Conflict-driven ASP Solving: Overview

1. Motivation
2. Preliminaries
3. Boolean constraints
4. Nogoods from logic programs
Outline

1 Motivation
2 Preliminaries
3 Boolean constraints
4 Nogoods from logic programs
Motivation of Conflict-driven ASP Solving

- **Goal** Approach to computing stable models of logic programs, based on concepts from
  - Constraint Processing (CP) and
  - Satisfiability Testing (SAT)
- **Idea** View inferences in ASP as unit propagation on nogoods
- **Benefits:**
  - A uniform constraint-based framework for different kinds of inferences in ASP
  - Advanced techniques from the areas of CP and SAT
  - Highly competitive implementation
1 Motivation
2 Preliminaries
3 Boolean constraints
4 Nogoods from logic programs
Partial interpretations
or: 3-valued interpretations

A partial interpretation maps atoms onto truth values \textit{true}, \textit{false},
and \textit{unknown}
Partial interpretations
or: 3-valued interpretations

A partial interpretation maps atoms onto truth values \textit{true}, \textit{false}, and \textit{unknown}

- **Representation:** \(\langle T, F \rangle\), where
  - \(T\) is the set of all \textit{true} atoms and
  - \(F\) is the set of all \textit{false} atoms
  - Truth of atoms in \(A \setminus (T \cup F)\) is \textit{unknown}
Partial interpretations
or: 3-valued interpretations

A partial interpretation maps atoms onto truth values true, false, and unknown

- **Representation:** \( \langle T, F \rangle \), where
  - \( T \) is the set of all true atoms and
  - \( F \) is the set of all false atoms
  - Truth of atoms in \( A \setminus (T \cup F) \) is unknown

- **Properties:**
  - \( \langle T, F \rangle \) is conflicting if \( T \cap F \neq \emptyset \)
  - \( \langle T, F \rangle \) is total if \( T \cup F = A \) and \( T \cap F = \emptyset \)
Partial interpretations
or: 3-valued interpretations

A partial interpretation maps atoms onto truth values \textit{true}, \textit{false}, and \textit{unknown}

- **Representation:** \langle T, F \rangle, where
  - \( T \) is the set of all \textit{true} atoms and
  - \( F \) is the set of all \textit{false} atoms
  - Truth of atoms in \( A \setminus (T \cup F) \) is \textit{unknown}

- **Properties:**
  - \langle T, F \rangle \text{ is conflicting} if \( T \cap F \neq \emptyset \)
  - \langle T, F \rangle \text{ is total} if \( T \cup F = A \) and \( T \cap F = \emptyset \)

- **Definition:** For \langle T_1, F_1 \rangle \text{ and } \langle T_2, F_2 \rangle, define
  - \langle T_1, F_1 \rangle \sqsubseteq \langle T_2, F_2 \rangle \text{ iff } T_1 \subseteq T_2 \text{ and } F_1 \subseteq F_2
  - \langle T_1, F_1 \rangle \sqcup \langle T_2, F_2 \rangle = \langle T_1 \cup T_2, F_1 \cup F_2 \rangle
Outline

1 Motivation

2 Preliminaries
   - Partial Interpretations
   - Unfounded Sets

3 Boolean constraints

4 Nogoods from logic programs
   - Nogoods from program completion
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.

- A set $U \subseteq \text{atom}(P)$ is an **unfounded set** of $P$ wrt $\langle T, F \rangle$.
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$.
  
  Intuitively, $\langle T, F \rangle$ is what we already know about $P$. 

  - Rules satisfying Condition 1 are not usable for further derivations.
  - Condition 2 is the unfounded set condition treating cyclic derivations: All rules still being usable to derive an atom in $U$ require an(other) atom in $U$ to be true.
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$

- if we have for each rule $r \in P$ such that $\text{head}(r) \in U$

• Rules satisfying Condition 1 are not usable for further derivations
• Condition 2 is the unfounded set condition treating cyclic derivations: All rules still being usable to derive an atom in $U$ require an(other) atom in $U$ to be true
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$

- if we have for each rule $r \in P$ such that $\text{head}(r) \in U$ either
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$.

- If we have for each rule $r \in P$ such that $\text{head}(r) \in U$ either
  1. $\text{body}(r)^+ \cap F \neq \emptyset$ or $\text{body}(r)^- \cap T \neq \emptyset$.

Intuitively, $\langle T, F \rangle$ is what we already know about $P$.

Rules satisfying Condition 1 are not usable for further derivations.

Condition 2 is the unfounded set condition treating cyclic derivations: All rules still being usable to derive an atom in $U$ require an(other) atom in $U$ to be true.
Unfounded sets

Let $P$ be a normal logic program, and let $(T, F)$ be a partial interpretation

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $(T, F)$

- if we have for each rule $r \in P$ such that $\text{head}(r) \in U$
  
  either
  
  1. $\text{body}(r)^+ \cap F \neq \emptyset$ or $\text{body}(r)^- \cap T \neq \emptyset$
  
  2. $\text{body}(r)^+ \cap U \neq \emptyset$
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$.

- If we have for each rule $r \in P$ such that $\text{head}(r) \in U$ either
  1. $\text{body}(r)^+ \cap F \neq \emptyset$ or $\text{body}(r)^- \cap T \neq \emptyset$ or
  2. $\text{body}(r)^+ \cap U \neq \emptyset$

- Rules satisfying Condition 1 are not usable for further derivations.
Unfounded sets

Let $P$ be a normal logic program, and let $\langle T, F \rangle$ be a partial interpretation.

- A set $U \subseteq \text{atom}(P)$ is an unfounded set of $P$ wrt $\langle T, F \rangle$.

- if we have for each rule $r \in P$ such that $\text{head}(r) \in U$ either
  
  1. $\text{body}(r)^+ \cap F \neq \emptyset$ or $\text{body}(r)^- \cap T \neq \emptyset$ or
  2. $\text{body}(r)^+ \cap U \neq \emptyset$

- Rules satisfying Condition 1 are not usable for further derivations.
- Condition 2 is the unfounded set condition treating cyclic derivations: All rules still being usable to derive an atom in $U$ require an(other) atom in $U$ to be true.
Example

\[ P = \left\{ \begin{array}{c c}
  a & \leftrightarrow & b \\
  b & \leftrightarrow & a
\end{array} \right\} \]
Example

\[ P = \{ \begin{array}{c} a \leftrightarrow b \\ b \leftrightarrow a \end{array} \} \]

- \( \emptyset \) is an unfounded set (by definition)
Example

\[ P = \{ a \leftrightarrow b, b \leftrightarrow a \} \]

- \( \emptyset \) is an unfounded set (by definition)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \{b\}, \emptyset \rangle \)
- \( \{a, b\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a, b\} \) is an unfounded set of \( P \) wrt any partial interpretation
Example

\[ P = \left\{ \begin{array}{c} a \\ b \end{array} \right\} \left\{ \begin{array}{c} b \\ a \end{array} \right\} \]

- \emptyset is an unfounded set (by definition)
- \{a\} is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \{a\} is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \{a\} is an unfounded set of \( P \) wrt any partial interpretation
Example

\( P = \left\{ \begin{array}{c c}
   a & \leftrightarrow b \\
   b & \leftrightarrow a
\end{array} \right\} \)

- \( \emptyset \) is an unfounded set (by definition)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \{b\}, \emptyset \rangle \)
Example

\[ P = \{ \begin{array}{c|c|c} a & b \\ \hline b & a \end{array} \} \]

- \emptyset is an unfounded set (by definition)
- \{a\} is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \{a\} is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \{a\} is not an unfounded set of \( P \) wrt \( \langle \{b\}, \emptyset \rangle \)
- Analogously for \{b\}
Example

\[ P = \{ a \leftarrow b, b \leftarrow a \} \]

- \( \emptyset \) is an unfounded set (by definition)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \{b\}, \emptyset \rangle \)
- \( \{a, b\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
Example

\[ P = \begin{cases} 
  a & \leftrightarrow b \\
  b & \leftrightarrow a 
\end{cases} \]

- \( \emptyset \) is an unfounded set (by definition)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \{b\} \rangle \)
- \( \{a\} \) is not an unfounded set of \( P \) wrt \( \langle \{b\}, \emptyset \rangle \)

- \( \{a, b\} \) is an unfounded set of \( P \) wrt \( \langle \emptyset, \emptyset \rangle \)
- \( \{a, b\} \) is an unfounded set of \( P \) wrt any partial interpretation
Assignments

- An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence
  $$(\sigma_1, \ldots, \sigma_n)$$
  of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$

- $T_v$ expresses that $v$ is true and $F_v$ that it is false
Assignments

- An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence $(\sigma_1, \ldots, \sigma_n)$ of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$
  - $T_v$ expresses that $v$ is true and $F_v$ that it is false
  - The complement, $\overline{\sigma}$, of a literal $\sigma$ is defined as $\overline{T_v} = F_v$ and $\overline{F_v} = T_v$
Assignments

• An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence

$$\left(\sigma_1, \ldots, \sigma_n\right)$$

of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$

• $T_v$ expresses that $v$ is true and $F_v$ that it is false

• The complement, $\bar{\sigma}$, of a literal $\sigma$ is defined as $\overline{T_v} = F_v$ and $\overline{F_v} = T_v$

• $A \circ \sigma$ stands for the result of appending $\sigma$ to $A$
Assignments

• An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence

$$(\sigma_1, \ldots, \sigma_n)$$

of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$

• $T_v$ expresses that $v$ is **true** and $F_v$ that it is **false**

• The complement, $\overline{\sigma}$, of a literal $\sigma$ is defined as $\overline{T_v} = F_v$ and $\overline{F_v} = T_v$

• $A \circ \sigma$ stands for the result of appending $\sigma$ to $A$

• Given $A = (\sigma_1, \ldots, \sigma_{k-1}, \sigma_k, \ldots, \sigma_n)$, we let $A[\sigma_k] = (\sigma_1, \ldots, \sigma_{k-1})$
Assignments

• An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence

$$(\sigma_1, \ldots, \sigma_n)$$

of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$

• $T_v$ expresses that $v$ is true and $F_v$ that it is false

• The complement, $\bar{\sigma}$, of a literal $\sigma$ is defined as $\bar{T_v} = F_v$ and $\bar{F_v} = T_v$

• $A \circ \sigma$ stands for the result of appending $\sigma$ to $A$

• Given $A = (\sigma_1, \ldots, \sigma_{k-1}, \sigma_k, \ldots, \sigma_n)$, we let $A[\sigma_k] = (\sigma_1, \ldots, \sigma_{k-1})$

• We sometimes identify an assignment with the set of its literals
Assignments

- An assignment $A$ over $\text{dom}(A) = \text{atom}(P) \cup \text{body}(P)$ is a sequence
  $$(\sigma_1, \ldots, \sigma_n)$$
  of signed literals $\sigma_i$ of form $T_v$ or $F_v$ for $v \in \text{dom}(A)$ and $1 \leq i \leq n$

- $T_v$ expresses that $v$ is true and $F_v$ that it is false

- The complement, $\overline{\sigma}$, of a literal $\sigma$ is defined as $\overline{T_v} = F_v$ and $\overline{F_v} = T_v$

- $A \circ \sigma$ stands for the result of appending $\sigma$ to $A$

- Given $A = (\sigma_1, \ldots, \sigma_{k-1}, \sigma_k, \ldots, \sigma_n)$, we let $A[\sigma_k] = (\sigma_1, \ldots, \sigma_{k-1})$

- We sometimes identify an assignment with the set of its literals

- Given this, we access true and false propositions in $A$ via

  $$A^T = \{v \in \text{dom}(A) \mid T_v \in A\} \quad \text{and} \quad A^F = \{v \in \text{dom}(A) \mid F_v \in A\}$$
Nogoods, solutions, and unit propagation

- A nogood is a set \( \{\sigma_1, \ldots, \sigma_n\} \) of signed literals, expressing a constraint violated by any assignment containing \( \sigma_1, \ldots, \sigma_n \).
A nogood is a set \( \{\sigma_1, \ldots, \sigma_n\} \) of signed literals, expressing a constraint violated by any assignment containing \( \sigma_1, \ldots, \sigma_n \).

An assignment \( A \) such that \( A^T \cup A^F = \text{dom}(A) \) and \( A^T \cap A^F = \emptyset \) is a solution for a set \( \Delta \) of nogoods, if \( \delta \not\subseteq A \) for all \( \delta \in \Delta \).
Nogoods, solutions, and unit propagation

- A nogood is a set \( \{\sigma_1, \ldots, \sigma_n\} \) of signed literals, expressing a constraint violated by any assignment containing \( \sigma_1, \ldots, \sigma_n \).

- An assignment \( A \) such that \( A^T \cup A^F = \text{dom}(A) \) and \( A^T \cap A^F = \emptyset \) is a solution for a set \( \Delta \) of nogoods, if \( \delta \nsubseteq A \) for all \( \delta \in \Delta \).

- For a nogood \( \delta \), a literal \( \sigma \in \delta \), and an assignment \( A \), we say that \( \sigma \) is unit-resulting for \( \delta \) wrt \( A \), if

  1. \( \delta \setminus A = \{\sigma\} \), and
  2. \( \bar{\sigma} \notin A \).
Nogoods, solutions, and unit propagation

- A nogood is a set \( \{ \sigma_1, \ldots, \sigma_n \} \) of signed literals, expressing a constraint violated by any assignment containing \( \sigma_1, \ldots, \sigma_n \).

- An assignment \( A \) such that \( A^T \cup A^F = \text{dom}(A) \) and \( A^T \cap A^F = \emptyset \) is a solution for a set \( \Delta \) of nogoods, if \( \delta \not\subseteq A \) for all \( \delta \in \Delta \).

- For a nogood \( \delta \), a literal \( \sigma \in \delta \), and an assignment \( A \), we say that \( \sigma \) is unit-resulting for \( \delta \) wrt \( A \), if:
  1. \( \delta \setminus A = \{ \sigma \} \) and
  2. \( \overline{\sigma} \not\in A \).

- For a set \( \Delta \) of nogoods and an assignment \( A \), unit propagation is the iterated process of extending \( A \) with unit-resulting literals until no further literal is unit-resulting for any nogood in \( \Delta \).
Outline

1. Motivation
2. Preliminaries
3. Boolean constraints
4. Nogoods from logic programs
The completion of a logic program $P$ can be defined as follows:

\[
\{ v_B \leftrightarrow a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \mid
\begin{aligned}
& B \in \text{body}(P), B = \{a_1, \ldots, a_m, \text{not } a_{m+1}, \ldots, \text{not } a_n \} \\
\cup & \{ a \leftrightarrow v_{B_1} \lor \cdots \lor v_{B_k} \mid a \in \text{atom}(P), \text{body}(a) = \{B_1, \ldots, B_k\} \}
\end{aligned}
\]

where $\text{body}(a) = \{ \text{body}(r) \mid r \in P, \text{head}(r) = a \}$
The (body-oriented) equivalence

\[ v_B \leftrightarrow a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \]

can be decomposed into two implications:
Nogoods from logic programs via program completion

- The (body-oriented) equivalence

\[ v_B \leftrightarrow a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \]

can be decomposed into two implications:

1. \[ v_B \rightarrow a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \]

is equivalent to the conjunction of

\[ \neg v_B \lor a_1, \ldots, \neg v_B \lor a_m, \neg v_B \lor \neg a_{m+1}, \ldots, \neg v_B \lor \neg a_n \]

and induces the set of nogoods

\[ \Delta(B) = \{ \{TB, Fa_1\}, \ldots, \{TB, Fa_m\}, \{TB, Ta_{m+1}\}, \ldots, \{TB, Ta_n\} \} \]
The (body-oriented) equivalence

\[ v_B \iff a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \]

can be decomposed into two implications:

\[ a_1 \land \cdots \land a_m \land \neg a_{m+1} \land \cdots \land \neg a_n \rightarrow v_B \]

gives rise to the nogood

\[ \delta(B) = \{ F_B, T_1, \ldots, T_m, F_{m+1}, \ldots, F_n \} \]
Analogously, the (atom-oriented) equivalence

\[ a \leftrightarrow v_{B_1} \lor \cdots \lor v_{B_k} \]

yields the nogoods

1. \[ \Delta(a) = \{ \{F_a, T_{B_1}\}, \ldots, \{F_a, T_{B_k}\} \} \] and

2. \[ \delta(a) = \{T_a, F_{B_1}, \ldots, F_{B_k}\} \]
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{T_a, FB_1, \ldots, FB_k\} \text{ and } \{\{F_a, TB_1\}, \ldots, \{F_a, TB_k\}\}$$
Nogoods from logic programs
atom-oriented nogoods

• For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get
  
  $\{Ta, FB_1, \ldots, FB_k\}$ and $\{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$

• Example Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

  \[
  \begin{align*}
  x & \leftarrow y \\
  x & \leftarrow \text{not } z
  \end{align*}
  \]

  $\{Tx, F\{y\}, F\{\text{not } z\}\}$

  $\{\{Fx, T\{y\}\}, \{Fx, T\{\text{not } z\}\}\}$
atom-oriented nogoods

- For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get
  \[
  \{T_a, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{F a, TB_1\}, \ldots, \{F a, TB_k\}\}
  \]

- **Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain
  \[
  \begin{array}{c}
  x \leftarrow y \\
  x \leftarrow \text{not } z
  \end{array}
  \]
  \[
  \{T x, F\{y\}, F\{\text{not } z\}\} \quad \{\{F x, T\{y\}\}, \{F x, T\{\text{not } z\}\}\}
  \]
  
  For nogood $\{T x, F\{y\}, F\{\text{not } z\}\}$, the signed literal
Nogoods from logic programs
atom-oriented nogoods

• For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$
\{ Ta, FB_1, \ldots, FB_k \} \quad \text{and} \quad \{ \{ Fa, TB_1 \}, \ldots, \{ Fa, TB_k \} \}
$$

• Example Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \leftarrow y$</td>
<td>${ Tx, F{y}, F{\text{not } z}}$</td>
</tr>
<tr>
<td>$x \leftarrow \text{not } z$</td>
<td>${ {Fx, T{y}}, {Fx, T{\text{not } z}} }$</td>
</tr>
</tbody>
</table>

For nogood $\{ Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal
Nogoods from logic programs  
atom-oriented nogoods

- For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get
  $$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

- **Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

  $x \leftarrow y$
  $x \leftarrow \text{not } z$

  $$\{Tx, F\{y\}, F\{\text{not } z\}\} \quad \{\{Fx, T\{y\}\}, \{Fx, T\{\text{not } z\}\}\}$$

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal
- $Fx$ is unit-resulting wrt assignment $(F\{y\}, F\{\text{not } z\})$ and
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

**Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

$$x \leftarrow y \quad \{Tx, F\{y\}, F\{\text{not } z\}\}$$
$$x \leftarrow \text{not } z \quad \{\{Fx, T\{y\}\}, \{Fx, T\{\text{not } z\}\}\}$$

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal

- $Fx$ is unit-resulting wrt assignment $(F\{y\}, F\{\text{not } z\})$ and
Nogoods from logic programs
atom-oriented nogoods

• For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get
  \[
  \{T_a, F_{B_1}, \ldots, F_{B_k}\} \quad \text{and} \quad \{\{F_a, T_{B_1}\}, \ldots, \{F_a, T_{B_k}\}\}
  \]

• Example Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain
  \[
  \begin{array}{c}
  x \leftarrow y \\
  x \leftarrow \text{not } z
  \end{array}
  \begin{array}{c}
  \{T_x, F\{y\}, F\{\text{not } z\}\} \\
  \{\{F_x, T\{y\}\}, \{F_x, T\{\text{not } z\}\}\}
  \end{array}
  \]

For nogood $\{T_x, F\{y\}, F\{\text{not } z\}\}$, the signed literal
  – $F_x$ is unit-resulting wrt assignment $(F\{y\}, F\{\text{not } z\})$ and
Nogoods from logic programs
atom-oriented nogoods

• For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

• Example Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

$$x \leftarrow y \quad \{Tx, F\{y\}, F\{\text{not } z\}\}$$
$$x \leftarrow \text{not } z \quad \{\{Fx, T\{y\}\}, \{Fx, T\{\text{not } z\}\}\}$$

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal

– $Fx$ is unit-resulting wrt assignment $(F\{y\}, F\{\text{not } z\})$ and
Nogoods from logic programs
atom-oriented nogoods

- For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

- Example Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

\[
\begin{array}{c}
\text{x} \leftarrow y \\
\text{x} \leftarrow \text{not } z
\end{array}
\]

\[
\{Tx, F\{y\}, F\{\text{not } z\}\} \\
\{Fx, T\{y\}, \{Fx, T\{\text{not } z\}\}\}
\]

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal
- $Fx$ is unit-resulting wrt assignment $(F\{y\}, F\{\text{not } z\})$ and
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

**Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

<table>
<thead>
<tr>
<th>$x \leftarrow y$</th>
<th>${Tx, F{y}, F{\text{not } z}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \leftarrow \text{not } z$</td>
<td>${{Fx, T{y}}, {Fx, T{\text{not } z}}}$</td>
</tr>
</tbody>
</table>

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal

- $T\{\text{not } z\}$ is unit-resulting wrt assignment $(Tx, F\{y\})$
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{Fa, TB_1\}, \ldots, \{Fa, TB_k\}\}$$

**Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

<table>
<thead>
<tr>
<th>$x$</th>
<th>$\leftarrow$</th>
<th>$y$</th>
<th>${Tx, F{y}, F{\text{not } z}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$\leftarrow$</td>
<td>$\text{not } z$</td>
<td>${{Fx, T{y}}, {Fx, T{\text{not } z}}}$</td>
</tr>
</tbody>
</table>

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal

- $T\{\text{not } z\}$ is unit-resulting wrt assignment $(Tx, F\{y\})$
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{T_a, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{\{F_a, TB_1\}, \ldots, \{F_a, TB_k\}\}$$

**Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not } z\}\}$, we obtain

$$x \leftarrow y \quad \quad \{Tx, F\{y\}, F\{\text{not } z\}\}$$

$$x \leftarrow \text{not } z \quad \quad \{\{Fx, T\{y\}\}, \{Fx, T\{\text{not } z\}\}\}$$

For nogood $\{Tx, F\{y\}, F\{\text{not } z\}\}$, the signed literal

$$T\{\text{not } z\}$$

is unit-resulting wrt assignment $(Tx, F\{y\})$
For an atom $a$ where $\text{body}(a) = \{B_1, \ldots, B_k\}$, we get

$$\{Ta, FB_1, \ldots, FB_k\} \quad \text{and} \quad \{ \{Fa, TB_1\}, \ldots, \{Fa, TB_k\} \}$$

**Example** Given Atom $x$ with $\text{body}(x) = \{\{y\}, \{\text{not} \ z\}\}$, we obtain

\[
\begin{array}{ll}
x & \leftarrow \ y \\
x & \leftarrow \ \text{not} \ z
\end{array}
\]

\[
\{ Tx, F\{y\}, F\{\text{not} \ z\} \} \quad \{ Fx, T\{y\} \}, \{ Fx, T\{\text{not} \ z\} \} \}
\]

For nogood $\{Tx, F\{y\}, F\{\text{not} \ z\}\}$, the signed literal

- $T\{\text{not} \ z\}$ is unit-resulting wrt assignment $(Tx, F\{y\})$
Nogoods from logic programs
atom-oriented nogoods

- For an atom \( a \) where \( \text{body}(a) = \{B_1, \ldots, B_k\} \), we get

\[
\{T a, F B_1, \ldots, F B_k\} \quad \text{and} \quad \{\{F a, T B_1\}, \ldots, \{F a, T B_k\}\}
\]

- Example Given Atom \( x \) with \( \text{body}(x) = \{\{y\}, \{\text{not } z\}\} \), we obtain

\[
\begin{align*}
x & \leftarrow y & \{T x, F\{y\}, F\{\text{not } z\}\} \\
x & \leftarrow \text{not } z & \{\{F x, T\{y\}\}, \{F x, T\{\text{not } z\}\}\}
\end{align*}
\]

For nogood \( \{T x, F\{y\}, F\{\text{not } z\}\} \), the signed literal

- \( T\{\text{not } z\} \) is unit-resulting wrt assignment \((T x, F\{y\})\)
For a body $B = \{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\}$, we get

$$\{FB, Ta_1, \ldots, Ta_m, Fa_{m+1}, \ldots, Fa_n\}$$

$$\{\{TB, Fa_1\}, \ldots, \{TB, Fa_m\}, \{TB, Ta_{m+1}\}, \ldots, \{TB, Ta_n\}\}$$
• For a body $B = \{a_1, \ldots, a_m, \text{not } a_{m+1}, \ldots, \text{not } a_n\}$, we get

$$\{FB, Ta_1, \ldots, Ta_m, Fa_{m+1}, \ldots, Fa_n\}$$

$$\{\{TB, Fa_1\}, \ldots, \{TB, Fa_m\}, \{TB, Ta_{m+1}\}, \ldots, \{TB, Ta_n\}\}$$

• Example Given Body $\{x, \text{not } y\}$, we obtain

$$\ldots \leftarrow x, \text{not } y$$

$$\ldots \leftarrow x, \text{not } y$$

$$\{F\{x, \text{not } y\}, Tx, Fy\}$$

$$\{\{T\{x, \text{not } y\}, Fx\}, \{T\{x, \text{not } y\}, Ty\}\}$$
Nogoods from logic programs
body-oriented nogoods

- For a body \( B = \{a_1, \ldots, a_m, \text{not } a_{m+1}, \ldots, \text{not } a_n\} \), we get

  \[
  \{FB, Ta_1, \ldots, Ta_m, Fa_{m+1}, \ldots, Fa_n\} \\
  \{TB, Fa_1\}, \ldots, \{TB, Fa_m\}, \{TB, Ta_{m+1}\}, \ldots, \{TB, Ta_n\} \}
  \]

- **Example** Given Body \( \{x, \text{not } y\} \), we obtain

  \[
  \ldots \leftarrow x, \text{not } y \\
  \vdots \\
  \ldots \leftarrow x, \text{not } y \\
  \{F\{x, \text{not } y\}, Tx, Fy\} \\
  \{\{T\{x, \text{not } y\},Fx\}, \{T\{x, \text{not } y\},Ty\}\}
  \]

  For nogood \( \delta(\{x, \text{not } y\}) = \{F\{x, \text{not } y\}, Tx, Fy\} \), the signed literal
  - \( T\{x, \text{not } y\} \) is unit-resulting wrt assignment \((Tx, Fy)\) and
  - \( Ty \) is unit-resulting wrt assignment \((F\{x, \text{not } y\}, Tx)\)
Characterization of stable models
for tight logic programs, i.e. free of positive recursion

Let $P$ be a logic program and

$$
\Delta_P = \{ \delta(a) \mid a \in \text{atom}(P) \} \cup \{ \delta \in \Delta(a) \mid a \in \text{atom}(P) \} \\
\cup \{ \delta(B) \mid B \in \text{body}(P) \} \cup \{ \delta \in \Delta(B) \mid B \in \text{body}(P) \}
$$
Characterization of stable models

for tight logic programs, ie. free of positive recursion

Let $P$ be a logic program and

$$
\Delta_P = \{ \delta(a) \mid a \in \text{atom}(P) \} \cup \{ \delta \in \Delta(a) \mid a \in \text{atom}(P) \} \\
\cup \{ \delta(B) \mid B \in \text{body}(P) \} \cup \{ \delta \in \Delta(B) \mid B \in \text{body}(P) \}
$$

**Theorem**

Let $P$ be a tight logic program. Then,

$X \subseteq \text{atom}(P)$ is a stable model of $P$ iff

$X = A_T \cap \text{atom}(P)$ for a (unique) solution $A$ for $\Delta_P$
Summary

• Partial assignments
• Unfounded sets
• Unit resulting literals
• Unit propagation
• Nogoods via program completion
• Characterization of stable models of tight programs in terms of nogoods.
References

Martin Gebser, Benjamin Kaufmann Roland Kaminski, and Torsten Schaub.
*Answer Set Solving in Practice.*
Synthesis Lectures on Artificial Intelligence and Machine Learning.
doi=10.2200/S00457ED1V01Y201211AIM019.

- **See also:** [http://potassco.sourceforge.net](http://potassco.sourceforge.net)