Improved Answer-Set Programming Encodings for Abstract Argumentation

Sarah Gaggl [1] Norbert Manthey [1] Alessandro Ronca [2]
Johannes Wallner [3] Stefan Woltran [4]

[1]Technische Universität Dresden, Germany

[2]La Sapienza, University of Rome, Italy

[3]HIIT, University of Helsinki, Finland

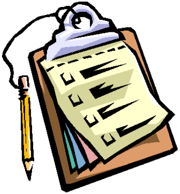
[4]Vienna University of Technology, Austria

Motivation

- Efficient solvers for abstract argumentation are an important development
- Reductions to answer set programming (ASP) are well-suited (enumeration of all solutions)
- For high-complexity semantics saturation technique is required
- Complex and tricky loop-techniques are hard to follow and potentially lead to performance bottlenecks
- We provide new and simpler encodings for preferred, stage and semi-stable semantics
- Based on alternative characterization and conditional literals in disjunction

Outline

1. Background
   - Abstract Argumentation
   - Syntax and Semantics (admissible, preferred)
2. ASP Encodings of AFs
   - Original Saturation Encodings
3. New Approach
   - New Characterization of Preferred Semantics
   - New ASP Encodings
4. Evaluation
5. Conclusion

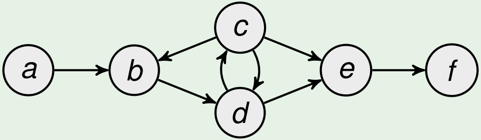


Argumentation Framework

Abstract Argumentation Framework [Dung95]

An abstract argumentation framework (AF) is a pair $F = (A, R)$, where A is a finite set of arguments and $R \subseteq A \times A$. Then $(a, b) \in R$ if a attacks b. Argument $a \in A$ is defended by $S \subseteq A$ (in F) iff, for each $b \in A$ with $(b, a) \in R$, S attacks b.

Example



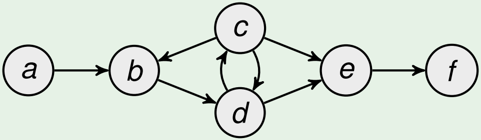
Semantics

Semantics for AFs

Let $F = (A, R)$ and $S \subseteq A$, we say S is conflict-free in F, i.e. $S \in cf(F)$, if $\forall a, b \in S$: $(a, b) \notin R$. Then, $S \in cf(F)$ is

- admissible in F, i.e. $S \in adm(F)$, if each $a \in S$ is defended by S;
- a preferred extension (of F), i.e. $S \in pref(F)$, if $S \in adm(F)$ and for each $T \in adm(F)$, $S \not\subset T$.

Example



$adm(F) = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{a, d\}, \{c, f\}, \{a, c, f\}, \{a, d, f\}\}$, and
$pref(F) = \{\{a, c, f\},\{a, d, f\}\}$

ASP Encodings

Admissible Sets

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F.

Encoding

$$\widehat{F} = \{\mathrm{arg}(a) \mid a \in A\} \cup \{\mathrm{att}(a,b) \mid (a,b) \in R\}$$

$$\pi_{adm} = \left\{ \begin{array}{lcl} \mathrm{in}(X) & \leftarrow & \mathit{not}\ \mathrm{out}(X), \mathrm{arg}(X) \\ \mathrm{out}(X) & \leftarrow & \mathit{not}\ \mathrm{in}(X), \mathrm{arg}(X) \\ & \leftarrow & \mathrm{in}(X), \mathrm{in}(Y), \mathrm{att}(X,Y) \\ \mathrm{defeated}(X) & \leftarrow & \mathrm{in}(Y), \mathrm{att}(Y,X) \\ & \leftarrow & \mathrm{in}(X), \mathrm{att}(Y,X), \mathit{not}\ \mathrm{defeated}(Y) \end{array} \right\}$$

Result: For each AF F, $adm(F) \equiv \mathcal{AS}(\pi_{adm}(\widehat{F}))$

Saturation Encodings

Preferred Extension

Given an AF (A, R). A set $S \subseteq A$ is preferred in F, if S is admissible in F and for each $T \subseteq A$ admissible in T, $S \not\subset T$.

Encoding

$$
\pi_{saturate} = \left\{ \begin{array}{lll}
\mathrm{inN}(X) | \mathrm{outN}(X) & \leftarrow & \mathrm{out}(X) \\
\mathrm{inN}(X) & \leftarrow & \mathrm{in}(X) \\
\mathrm{spoil} & \leftarrow & \mathrm{eq} \\
\mathrm{spoil} & \leftarrow & \mathrm{inN}(X), \mathrm{inN}(Y), \mathrm{att}(X, Y) \\
\mathrm{spoil} & \leftarrow & \mathrm{inN}(X), \mathrm{outN}(Y), \mathrm{att}(Y, X), \\
 & & \mathrm{undefeated}(Y) \\
\mathrm{inN}(X) & \leftarrow & \mathrm{spoil}, \mathrm{arg}(X) \\
\mathrm{outN}(X) & \leftarrow & \mathrm{spoil}, \mathrm{arg}(X) \\
 & \leftarrow & \mathit{not}\ \mathrm{spoil}
\end{array} \right\}
$$

$$
\pi_{pref} = \pi_{adm} \cup \pi_{helpers} \cup \pi_{saturate}
$$

Result: For each AF F, $\mathit{pref}(F) \equiv \mathcal{AS}(\pi_{pref}(\widehat{F}))$

Loop Encodings

Check if second guess is equal to the first one.

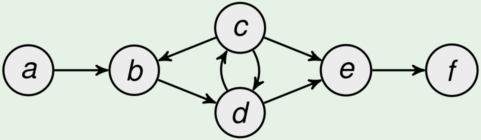
$$
\begin{array}{lcl}
\mathrm{equpto}(Y) & \leftarrow & \mathrm{inf}(Y), \mathrm{in}(Y), \mathrm{inN}(Y) \\
\mathrm{equpto}(Y) & \leftarrow & \mathrm{inf}(Y), \mathrm{out}(Y), \mathrm{outN}(Y) \\
\mathrm{equpto}(Y) & \leftarrow & \mathrm{succ}(Z, Y), \mathrm{in}(Y), \mathrm{inN}(Y), \mathrm{equpto}(Z) \\
\mathrm{equpto}(Y) & \leftarrow & \mathrm{succ}(Z, Y), \mathrm{out}(Y), \mathrm{outN}(Y), \mathrm{equpto}(Z) \\
\mathrm{eq} & \leftarrow & \mathrm{sup}(Y), \mathrm{equpto}(Y)
\end{array}
$$

Alternative Characterization for Preferred

Proposition 1

Let $F = (A, R)$ be an AF and $S \subseteq A$ be admissible in F. Then, $S \in \mathit{pref}(F)$ iff, for each $E \in \mathit{adm}(F)$ such that $E \not\subseteq S$, $E \cup S \notin \mathit{cf}(F)$.

Example



$\mathit{adm}(F) = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{a, d\}, \{c, f\}, \{a, c, f\}, \{a, d, f\}\}$, and $\mathit{pref}(F) = \{\{a, c, f\},\{a, d, f\}\}$

New Encodings for Preferred

Proposition 1

Let $F = (A, R)$ be an AF and $S \subseteq A$ be admissible in F. Then, $S \in \mathit{pref}(F)$ iff, for each $E \in \mathit{adm}(F)$ such that $E \not\subseteq S$, $E \cup S \notin \mathit{cf}(F)$.

π_{satpref^2}

$$
\pi_{\mathit{satpref}^2} = \left\{ \begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil}|\text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \\
 & & \text{witness}(Y) \\
\text{spoil} & \leftarrow & \text{in}(X), \text{witness}(Y), \text{att}(X, Y) \\
\text{witness}(X) & \leftarrow & \text{spoil}, \text{arg}(X) \\
 & \leftarrow & \mathit{not}\ \text{spoil}, \text{nontrivial}
\end{array} \right\}
$$

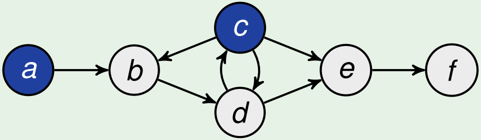
$$
\pi_{\mathit{pref}^2} = \pi_{\mathit{adm}} \cup \pi_{\mathit{satpref}^2}
$$

Result: For each AF F, $\mathit{pref}(F) \equiv \mathcal{AS}(\pi_{\mathit{pref}^2}(\widehat{F}))$

Functionality of New Encodings

$$\begin{array}{lcl} \text{nontrivial} & \leftarrow & \text{out}(X) \\ \text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \end{array}$$

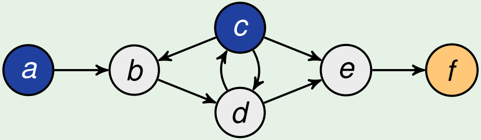
Example



Functionality of New Encodings

$$\begin{array}{rcl} \text{nontrivial} & \leftarrow & \text{out}(X) \\ \text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \end{array}$$

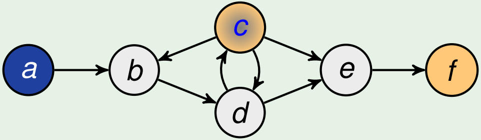
Example



Functionality of New Encodings

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil}|\text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X)
\end{array}
$$

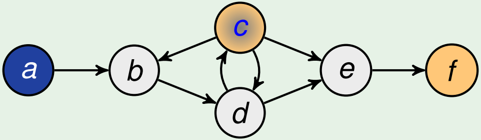
Example



Functionality of New Encodings

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil}|\text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \text{witness}(Y)
\end{array}
$$

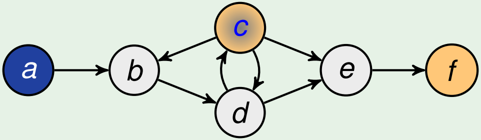
Example



Functionality of New Encodings

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil} | \text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \text{witness}(Y) \\
\text{spoil} & \leftarrow & \text{in}(X), \text{witness}(Y), \text{att}(X, Y)
\end{array}
$$

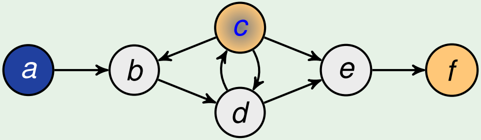
Example



Functionality of New Encodings

$$
\begin{array}{lcl}
\mathrm{nontrivial} & \leftarrow & \mathrm{out}(X) \\
\mathrm{witness}(X) : \mathrm{out}(X) & \leftarrow & \mathrm{nontrivial} \\
\mathrm{spoil} | \mathrm{witness}(Z) : \mathrm{att}(Z, Y) & \leftarrow & \mathrm{witness}(X), \mathrm{att}(Y, X) \\
\mathrm{spoil} & \leftarrow & \mathrm{att}(X, Y), \mathrm{witness}(X), \mathrm{witness}(Y) \\
\mathrm{spoil} & \leftarrow & \mathrm{in}(X), \mathrm{witness}(Y), \mathrm{att}(X, Y) \\
\mathrm{witness}(X) & \leftarrow & \mathrm{spoil}, \mathrm{arg}(X) \\
 & \leftarrow & \mathit{not}\ \mathrm{spoil}, \mathrm{nontrivial}
\end{array}
$$

Example

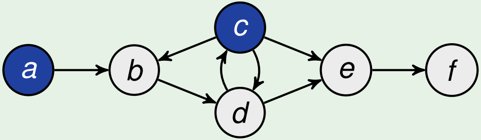


Functionality of New Encodings

Proposition 1

Let $F = (A, R)$ be an AF and $S \subseteq A$ be admissible in F. Then, $S \in pref(F)$ iff, for each $E \in adm(F)$ such that $E \not\subseteq S$, $E \cup S \notin cf(F)$.

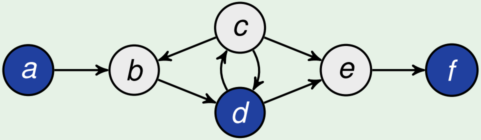
Example



Positive Example

$$\begin{aligned} \textcolor{red}{\text{nontrivial}} &\leftarrow \textcolor{red}{\text{out}(X)} \\ \text{witness}(X) : \text{out}(X) &\leftarrow \text{nontrivial} \end{aligned}$$

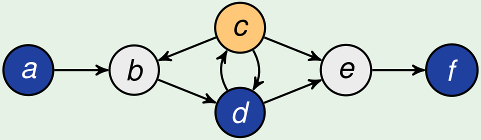
Example



Positive Example

$$\begin{aligned} \text{nontrivial} &\leftarrow \text{out}(X) \\ \text{witness}(X) : \text{out}(X) &\leftarrow \text{nontrivial} \end{aligned}$$

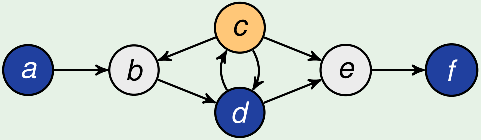
Example



Positive Example

$$\begin{array}{lcl} \text{nontrivial} & \leftarrow & \text{out}(X) \\ \text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\ \text{spoil}|\text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \end{array}$$

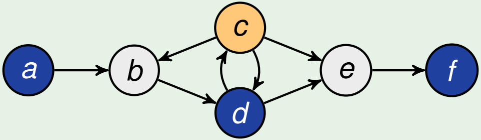
Example



Positive Example

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil}|\text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \text{witness}(Y)
\end{array}
$$

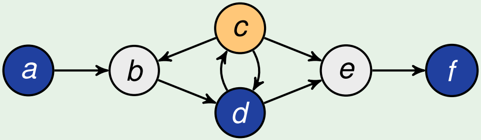
Example



Positive Example

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil} | \text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \text{witness}(Y) \\
\color{red}{\text{spoil}} & \color{red}{\leftarrow} & \color{red}{\text{in}(X), \text{witness}(Y), \text{att}(X, Y)}
\end{array}
$$

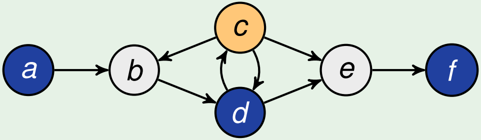
Example



Positive Example

$$
\begin{array}{lcl}
\text{nontrivial} & \leftarrow & \text{out}(X) \\
\text{witness}(X) : \text{out}(X) & \leftarrow & \text{nontrivial} \\
\text{spoil} | \text{witness}(Z) : \text{att}(Z, Y) & \leftarrow & \text{witness}(X), \text{att}(Y, X) \\
\text{spoil} & \leftarrow & \text{att}(X, Y), \text{witness}(X), \text{witness}(Y) \\
\text{spoil} & \leftarrow & \text{in}(X), \text{witness}(Y), \text{att}(X, Y) \\
\text{witness}(X) & \leftarrow & \text{spoil}, \text{arg}(X) \\
 & \leftarrow & \mathit{not}\ \text{spoil}, \text{nontrivial}
\end{array}
$$

Example



Evaluation

- New encodings were tested against CSP system ConArg, original encodings, and Metasp encodings
- Collection of 4972 frameworks (structured and random)
- Reasoning task: enumeration of all extensions
- 10 min timeout

Bull HPC-Cluster (Taurus)

- Intel Xeon CPU (E5-2670) with 2.60GHz
- 6.5 GB Ram, 600 seconds
- from 16 cores we used every 4th

We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time.

Results

PR	usc	solved	med
ConArg	60	2814	43.65
Original	-	3425	180.36
Meta	1	4626	20.83
New	101	4765	5.77

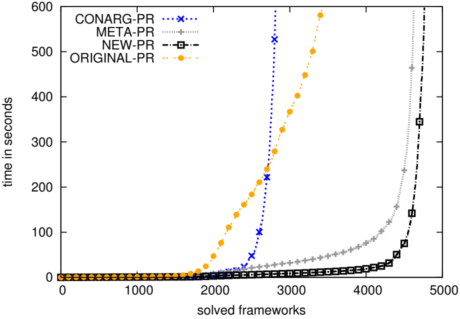


Figure : Runtimes for preferred (PR) semantics.

ICCMA 2015 Results

New encodings for preferred semantics reached in two categories of the first International Competition on Computational Models of Argument the 4th rank.

SE-PR

1. Cegartix
2. ArgSemSAT
3. LabSATSolver
4. ASPARTIX-V
5. CoQuiAAS
6. ASGL
7. ConArg
8. ASPARTIX-D
9. ArgTools
10. GRIS
11. DIAMOND
12. Dungell
13. Carneades

EE-PR

1. Cegartix
2. ArgSemSAT
3. CoQuiAAS
4. ASPARTIX-V
5. LabSATSolver
6. prefMaxSAT
7. ASGL
8. ASPARTIX-D
9. ConArg
10. ArgTools
11. ZJU-ARG
12. GRIS
13. DIAMOND
14. Dungell
15. Carneades

Conclusion and Future Work

- With new characterization we avoided complicated looping techniques
- New encodings clearly outperform original and metasp encodings
- New encodings scored good results at ICCMA 2015
- Same results also for stage and semi-stable semantics (in the paper)
- Encodings and benchmarks are available at
  `http://dbai.tuwien.ac.at/research/project/argumentation/systempage/#conditional`

Future Work

Optimize ASP encodings for ideal and eager semantics