ASP Propagation

Torsten Grote

Universität Potsdam - Institut für Informatik

May 10, 2007
Anwserset Computation

\[
\text{answerset}(T, F, \Pi) \\
\text{begin} \\
A = \text{atom}(\Pi) \\
(T, F) \leftarrow \text{propagation}(T, F, \Pi) \\
\text{if } (T \cap F) \neq \emptyset \text{ then fail} \\
\text{if } (T \cup F) = A \text{ then return}(T) \\
A \leftarrow \text{select}(A \setminus (T \cup F), \Pi) \\
\text{answerset}(T \cup \{A\}, F) \\
\text{answerset}(T, F \cup \{A\}) \\
\text{end}
\]
Before 2006

- no transparent approach
- publications deal only with implementation details
- lack of formal framework for characterizing ASP computation
- very difficult to
  - understand the functioning of solvers
  - compare different solvers
  - reveal their shortcomings
  - come up with improvements
- SAT approach doesn’t have this deficiency
After 2006
- two proposals by Gebser and Schaub
- view answer set computations as derivations in an inference system
- don't look so much at logical equivalences in terms of models

First Proposal
- characterizing ASP inferences by unit propagation based on concepts from Constraint Processing
- nogoods already used in SAT solvers, can be used in ASP, too

Second Proposal
- using Tableau Calculi as inference system
- easy to understand concept known from proof theory
- declarative and fine-grained instrument for ASP operations and strategies
Tableau Calculi

General properties

- A tableau calculus is a set of tableau rules.
- Tableaux in most cases a binary tree.
- Root is a formula which is to proof.
- It is extended to branches only by application of rules.
- A branch is contradictory if it contains $a$ and $\neg a$ for an $a \in \text{atoms}$.
- A formula is invalid if every branch of its tableaux is contradictory.

Rules for classical logic

$$(\wedge) \frac{a \wedge b}{a} \quad (\vee) \frac{a \vee b}{a \mid b} \quad (\Rightarrow) \frac{a \Rightarrow b}{\neg a \mid b} \quad (\neg 1) \frac{a}{\neg \neg a} \quad (\neg 2) \frac{\neg \neg a}{a}$$
Properties of ASP tableaux

- root consists of all rules of logic program $\Pi$ and all atoms
- nodes are signed propositions preceded by $T$ for true and $F$ for false
- branches are extended by special ASP tableau rules
- branch is complete if it is contradictory or closed under all rules
- every branch corresponds to a pair $(\Pi, A)$ where $A$ is an assignment
- a non contradictory complete branch contains an answer set
- an entire tableaux represents traversal of the search space
### Atom-oriented inferences

<table>
<thead>
<tr>
<th>Rule Type</th>
<th>Rule Expression</th>
</tr>
</thead>
</table>
| **Atom-oriented tableau rules 1** | \[
\begin{align*}
(FTA) & \quad p \leftarrow l_1, \ldots, l_n \\
&T_{\{l_1, \ldots, l_n\}} \\
&\quad T_p
\end{align*}
\quad \Rightarrow \quad \begin{align*}
(BFA) & \quad p \leftarrow l_1, \ldots, l_n \\
&F_p \\
&\quad F_{\{l_1, \ldots, l_n\}}
\end{align*}
\] |

| **Atom-oriented nogoods 1** | \[
\Delta(p) = \{\{F_p, T_{\beta_1}\}, \ldots, \{F_p, T_{\beta_m}\}\}
\quad \text{body}(p) = \{\beta_1, \ldots, \beta_m\}
\] |

| **Atom-oriented tableau rules 2** | \[
\begin{align*}
(FFA) & \quad F_{\beta_1}, \ldots, F_{\beta_m} \\
&\quad F_p
\end{align*}
\quad \Rightarrow \quad \begin{align*}
(BTA) & \quad F_{\beta_1}, \ldots, F_{\beta_{i-1}}, F_{\beta_{i+1}}, \ldots, F_{\beta_m} \\
&\quad T_{\beta_i}
\end{align*}
\] |

| **Atom-oriented nogoods 2** | \[
\delta(p) = \{T_p, F_{\beta_1}, \ldots, F_{\beta_m}\}
\] |
Body-oriented tableau rules 1

\[
\begin{align*}
\text{(FTB)} & \
\frac{p \leftarrow l_1, \ldots, l_n}{T_1, \ldots, T_n} & \frac{F\{l_1, \ldots, l_i, \ldots, l_n\}}{T\{l_1, \ldots, l_n\}} \\
\text{(BFB)} & \
\frac{T_1, \ldots, T_i, \ldots, T_n}{T\{l_1, \ldots, l_n\}} & \frac{F_i}{F\{l_1, \ldots, l_n\}}
\end{align*}
\]

Body-oriented nogoods 1

\[
\delta(p) = \{F_\beta, T_1, \ldots, T_n\} \quad (\neg a \in \beta) \land (a = \bot) \Rightarrow T_l_a
\]

Body-oriented tableau rules 2

\[
\begin{align*}
\text{(FFB)} & \
\frac{p \leftarrow l_1, \ldots, l_i, \ldots, l_n}{F_i} & \frac{F\{l_1, \ldots, l_i, \ldots, l_n\}}{T\{l_1, \ldots, l_n\}} \\
\text{(BTB)} & \
\frac{T\{l_1, \ldots, l_i, \ldots, l_n\}}{T_i}
\end{align*}
\]

Body-oriented nogoods 2

\[
\Delta(p) = \{\{T_\beta, F_1\}, \ldots, \{T_\beta, F_n\}\} \quad (\neg a \in \beta) \land (a = \top) \Rightarrow F_l_a
\]
The Cut rule

\[(Cut[X]) \quad T_a | F_a \quad (a \in X \subseteq atom(\Pi) \cup body(\Pi))\]

Clark’s Completion

- unit propagation on a program’s completion is described by the previous rules
- \(T_{comp} = \{FTA, BFA, FFA, BTA, FTB, BFB, FFB, BTB\}\)

Fitting’s operator

- \(\Phi\) applies only forward propagation and can be viewed as a set of tableau rules, too
- \(T_\phi = \{FTA, FFA, FTB, FFB\}\)
Well-founded and Loop rules

Well-founded rules

(WFN) \( \frac{F_{\beta_1}, \ldots, F_{\beta_m}}{F_p} \)  \( (WFJ) \frac{F_{\beta_1}, \ldots, F_{\beta_{i-1}}, F_{\beta_{i+1}}, \ldots, F_{\beta_m}}{T_{\beta_i}} \)

\((p \in GUS(\{r \in \Pi | body(r) \notin \{\beta_1, \ldots, \beta_m\}\}))\)

Loop rules

(FL) \( \frac{F_{\beta_1}, \ldots, F_{\beta_m}}{F_p} \)  \( (BL) \frac{F_{\beta_1}, \ldots, F_{\beta_{i-1}}, F_{\beta_{i+1}}, \ldots, F_{\beta_m}}{T_{\beta_i}} \)

\((p \in L, L \in Loop(\Pi), EB(L) = \{\beta_1, \ldots, \beta_m\})\)
Well-founded operator

- \( \Omega \) is like \( \Phi \) except that it replaces negation of single atoms with negation of unfounded sets
- \( \mathcal{T}_\Omega = \{FTA, WFN, FTB, FFB\} \)

Existing ASP-Solvers

\[
\begin{align*}
\mathcal{T}_{\text{smmodels}} &= \mathcal{T}_{\text{comp}} \cup \{WFN, \text{Cut}[\text{atom}(\Pi)]\} \\
\mathcal{T}_{\text{noMoRe}} &= \mathcal{T}_{\text{comp}} \cup \{WFN, \text{Cut}[\text{body}(\Pi)]\} \\
\mathcal{T}_{\text{nomore++}} &= \mathcal{T}_{\text{comp}} \cup \{WFN, \text{Cut}[\text{atom}(\Pi) \cup \text{body}(\Pi)]\} \\
\mathcal{T}_{\text{cmodels}–1} &= \mathcal{T}_{\text{comp}} \cup \{\text{Cut}[\text{atom}(\Pi) \cup \text{body}(\Pi)]\} \\
\mathcal{T}_{\text{assat}} &= \mathcal{T}_{\text{comp}} \cup \{FL, \text{Cut}[\text{atom}(\Pi) \cup \text{body}(\Pi)]\}
\end{align*}
\]
Questions?