Verifiability of Argumentation Semantics

Ringo Baumann, Thomas Linsbichler, Stefan Woltran

16th International Workshop on Non-Monotonic Reasoning
Cape Town, South Africa

April 22, 2016
Abstract Argumentation Framework (AF) [Dung, 1995]:
Abstract Argumentation Framework (AF) [Dung, 1995]:

Evaluation: argumentation semantics

Extension: set of jointly acceptable arguments
Introduction

- Abstract Argumentation Framework (AF) [Dung, 1995]:

  ![Diagram](image)

  - Evaluation: argumentation semantics
  - Extension: set of jointly acceptable arguments

  \[ stb(\mathcal{F}) = \]

Thomas Linsbichler, April 22, 2016
Verifiability of Argumentation Semantics
Abstract Argumentation Framework (AF) [Dung, 1995]:

- **Evaluation**: argumentation semantics
- **Extension**: set of jointly acceptable arguments

\[
\text{stb}(\mathcal{F}) = \{ a, d, e \},
\]
Abstract Argumentation Framework (AF) [Dung, 1995]:

- Evaluation: argumentation semantics
- Extension: set of jointly acceptable arguments

\[ stb(\mathcal{F}) = \{\{a, d, e\}, \{b, c, e\}\} \]
Introduction

- Abstract Argumentation Framework (AF) [Dung, 1995]:

- Evaluation: argumentation semantics
- Extension: set of jointly acceptable arguments

\[ stb(\mathcal{F}) = \{ \{a, d, e\}, \{b, c, e\} \} \]

- Further semantics: preferred, complete, semi-stable, stage, ...
Conflict-freeness: basic requirement for argumentation semantics.
Conflict-freeness: basic requirement for argumentation semantics.

Example

- Given conflict-free sets $\emptyset$, $\{a\}$, $\{b\}$. 
Conflict-freeness: basic requirement for argumentation semantics.

Example

- Given conflict-free sets $\emptyset$, $\{a\}$, $\{b\}$.
- Can we compute semantics based on this? 
  $\Rightarrow$ only naive semantics (maximal conflict-free sets)
Conflict-freeness: basic requirement for argumentation semantics.

Example

- Given conflict-free sets $\emptyset$, \{a\}, \{b\}.
- Can we compute semantics based on this? 
  $\Rightarrow$ only naive semantics (maximal conflict-free sets)

$\mathcal{F}: a \rightarrow b$ \hspace{1cm} $\mathcal{G}: a \rightarrow b$ \hspace{1cm} $\mathcal{H}: a \rightarrow b$
Conflict-freeness: basic requirement for argumentation semantics.

Example

- Given conflict-free sets $\emptyset$, $\{a\}$, $\{b\}$.
- Can we compute semantics based on this?
  - $\Rightarrow$ only naive semantics (maximal conflict-free sets)

$$\mathcal{F} : \text{ } a \rightarrow b \quad \mathcal{G} : \text{ } a \rightarrow b \quad \mathcal{H} : \text{ } a \rightarrow b$$

- Conflict free sets + their range: $\emptyset, \emptyset$, $\{a\}, \{a, b\}$, $\{b\}, \{b\}$
  - $\Rightarrow$ enough to compute stage semantics (range-maximal conflict-free sets)
Conflict-freeness: basic requirement for argumentation semantics.

Example

- Given conflict-free sets $\emptyset$, $\{a\}$, $\{b\}$.
- Can we compute semantics based on this?  
  $\Rightarrow$ only naive semantics (maximal conflict-free sets)

$$
\mathcal{F} : a \rightarrow b \quad \mathcal{G} : a \rightarrow b \quad H : a \leftrightarrow b
$$

- Conflict free sets + their range: $(\emptyset, \emptyset)$, $(\{a\}, \{a, b\})$, $(\{b\}, \{b\})$
  $\Rightarrow$ enough to compute stage semantics (range-maximal conflict-free sets)

Which information on top of conflict-free sets has to be added in order to compute a certain semantics?
Systematic comparison of argumentation semantics

- Principle-based evaluation [Baroni and Giacomin, 2007]
  ⇒ Hierarchy of verification classes
  ⇒ Each “rational” semantics is exactly verifiable by one of these classes
Systematic comparison of argumentation semantics

- Principle-based evaluation [Baroni and Giacomin, 2007]
  - Hierarchy of verification classes
  - Each “rational” semantics is exactly verifiable by one of these classes

Strong equivalence

- Studied for most argumentation semantics
  - [Oikarinen and Woltran, 2011, Baumann, 2016]
  - Missing results for naive and strong admissible semantics
  - Characterization theorems for intermediate semantics
An argumentation framework (AF) is a pair $(A, R)$ where
- $A \subseteq U$ is a finite set of arguments and
- $R \subseteq A \times A$ is the attack relation representing conflicts.

Given an AF $\mathcal{F} = (A, R)$ and $S \subseteq A$,
- $S$ is conflict-free ($S \in \text{cf}(\mathcal{F})$) if $\forall a, b \in S : (a, b) \notin R$.
- $a \in A$ is defended by $S$ if $\forall b \in A : (b, a) \in R \Rightarrow \exists c \in S : (c, b) \in R$
- $S^+_\mathcal{F} = S \cup \{a \mid \exists b \in S : (b, a) \in R\}$ (the range of $S$)
- $S^-_\mathcal{F} = S \cup \{a \mid \exists b \in S : (a, b) \in R\}$ (the anti-range of $S$)
Background

Definition

Given an AF $\mathcal{F} = (A, R)$, a set $S \subseteq A$ is

- **admissible set** if $S \in cf(\mathcal{F})$ and each $a \in S$ is defended by $S$,
- **complete extension** if $S \in ad(\mathcal{F})$ and $a \in S$ if $a$ is defended by $S$,
- **naive extension** if $S \in cf(\mathcal{F})$ and $\nexists T \in cf(\mathcal{F}) : T \supset S$,
- **stable extension** if $S \in cf(\mathcal{F})$ and $S^+_\mathcal{F} = A$,
- **stage extension** if $S \in cf(\mathcal{F})$ and $\nexists T \in cf(\mathcal{F}) : T^+_\mathcal{F} \supset S^+_\mathcal{F}$,
- **preferred, grounded, semi-stable, ideal, eager, strongly admissible extensions**
ad(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, d\}, \{b, c\}, \{a, d, e\}, \{b, c, e\}\}
Background

Example

\[ \text{ad}(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{a,d,e\}, \{b,c,e\} \} \]

\[ \text{co}(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d,e\}, \{b,c,e\}\} \]
Background

Example

\[ \text{ad}(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d\}, \{b,c\}, \{a,d,e\}, \{b,c,e\}\} \]

\[ \text{co}(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,d,e\}, \{b,c,e\}\} \]

\[ \text{na}(\mathcal{F}) = \{\{a,b,e\}, \{a,d,e\}, \{b,c,e\}\} \]
Background

Example

\[
ad(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, d\}, \{b, c\}, \{a, d, e\}, \{b, c, e\}\}
\]

\[
co(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, d, e\}, \{b, c, e\}\}
\]

\[
na(\mathcal{F}) = \{\{a, b, e\}, \{a, d, e\}, \{b, c, e\}\}
\]

\[
stb(\mathcal{F}) = stg(\mathcal{F}) = \{\{a, d, e\}, \{b, d, e\}\}
\]
We call a function \( r^x : 2^U \times 2^U \to (2^U)^n \) which is expressible via basic set operations only \(^a\) neighborhood function. A neighborhood function \( r^x \) induces the verification class mapping each AF \( \mathcal{F} \) to:

\[
\tilde{\mathcal{F}}^x = \left\{ \left( S, r^x(S^+_\mathcal{F}, S^-_\mathcal{F}) \right) \mid S \in \text{cf}(\mathcal{F}) \right\}.
\]

\(^a\) \( r^x(A, B) \) is in the language \( X ::= A \mid B \mid (X \cup X) \mid (X \cap X) \mid (X \setminus X) \)
Verifiability

**Definition**

We call a function \( r^x : 2^U \times 2^U \rightarrow (2^U)^n \) which is expressible via basic set operations only* a neighborhood function. A neighborhood function \( r^x \) induces the verification class mapping each AF \( \mathcal{F} \) to

\[
\tilde{\mathcal{F}}^x = \left\{ (S, r^x(S^+_{\mathcal{F}}, S^-_{\mathcal{F}})) \mid S \in cf(\mathcal{F}) \right\}.
\]

*\( r_x(A, B) \) is in the language \( X ::= A \mid B \mid (X \cup X) \mid (X \cap X) \mid (X \setminus X) \)

**Example**

\( \mathcal{F} : \quad a \rightarrow b \rightarrow c \)

\[
r^+ : r^x(A, B) = A \\
\tilde{\mathcal{F}}^+ = \{(\emptyset, \emptyset), (\{a\}, \{a, b\}), (\{c\}, \{b, c\}), (\{a, c\}, \{a, b, c\})\}
\]

\[
r^-_{\pm} : r^x(A, B) = (B, A \setminus B) \\
\tilde{\mathcal{F}}^-_{\pm} = \{(\emptyset, \emptyset, \emptyset), (\{a\}, \{a, b\}, \emptyset), (\{c\}, \{c\}, \{b\}), (\{a, c\}, \{a, b, c\}, \emptyset)\}
\]
Neighborhood functions for $n = 1$:

- $r^\epsilon(A, B) = \emptyset$
- $r^+(A, B) = A$
- $r^-(A, B) = B$
- $r^\top(A, B) = B \setminus A$
- $r^\bot(A, B) = A \setminus B$
- $r^\cap(A, B) = A \cap B$
- $r^\cup(A, B) = A \cup B$
- $r^\Delta(A, B) = (A \cup B) \setminus (A \cap B)$

$2^7 + 1$ syntactically different neighborhood functions

$\quad r^{x_1, \ldots, x_n}(A, B) ::= (r^{x_1}(A, B), \ldots, r^{x_n}(A, B))$
Verifiability

**Definition**

For neighborhood functions \( r^x \) and \( r^y \), we say that \( r^x \) is **more informative** than \( r^y \), short \( r^x \succeq r^y \), if there is a function \( \delta : (2^U)^n \rightarrow (2^U)^m \) such that for any \( A, B \subseteq U \), it holds that \( \delta (r^x (A, B)) = r^y (A, B) \).

In case \( r^x \approx r^y \) \((r^x \succeq r^y \text{ and } r^y \succeq r^x)\), we say that \( r^x \) **represents** \( r^y \).
Verifiability

Definition

For neighborhood functions $r^x$ and $r^y$, we say that $r^x$ is more informative than $r^y$, short $r^x \succeq r^y$, if there is a function $\delta : (2^\mathcal{U})^n \to (2^\mathcal{U})^m$ such that for any $A, B \subseteq \mathcal{U}$, it holds that $\delta (r^x(A, B)) = r^y(A, B)$.

In case $r^x \approx r^y$ ($r^x \succeq r^y$ and $r^y \succeq r^x$), we say that $r^x$ represents $r^y$.

Example

- $\delta_1(r^{\pm\cap}(A, B)) = \delta_1(A, A \setminus B) =_{def} (A, A \setminus (A \setminus B)) = (A, A \cap B) = r^{+\cap}(A, B)$;
- $\delta_2(r^{+\cap}(A, B)) = \delta_2(A, A \cap B) =_{def} (A \setminus (A \cap B), A \cap B) = (A \setminus B, A \cap B) = r^{\pm\cap}(A, B)$;
- $\delta_3(r^{\pm\cap}(A, B)) = \delta_3(A \setminus B, A \cap B) =_{def} ((A \setminus B) \cup (A \cap B), A \setminus B) = (A, A \setminus B) = r^{+\pm}(A, B)$.

$\Rightarrow r^{+\pm} \approx r^{+\cap} \approx r^{\pm\cap}$
Lemma

All neighborhood functions are represented by the ones depicted below and the $\prec$-relation represented by arcs holds.
A semantics $\sigma$ is verifiable by the verification class induced by the neighborhood function $\tau^x$ (or simply, $x$-verifiable) iff there is a function $\gamma_\sigma : (2^U)^n \times 2^U \rightarrow 2^{2^U}$ s.t. for every AF $\mathcal{F}$:

$$\gamma_\sigma \left( \tilde{\mathcal{F}}^x, A_\mathcal{F} \right) = \sigma(\mathcal{F}).$$

Moreover, $\sigma$ is exactly $x$-verifiable iff $\sigma$ is $x$-verifiable and there is no $\tau^y$ with $\tau^y \prec \tau^x$ such that $\sigma$ is $y$-verifiable.
Verifiability

Proposition

Complete semantics is exactly $\pm$-verifiable.

Proof

- Verifiability:

$$\gamma_{co}(\widetilde{\mathcal{F}}^{+-}, A_{\mathcal{F}}) = \{ S \mid (S, S^+, S^-) \in \widetilde{\mathcal{F}}^{+-}, (S^- \setminus S^+) = \emptyset, \quad \forall (\bar{S}, \bar{S}^+, \bar{S}^-) \in \widetilde{\mathcal{F}}^{+-} : \bar{S} \supset S \Rightarrow (\bar{S}^- \setminus \bar{S}^+) \neq \emptyset \}$$

- Exactness:

$$\Delta \pm : \quad \mathcal{F}_1 : \quad a \quad \overrightarrow{b} \quad \mathcal{F}_1' : \quad a \quad \overrightarrow{b}$$

- $$\widetilde{\mathcal{F}}_1^{\pm} = \{(\emptyset, \emptyset, \emptyset), (\{a\}, \{a\}, \emptyset)\} = \widetilde{\mathcal{F}}_1'$$

- $$co(\mathcal{F}_1) = \{\emptyset\} \neq \{\{a\}\} = co(\mathcal{F}_1')$$

$$\Rightarrow \quad co \text{ is not } \pm\text{-verifiable}$$
Complete semantics is exactly $+---$-verifiable.

Proof (ctd.)

1. $F_2 : a \rightarrow b \rightarrow c$
2. $F_2' : a \rightarrow b \rightarrow c$
3. $F_3 : a \rightarrow b$
4. $F_3' : a \rightarrow b$
5. $F_4 : a \rightarrow b$
6. $F_4' : a \rightarrow b$
7. $F_5 : a \rightarrow b$
8. $F_5' : a \rightarrow b$
9. $F_6 : a \rightarrow b$
10. $F_6' : a \rightarrow b$
Verifiability

\[ \epsilon: na \]
\[ +: \text{stb, stg} \]
\[ -: \text{ad, pr, id} \]
\[ +: \text{ss, eg} \]
\[ -: \text{gr, sad} \]
\[ +: \text{co} \]
Verifiability

**Definition**

We call a semantics $\sigma$ **rational** if self-loop-chains are irrelevant. That is, for every AF $\mathcal{F}$ it holds that $\sigma(\mathcal{F}) = \sigma(\mathcal{F}^l)$, where

$$\mathcal{F}^l = (A_\mathcal{F}, R_\mathcal{F} \setminus \{(a, b) \in R_\mathcal{F} \mid (a, a), (b, b) \in R_\mathcal{F}, a \neq b\}).$$
Verifiability

**Definition**

We call a semantics $\sigma$ **rational** if self-loop-chains are irrelevant. That is, for every AF $\mathcal{F}$ it holds that $\sigma(\mathcal{F}) = \sigma(\mathcal{F}^l)$, where

$$\mathcal{F}^l = (A_\mathcal{F}, R_\mathcal{F} \setminus \{(a, b) \in R_\mathcal{F} | (a, a), (b, b) \in R_\mathcal{F}, a \neq b\}).$$

**Theorem**

Every semantics which is rational is exactly verifiable by a verification class induced by one of the neighborhood functions below.
Strong Equivalence

Definition

Given semantics $\sigma$, two AFs $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent w.r.t. $\sigma$ ($\mathcal{F} \equiv_\sigma \mathcal{G}$) iff for all AFs $\mathcal{H}$: $\sigma(\mathcal{F} \cup \mathcal{H}) = \sigma(\mathcal{G} \cup \mathcal{H})$.
Strong Equivalence

Definition

Given semantics $\sigma$, two AFs $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent w.r.t. $\sigma$ ($\mathcal{F} \equiv^E \mathcal{G}$) iff for all AFs $\mathcal{H}$: $\sigma(\mathcal{F} \cup \mathcal{H}) = \sigma(\mathcal{G} \cup \mathcal{H})$

$\Rightarrow$ syntactical criteria exist

Example (stable semantics)

- $stb$-kernel: $\mathcal{F}^{k(stb)} = (A, R \setminus \{(a, b) \mid a \neq b, (a, a) \in R\})$.
- Theorem: $\mathcal{F}^{k(stb)} = \mathcal{G}^{k(stb)} \iff \mathcal{F}$ and $\mathcal{G}$ are strongly equivalent.

We have $\mathcal{F}^{k(stb)} = \mathcal{G}^{k(stb)} = \mathcal{G}$. Thus, $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent.
Strong Equivalence

**Definition (σ-kernel)**

Let $\mathcal{F} = (A, R)$. We define $\sigma$-kernels $\mathcal{F}^k(\sigma) = (A, R^k(\sigma))$ whereby

1. $R^k(\text{stb}) = R \setminus \{(a, b) | a \neq b, (a, a) \in R\}$,
2. $R^k(\text{ad}) = R \setminus \{(a, b) | a \neq b, (a, a) \in R, \{(b, a), (b, b)\} \cap R \neq \emptyset\}$,
3. $R^k(\text{gr}) = R \setminus \{(a, b) | a \neq b, (b, b) \in R, \{(a, a), (b, a)\} \cap R \neq \emptyset\}$,
4. $R^k(\text{co}) = R \setminus \{(a, b) | a \neq b, (a, a), (b, b) \in R\}$,
5. $R^k(\text{na}) = R \cup \{(a, b) | a \neq b, \{(a, a), (b, a), (b, b)\} \cap R \neq \emptyset\}$.

A relation $\equiv$ is **characterizable through kernels** if there is a kernel $k$, s.t. $\mathcal{F} \equiv \mathcal{G} \iff \mathcal{F}^k = \mathcal{G}^k$. 
### Strong Equivalence

#### Theorem

*Strong equivalence is characterizable through kernels (see below).*

<table>
<thead>
<tr>
<th></th>
<th>$stg$</th>
<th>$stb$</th>
<th>$ss$</th>
<th>$eg$</th>
<th>$ad$</th>
<th>$pr$</th>
<th>$id$</th>
<th>$gr$</th>
<th>$sad$</th>
<th>$co$</th>
<th>$na$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k(stb)$</td>
<td>$k(stb)$</td>
<td>$k(ad)$</td>
<td>$k(ad)$</td>
<td>$k(ad)$</td>
<td>$k(ad)$</td>
<td>$k(ad)$</td>
<td>$k(gr)$</td>
<td>$k(gr)$</td>
<td>$k(co)$</td>
<td>$k(na)$</td>
</tr>
</tbody>
</table>
Intermediate Semantics

- Note that \textit{stb} and \textit{stg} are both characterizable through \(k(stb)\).
- Does this also hold for arbitrary semantics \(\sigma\) with \(stb(F) \subseteq \sigma(F) \subseteq stg(F)\) for each AF \(F\)?
Intermediate Semantics

- Note that \( stb \) and \( stg \) are both characterizable through \( k(stb) \).
- Does this also hold for arbitrary semantics \( \sigma \) with \( stb(\mathcal{F}) \subseteq \sigma(\mathcal{F}) \subseteq stg(\mathcal{F}) \) for each AF \( \mathcal{F} \)?

Example

- “Stagle semantics”:
  \[ S \in sta(\mathcal{F}) \iff S \in cf(\mathcal{F}), \quad S_+^F \cup S_-^F = A_F \quad \text{and} \quad \forall T \in cf(\mathcal{F}) : S_+^F \not\subset T_+^F \]
Intermediate Semantics

- Note that \( stb \) and \( stg \) are both characterizable through \( k(stb) \).
- Does this also hold for arbitrary semantics \( \sigma \) with \( stb(\mathcal{F}) \subseteq \sigma(\mathcal{F}) \subseteq stg(\mathcal{F}) \) for each AF \( \mathcal{F} \)?

**Example**

- “Stagle semantics”:
  \[ S \in sta(\mathcal{F}) \iff S \in cf(\mathcal{F}), S^+_\mathcal{F} \cup S^-_\mathcal{F} = A_\mathcal{F} \text{ and } \forall T \in cf(\mathcal{F}) : S^+_\mathcal{F} \not\subset T^+_\mathcal{F} \]

\[ \mathcal{F} : \begin{array}{ccc} a & \rightarrow & b \\ \rightarrow & & \rightarrow \\ b & \rightarrow & c \end{array} \]

- \( stb(\mathcal{F}) = \emptyset \subset sta(\mathcal{F}) = \{\{b\}\} \subset stg(\mathcal{F}) = \{\{b\}, \{c\}\} \).
Intermediate Semantics

- Note that \( stb \) and \( stg \) are both characterizable through \( k(stb) \).
- Does this also hold for arbitrary semantics \( \sigma \) with \( stb(\mathcal{F}) \subseteq \sigma(\mathcal{F}) \subseteq stg(\mathcal{F}) \) for each AF \( \mathcal{F} \)?

**Example**

- "Stagle semantics":
  \[
  S \in st\alpha(\mathcal{F}) \iff S \in cf(\mathcal{F}), S^+_\mathcal{F} \cup S^-_\mathcal{F} = A_\mathcal{F} \text{ and } \forall T \in cf(\mathcal{F}) : S^+_\mathcal{F} \not\subseteq T^+_\mathcal{F}
  \]

- \( stb(\mathcal{F}) = \emptyset \subset st\alpha(\mathcal{F}) = \{\{b\}\} \subset stg(\mathcal{F}) = \{\{b\}, \{c\}\} \).

- \( st\alpha(\mathcal{F}^{k(stb)}) = \{\{b\}, \{c\}\} \Rightarrow \mathcal{F} \not\equiv_{st\alpha} \mathcal{F}^{k(stb)}, \mathcal{F}^{k(stb)} = (\mathcal{F}^{k(stb)})^{k(stb)} \)

\( \Rightarrow \) Stagle semantics is not compatible with the stable kernel.
Theorem
For each semantics $\sigma$ which is $++$-verifiable and $stb-stg$-intermediate, it holds that

$$\mathcal{F}^k(stb) = \mathcal{G}^k(stb) \iff \mathcal{F} \equiv_{E} \mathcal{G}.$$
Intermediate Semantics

**Theorem**

For each semantics $\sigma$ which is ++-verifiable and $stb-stg$-intermediate, it holds that

$$F^k(stb) = G^k(stb) \Leftrightarrow F \equiv^\sigma_E G.$$ 

**Theorem**

For each semantics $\sigma$ which is $+\mp$-verifiable and $\rho$-$ad$-intermediate with $\rho \in \{ss, id, eg\}$, it holds that

$$F^k(ad) = G^k(ad) \Leftrightarrow F \equiv^\sigma_E G.$$ 

**Theorem**

For each semantics $\sigma$ which is --±-verifiable and $gr$-$sad$-intermediate, it holds that

$$F^k(gr) = G^k(gr) \Leftrightarrow F \equiv^\sigma_E G.$$
Conclusion

Summary:

- Hierarchy of verification classes
- Each “rational” semantics is exactly verifiable by a certain class
- Characterization of strong equivalence for intermediate semantics

Future work:

- Semantics not captured by the approach, e.g. $cf2$ semantics [Baroni et al., 2005]
- Investigating labelling-based semantics [Caminada and Gabbay, 2009]


On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games.

Strongly equivalent logic programs.

Characterizing strong equivalence for argumentation frameworks.

Strong and uniform equivalence of nonmonotonic theories - an algebraic approach.

Strong equivalence for causal theories.
In 7th International Conference on Logic Programming and Nonmonotonic Reasoning, Proceedings, volume 2923 of Lecture Notes in Computer Science, pages 289–301. Springer.
\[ \gamma_{na}(\tilde{F}_A) = \{S \mid S \in \tilde{F}, S \text{ is } \subseteq\text{-maximal in } \tilde{F}\}; \]

\[ \gamma_{stg}(\tilde{F}_A^+) = \{S \mid (S, S^+) \in \tilde{F}^+, S^+ \text{ is } \subseteq\text{-maximal in } \{C^+ \mid (C, C^+ \in \tilde{F}^+)\}; \]

\[ \gamma_{stb}(\tilde{F}_A^+) = \{S \mid (S, S^+) \in \tilde{F}^+, S^+ = A\}; \]

\[ \gamma_{ad}(\tilde{F}_A^+) = \{S \mid (S, S^+) \in \tilde{F}^+, S^+ = \emptyset\}; \]

\[ \gamma_{pr}(\tilde{F}_A^+) = \{S \mid S \in \gamma_{ad}(\tilde{F}_A^+), S \text{ is } \subseteq\text{-maximal in } \gamma_{ad}(\tilde{F}_A^+)\}; \]

\[ \gamma_{ss}(\tilde{F}_A^+\mp) = \{S \mid S \in \gamma_{ad}(\tilde{F}_A^+)\}, S^+ \text{ is } \subseteq\text{-maximal in } \{C^+ \mid (C, C^+, C^\mp) \in \tilde{F}^+\mp, C \in \gamma_{ad}(\tilde{F}_A^+)\}; \]

\[ \gamma_{id}(\tilde{F}_A^+\mp) = \{S \mid S \text{ is } \subseteq\text{-maximal in } \{C \mid C \in \gamma_{ad}(\tilde{F}_A^+\mp), C \subseteq \bigcap \gamma_{pr}(\tilde{F}_A^+\mp)\}; \]

\[ \gamma_{eg}(\tilde{F}_A^+\mp) = \{S \mid S \text{ is } \subseteq\text{-maximal in } \{C \mid C \in \gamma_{ad}(\tilde{F}_A^+\mp), C \subseteq \bigcap \gamma_{ss}(\tilde{F}_A^+\mp)\}; \]

\[ \gamma_{sad}(\tilde{F}_A^\pm) = \{S \mid (S, S^-, S^\pm) \in \tilde{F}^\pm, \exists(S_0, S^-_0, S^\pm_0), \ldots, (S_n, S^-_n, S^\pm_n) \in \tilde{F}^\pm : \}

\[ (\emptyset = S_0 \subset \cdots \subset S_n = S \land \forall i \in \{1, \ldots, n\} : S^-_i \subseteq S^\pm_i \} ; \]

\[ \gamma_{eg}(\tilde{F}_A^-\mp) = \{S \mid S \in \gamma_{sad}(\tilde{F}_A^-\mp), \forall(\bar{S}, \bar{S}^-, \bar{S}^\pm) \in \tilde{F}^-\mp : \bar{S} \supset S \Rightarrow (\bar{S}^- \setminus S^\pm) \neq \emptyset\}. \]