introduce fresh atoms $s_r^{\mathrm{a}}$ for each $r \in \mathcal{A}$.

# Minimal Correction Sets and Maximal Consistency

To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s_r^{\mathrm{r}}$ for each $r \in \mathcal{R}$.
- Introduce fresh atoms $s_r^{\mathrm{a}}$ for each $r \in \mathcal{A}$.
- Replace each rule $r \in \mathcal{R}$ with $\mathit{head}(r) \leftarrow s_r^{\mathrm{r}},\ \mathit{body}(r)$

# Minimal Correction Sets and Maximal Consistency

To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s_r^{\mathrm{r}}$ for each $r \in \mathcal{R}$.
- Introduce fresh atoms $s_r^{\mathrm{a}}$ for each $r \in \mathcal{A}$.
- Replace each rule $r \in \mathcal{R}$ with $\mathit{head}(r) \leftarrow s_r^{\mathrm{r}},\ \mathit{body}(r)$
- Replace each rule $r \in \mathcal{A}$ with $\mathit{head}(r) \leftarrow \mathit{not}\ s_r^{\mathrm{a}},\ \mathit{body}(r)$.

# Minimal Correction Sets and Maximal Consistency

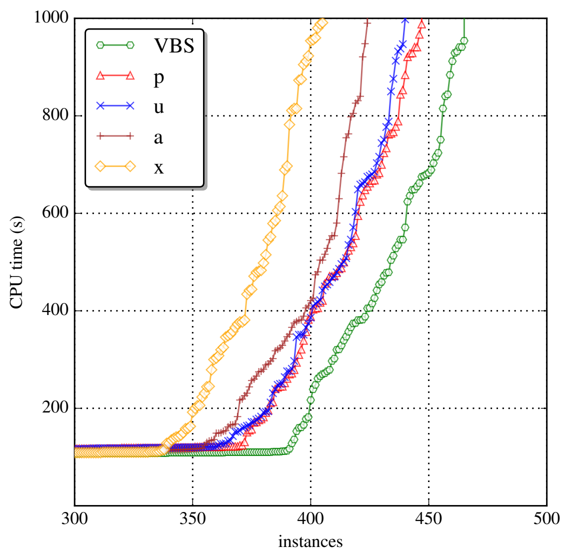
To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s_r^{\mathrm{r}}$ for each $r \in \mathcal{R}$.
- Introduce fresh atoms $s_r^{\mathrm{a}}$ for each $r \in \mathcal{A}$.
- Replace each rule $r \in \mathcal{R}$ with $\mathit{head}(r) \leftarrow s_r^{\mathrm{r}},\ \mathit{body}(r)$
- Replace each rule $r \in \mathcal{A}$ with $\mathit{head}(r) \leftarrow \mathit{not}\ s_r^{\mathrm{a}},\ \mathit{body}(r)$.
- Maximal consistent subset of the fresh atoms gives a minimal correction.

# Experimental Results

| Family | a | p | u | x | VBS |
|---|---|---|---|---|---|
| knight [8,10] (95) | 74 | 75 | **78** | 60 | 80 |
| knight [8,4] (51) | 7 | **13** | **13** | 7 | 14 |
| patterns [16,10] (100) | **100** | **100** | **100** | **100** | 100 |
| patterns [20,15] (100) | **100** | **100** | **100** | **100** | 100 |
| solitaire [12] (18) | **18** | **18** | **18** | 17 | 18 |
| solitaire [14] (16) | **12** | 9 | 11 | 4 | 13 |
| graceful graphs [10,50] (100) | 57 | **75** | 63 | 62 | 83 |
| graceful graphs [30,20] (57) | 56 | **57** | **57** | 55 | 57 |
| *total (537)* | 424 | **447** | 440 | 405 | 465 |

# Experimental Results (Cont.)



# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.
- Link between maximal consistency and corrections.

## Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.
- Link between maximal consistency and corrections.
- More experiments.

## Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.
- Link between maximal consistency and corrections.
- More experiments.
- More algorithms?

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.
- Link between maximal consistency and corrections.
- More experiments.
- More algorithms?
- How to obtain the “addition set”?

# Summary

- Many recent results on algorithms for propositional logic involving monotone predicates.
- What about ASP, which is not monotone?
- Idea: Using a choice rule let the solver choose one of the supersets.
- 3 different algorithms developed.
- Link between maximal consistency and corrections.
- More experiments.
- More algorithms?
- How to obtain the “addition set”?
- What are the good means for users to specify the addition and removal sets?

Thank You for Your Attention!

Questions?

Marques-Silva, J., Janota, M., and Belov, A. (2013).
**Minimal sets over monotone predicates in boolean formulae.**
In *Computer Aided Verification - International Conference (CAV)*, pages 592–607.