On Minimal Corrections in ASP

Mikoláš Janota\textsuperscript{1}    Joao Marques-Silva\textsuperscript{2}

RCRA 2017, Bari

\textsuperscript{1} INESC-ID/IST, University of Lisbon, Portugal
\textsuperscript{2} LaSIGE, Faculty of Science, University of Lisbon, Portugal
• Number of interesting problems about propositional formulae:
• 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
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 $\phi, \psi$

\[ \phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi)) \]
\[ \phi \subseteq \psi \Rightarrow (\text{UNSAT}(\phi) \Rightarrow \text{UNSAT}(\psi)) \]

MUS — subset minimum for the UNSAT predicate.

MSS — subset maximum for the SAT predicate.

\[ L_1, L_2 \text{ sets of literals:} \]
\[ L_1 \subseteq L_2 \Rightarrow (L_1 | \phi = \bigwedge l \in L_2 \phi \bigwedge l \in L_1 \phi) \]

Prime implicant — subset minimum for the $\phi =$ predicate.

Literal a backbone if $\phi | l = L_1 \subseteq L_2 \Rightarrow \bigwedge l \in L_2 \phi | l = \bigwedge l \in L_1 \phi$
Monotone Predicates

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

• Example for sets of clauses $\phi$, $\psi$
  • $\phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi))$
Monotone Predicates

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

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

- 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 $\phi, \psi$
  - $\phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi))$
  - $\phi \subseteq \psi \Rightarrow (\text{UNSAT}(\phi) \Rightarrow \text{UNSAT}(\psi))$

- MUS — subset minimum for the UNSAT predicate.
- MSS — subset maximum for the SAT predicate.

- $L_1, L_2$ sets of literals:
Monotone Predicates

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

• Example for sets of clauses \( \phi, \psi \)
  • \( \phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi)) \)
  • \( \phi \subseteq \psi \Rightarrow (\text{UNSAT}(\phi) \Rightarrow \text{UNSAT}(\psi)) \)

• MUS — subset minimum for the UNSAT predicate.
• MSS — subset maximum for the SAT predicate.

• \( \mathcal{L}_1, \mathcal{L}_2 \) sets of literals:
  • \( \mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow (\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi) \)
Monotone Predicates

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

- Example for sets of clauses \( \phi, \psi \)
  - \( \phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi)) \)
  - \( \phi \subseteq \psi \Rightarrow (\text{UNSAT}(\phi) \Rightarrow \text{UNSAT}(\psi)) \)
  - MUS — subset minimum for the UNSAT predicate.
  - MSS — subset maximum for the SAT predicate.

- \( \mathcal{L}_1, \mathcal{L}_2 \) sets of literals:
  - \( \mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow (\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi) \)
  - 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 $\phi$, $\psi$
  - $\phi \subseteq \psi \Rightarrow (\text{SAT}(\psi) \Rightarrow \text{SAT}(\phi))$
  - $\phi \subseteq \psi \Rightarrow (\text{UNSAT}(\phi) \Rightarrow \text{UNSAT}(\psi))$

• MUS — subset minimum for the UNSAT predicate.
• MSS — subset maximum for the SAT predicate.

• $\mathcal{L}_1, \mathcal{L}_2$ sets of literals:
  - $\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow (\mathcal{L}_1 \models \varphi \Rightarrow \mathcal{L}_2 \models \varphi)$
  - prime implicant — subset minimum for the $\cdot \models \varphi$ predicate.

• Literal a backbone if $\phi \models I$

$$\mathcal{L}_1 \subseteq \mathcal{L}_2 \Rightarrow (\bigwedge_{I \in \mathcal{L}_2} \phi \models I \Rightarrow \bigwedge_{I \in \mathcal{L}_1} \phi \models I)$$
What about ASP?

- Unlike propositional logic, ASP is not monotone.
What about ASP?

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

Example:

```
not move(a).

move(a) ← stone(b), not stone(c).

stone(c) ← .
```

% program

% fact (input)
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**

\[
\text{move}(a) \leftarrow \text{stone}(b), \neg \text{stone}(c). \quad \% \text{ program}
\]

\[
\text{not move}(a). \quad \% \text{ program}
\]

\[
\text{stone}(c) \leftarrow . \quad \% \text{ fact (input)}
\]
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**

\[
\begin{align*}
\text{not move}(a) & \quad \% \text{ program} \\
\text{move}(a) & \quad \text{stone}(b), \text{not stone}(c). \quad \% \text{ program} \\
\text{stone}(c) & \quad \% \text{ fact (input)}
\end{align*}
\]

Possible fix: add \text{stone}(b), remove \text{stone}(c)
Maximal Consistent Set in ASP

Definition

Let $P$ be a consistent ASP program and $S$ be a set of atoms.

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

Definition

• Let $P$ be a consistent ASP program and $S$ be a set of atoms.
Definition

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

**Definition**

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

Notation

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

Notation

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

Idea

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

Janota and Silva On Minimal Corrections in ASP 6 / 15
**Choice Rules**

**Notation**

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

Notation

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

Idea
Choice Rules

Notation

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

Idea

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

Janota and Silva On Minimal Corrections in ASP
Choice Rules

Notation

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

Idea

- Define \( P' = P \cup \{s. \mid s \in \mathcal{L}\} \cup \{\text{choice}(S \setminus \mathcal{L}).\} \).
- There exists a consistent set \( \mathcal{L}' \) s.t. \( \mathcal{L} \subseteq \mathcal{L}' \subseteq S \) iff \( P' \) has an answer set \( \mu \) such that \( \mathcal{L}' = S \cap \mu \).
Algorithm: At-least-1

1 \( \mathcal{L} \leftarrow \emptyset \) // consistency lower bound
2 while true do
3 \( P' \leftarrow P \cup \{s. \mid s \in \mathcal{L}\} \)
4 \( P' \leftarrow P' \cup \{\text{atleast1}(S \setminus \mathcal{L}).\} \)
5 \( (\text{res}, \mu) \leftarrow \text{solve}(P') \)
6 if \( \neg \text{res} \) then return \( \mathcal{L} \)
7 \( \mathcal{L} \leftarrow \mathcal{L} \cup (\mu \cap S) \)
Algorithm: **Unit addition**

1. $\mathcal{L} \leftarrow \emptyset$  \hspace{1cm} // consistency lower bound
2. while $\mathcal{S} \neq \emptyset$ do
3.     $s_f \leftarrow$ pick an arbitrary element from $\mathcal{S}$
4.     $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_f\}$
5.     $\mathcal{L} \leftarrow \mathcal{L} \cup \{s_f\}$
6.     $P' \leftarrow P' \cup \{s. \mid s \in \mathcal{L}\}$
7.     $P' \leftarrow P \cup \text{choice}(\mathcal{S}).$
8.     $(\text{res}, \mu) \leftarrow \text{solve}(P')$
9.     if $\neg \text{res}$ then $\mathcal{L} \leftarrow \mathcal{L} \setminus \{s_f\}$
10. else $\mathcal{L} \leftarrow \mathcal{L} \cup (\mu \cap \mathcal{S})$
11. return $\mathcal{L}$
Algorithm: Progression

1. $L \leftarrow \emptyset$  // consistency lower bound
2. $K \leftarrow 1$  // chunk size

while $S \neq \emptyset$ do

3. $C \leftarrow \text{pick min}(\lvert S \rvert, K)$ arbitrary elements from $S$
4. $S \leftarrow S \setminus C$
5. $L \leftarrow L \cup C$
6. $P' \leftarrow P' \cup \{s \mid s \in L\}$
7. $P' \leftarrow P \cup \{	ext{choice}(S)\}$
8. $(\text{res}, \mu) \leftarrow \text{solve}(P')$
9. if $\neg \text{res}$ then  // re-analyze chunk more finely
10.     $L \leftarrow L \setminus C$
11.     if $K > 1$ then $S \leftarrow S \cup C$
12.     $K = 1$  // reset chunk size

else

13.     $K \leftarrow 2K$  // double chunk size
14.     $L \leftarrow L \cup (\mu \cap S)$

17. return $L$
Definition

Let $P$ be an inconsistent logic program, $A$ and $R$ be sets of rules. An $(A, R)$-correction of $P$ is a pair $(M_r, M_a)$ such that:
- $M_r \subseteq R$ and $M_a \subseteq A$
- The program $(P \setminus M_r) \cup M_a$ is consistent.

An $(A, R)$-correction $(M_r, M_a)$ is minimal if for any $(A, R)$-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 $M_r = M'_r$. 
Definition

- $P$ be an inconsistent logic program
Definition

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

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

- An $(A, R)$-correction $(M'_r, M'_a)$ is minimal if for any $(A, R)$-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 $M_r = M'_r$. 
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 $M_r = M_r'$. 
Definition

- \( P \) be an inconsistent logic program
- \( A \) and \( R \) be sets of rules
- An \((A, R)\)-correction of \( P \) is a pair \((M_r, M_a)\) s.t.
  - \( M_r \subseteq R \) and \( M_a \subseteq A \) 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 \( M_r = M'_r \).
To calculate \((\mathcal{A}, \mathcal{R})\)-correction via Maximal Consistency:

- Introduce fresh atoms \(s^r_r\) for each \(r \in \mathcal{R}\).
To calculate $(\mathcal{A}, \mathcal{R})$-correction via Maximal Consistency:

- Introduce fresh atoms $s^r_r$ for each $r \in \mathcal{R}$.
- Introduce fresh atoms $s^a_r$ for each $r \in \mathcal{A}$.
To calculate \((\mathcal{A}, \mathcal{R})\)-correction via Maximal Consistency:

- Introduce fresh atoms \(s^r_r\) for each \(r \in \mathcal{R}\).
- Introduce fresh atoms \(s^a_r\) for each \(r \in \mathcal{A}\).
- Replace each rule \(r \in \mathcal{R}\) with \(\text{head}(r) \leftarrow s^r_r, \text{body}(r)\)
To calculate \((\mathcal{A}, \mathcal{R})\)-correction via Maximal Consistency:

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

Maximal consistent subset of the fresh atoms gives a minimal correction.
To calculate \((\mathcal{A}, \mathcal{R})\)-correction via Maximal Consistency:

- Introduce fresh atoms \(s^r_r\) for each \(r \in \mathcal{R}\).
- Introduce fresh atoms \(s^a_r\) for each \(r \in \mathcal{A}\).
- Replace each rule \(r \in \mathcal{R}\) with \(\text{head}(r) \Leftarrow s^r_r, \text{body}(r)\).
- Replace each rule \(r \in \mathcal{A}\) with \(\text{head}(r) \Leftarrow \text{not } s^a_r, \text{body}(r)\).
- Maximal consistent subset of the fresh atoms gives a minimal correction.
## Experimental Results

<table>
<thead>
<tr>
<th>Family</th>
<th>a</th>
<th>p</th>
<th>u</th>
<th>x</th>
<th>VBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>knight [8, 10] (95)</td>
<td>74</td>
<td>75</td>
<td>78</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>knight [8, 4] (51)</td>
<td>7</td>
<td>13</td>
<td>13</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>patterns [16, 10] (100)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>patterns [20, 15] (100)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>solitaire [12] (18)</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>solitaire [14] (16)</td>
<td>12</td>
<td>9</td>
<td>11</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>graceful graphs [10, 50] (100)</td>
<td>57</td>
<td>75</td>
<td>63</td>
<td>62</td>
<td>83</td>
</tr>
<tr>
<td>graceful graphs [30, 20] (57)</td>
<td>56</td>
<td>57</td>
<td>57</td>
<td>55</td>
<td>57</td>
</tr>
<tr>
<td>total (537)</td>
<td>424</td>
<td>447</td>
<td>440</td>
<td>405</td>
<td>465</td>
</tr>
</tbody>
</table>
Experimental Results (Cont.)

![Graph showing CPU time (s) vs instances for different algorithms]

- VBS
- p
- u
- a
- x

Janota and Silva On Minimal Corrections in ASP
• 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?

- 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?
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.
• 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.
• 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”? 

Janota and Silva  
On Minimal Corrections in ASP
• 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?