A Translation-based Approach for Revision of Argumentation Frameworks

Sylvie Coste-Marquis Sébastien Konieczny
Jean-Guy Mailly Pierre Marquis

Centre de Recherche en Informatique de Lens
Université d'Artois – CNRS UMR 8188

14th European Conference on Logics in Artificial Intelligence
September 24th - 26th – Madeira

AMANDE

Schematic Explanation of the Approach

- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F

$$F, \sigma, \varphi$$

Schematic Explanation of the Approach

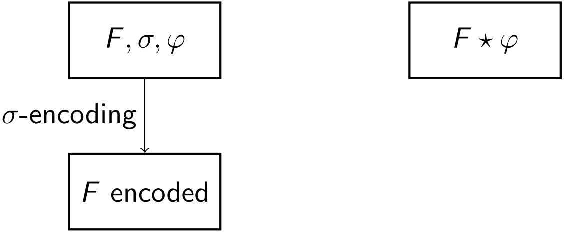
- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F

F, σ, φ

$F \star \varphi$

Schematic Explanation of the Approach

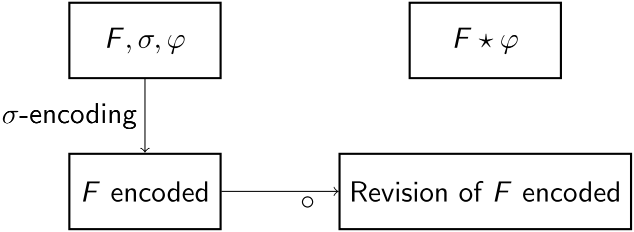
- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



AMANDE

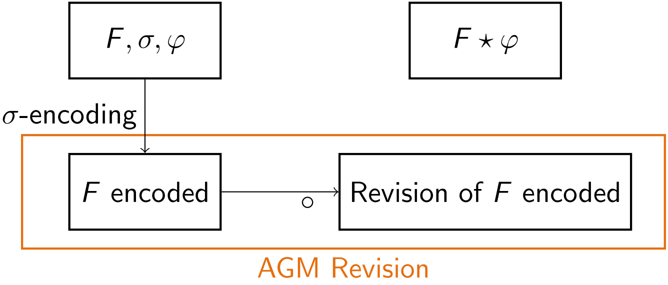
Schematic Explanation of the Approach

- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



Schematic Explanation of the Approach

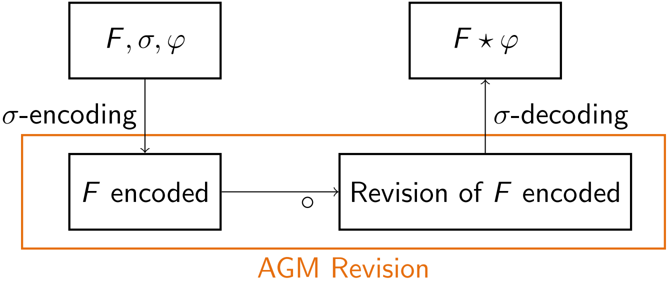
- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



AMANDE

Schematic Explanation of the Approach

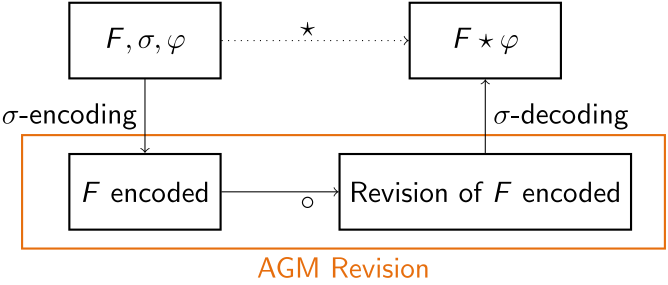
- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



AMANDE

Schematic Explanation of the Approach

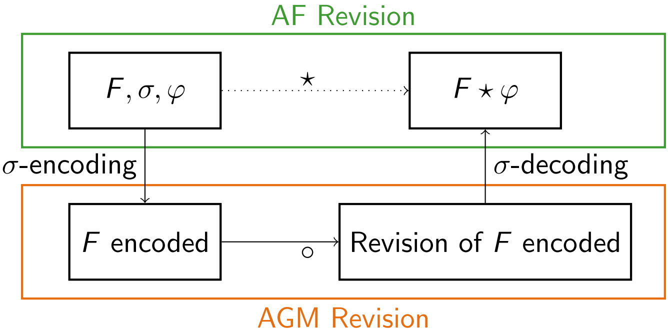
- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



AMANDE

Schematic Explanation of the Approach

- F: an argumentation framework
- σ: a semantics to define acceptable arguments
- φ: a propositional formula indicating how to revise F



AMANDE

Plan

Abstract Argumentation [Dung 1995]

- An abstract argumentation framework is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$ with $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$:

- An extension is a set of arguments that can be accepted together
  - Different semantics to define the extensions: complete, stable, preferred, grounded, etc.
- The aim is to know whether an argument is accepted or not w.r.t. the chosen semantics σ
  - An argument $a \in \mathcal{A}$ is (skeptically) accepted iff it belongs to every extension of the AF w.r.t. the considered semantics σ:

$$F \mid\!\sim_{\sigma} a \Leftrightarrow a \in \bigcap Ext_{\sigma}(F)$$

AGM framework for Belief Revision

- Principles of Belief Revision:

AGM framework for Belief Revision

- Principles of Belief Revision:
  - Primacy of update

AGM framework for Belief Revision

- Principles of Belief Revision:
  - Primacy of update
  - Consistency

AGM framework for Belief Revision

- Principles of Belief Revision:
  - Primacy of update
  - Consistency
  - Minimal change

AGM framework for Belief Revision

- Principles of Belief Revision:
  - Primacy of update
  - Consistency
  - Minimal change
- Set of postulates: characterizes the “good” operators

  [Alchourrón, Gärdenfors and Makinson 1985],[Katsuno et Mendelzon 1991]

AGM framework for Belief Revision

- Principles of Belief Revision:
  - Primacy of update
  - Consistency
  - Minimal change
- Set of postulates: characterizes the "good" operators

  [Alchourrón, Gärdenfors and Makinson 1985],[Katsuno et Mendelzon 1991]
- Representation theorem: "An operator satisfies the postulates iff it belongs to a particular class"

AMANDE

UNIVERSITÉ D'ARTOIS

AFs Revision

- Aim: Incorporation of a new piece of information about the attack relation and/or the acceptance statuses of arguments
- Two kind of minimal change:

$$\text{Attack} \neq \text{Acceptance}$$

Propositional Language

- $\forall x \in A$, $acc(x) = $ « x is skeptically accepted by F »
- $\forall x, y \in A$, $att(x, y) = $ « x attacks y in F »
- $Prop_A = \{acc(x) | x \in A\} \cup \{att(x, y) | x, y \in A\}$
- $\mathcal{L}_A$ is the propositional language built on the set of variables $Prop_A$ and the connectives $\neg, \vee, \wedge$

Encoding an AF

σ-formula of F

Given an AF $F = \langle A, R \rangle$ and a semantics σ, the σ-formula of F is

$$f_\sigma(F) = \bigwedge_{(x,y)\in R} att(x,y) \wedge \bigwedge_{(x,y)\notin R} \neg att(x,y)$$

Encoding an AF

σ-formula of F

Given an AF $F = \langle A, R\rangle$ and a semantics σ, the σ-formula of F is

$$f_\sigma(F) = \bigwedge_{(x,y)\in R} att(x,y) \wedge \bigwedge_{(x,y)\notin R} \neg att(x,y) \wedge th_\sigma(A)$$

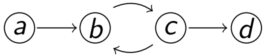
where the σ-theory of A $th_\sigma(A)$ is a formula which encodes the semantics σ.

Encoding the Stable Semantics (1)

Stable extensions of an AF $F = \langle A, R \rangle$ [Besnard and Doutre 2004]

$$\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b)$$

Example

$F_1 =$ 

One stable extension: $\{a, c\}$

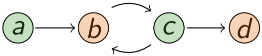
AMANDE

Encoding the Stable Semantics (1)

Stable extensions of an AF $F = \langle A, R \rangle$ [Besnard and Doutre 2004]

$$\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b)$$

Example

$F_1 =$ 

One stable extension: $\{a, c\}$

AMANDE

UNIVERSITÉ D'ARTOIS

Encoding the Stable Semantics (2)

Skeptical acceptance of an argument a_i

$$\models \left(\bigwedge_{a \in A} \left(a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b \right) \Rightarrow a_i \right)$$

AMANDE

Encoding the Stable Semantics (2)

Skeptical acceptance of an argument a_i

$$\models \left(\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i \right)$$

equivalent to

$$\forall a_1, \ldots, a_n, \left(\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i \right)$$

Encoding the Stable Semantics (3)

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i)$$

- Knowledge of the AF is required *a priori*

AMANDE

Encoding the Stable Semantics (3)

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i)$$

- Knowledge of the AF is required *a priori*
- We use $att(x, y)$ to relax this requirement

Encoding the Stable Semantics (3)

$$\forall a_1, \ldots, a_n, \left(\bigwedge_{a \in A} \left(a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b\right) \Rightarrow a_i\right)$$

- Knowledge of the AF is required *a priori*
- We use $att(x, y)$ to relax this requirement

$$\forall a_1, \ldots, a_n, \left(\bigwedge_{a \in A} \left(a \Leftrightarrow \bigwedge_{b \in A} (att(b, a) \Rightarrow \neg b)\right) \Rightarrow a_i\right)$$

Encoding the Stable Semantics (3)

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i)$$

- Knowledge of the AF is required *a priori*
- We use $att(x, y)$ to relax this requirement

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b \in A} (att(b, a) \Rightarrow \neg b)) \Rightarrow a_i)$$

Stable theory of the set A

$$\begin{aligned} th_{st}(A) = \quad & \textstyle\bigwedge_{a_i \in A}(acc(a_i) \Leftrightarrow \forall a_1, \ldots, a_n, \\ & (\textstyle\bigwedge_{a \in A}(a \Leftrightarrow \bigwedge_{b \in A}(att(b, a) \Rightarrow \neg b)) \Rightarrow a_i)) \end{aligned}$$

Encoding the Stable Semantics (3)

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b:(b,a) \in R} \neg b) \Rightarrow a_i)$$

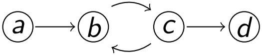
- Knowledge of the AF is required *a priori*
- We use $att(x, y)$ to relax this requirement

$$\forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b \in A} (att(b, a) \Rightarrow \neg b)) \Rightarrow a_i)$$

Stable theory of the set A

$$th_{st}(A) = \bigwedge_{a_i \in A} (acc(a_i) \Leftrightarrow \forall a_1, \ldots, a_n, (\bigwedge_{a \in A} (a \Leftrightarrow \bigwedge_{b \in A} (att(b, a) \Rightarrow \neg b)) \Rightarrow a_i))$$

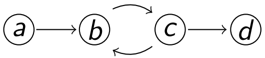
Example

$F_1 =$ 

$$th_{st}(A) =$$

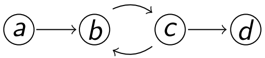
$$acc(a) \Leftrightarrow \quad \forall a, b, c, d, [[(a \Leftrightarrow ((att(a,a) \Rightarrow \neg a) \wedge (att(b,a) \Rightarrow \neg b) \wedge (att(c,a) \Rightarrow \neg c) \wedge (att(d,a) \Rightarrow \neg d)))$$

Example

$F_1 =$ 

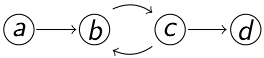
$$
\begin{aligned}
&th_{st}(A) = \\
&acc(a) \Leftrightarrow \quad \forall a, b, c, d, [[(a \Leftrightarrow ((att(a,a) \Rightarrow \neg a) \wedge (att(b,a) \Rightarrow \neg b) \\
&\qquad\qquad\qquad\qquad \wedge (att(c,a) \Rightarrow \neg c) \wedge (att(d,a) \Rightarrow \neg d))) \\
&\qquad\qquad \wedge (b \Leftrightarrow ((att(a,b) \Rightarrow \neg a) \wedge (att(b,b) \Rightarrow \neg b) \\
&\qquad\qquad\qquad\qquad \wedge (att(c,b) \Rightarrow \neg c) \wedge (att(d,b) \Rightarrow \neg d)))
\end{aligned}
$$

Example

$F_1 =$ 

$$
\begin{aligned}
& th_{st}(A) = \\
& acc(a) \Leftrightarrow \quad \forall a, b, c, d, [[(a \Leftrightarrow ((att(a,a) \Rightarrow \neg a) \wedge (att(b,a) \Rightarrow \neg b) \\
& \qquad\qquad\qquad\qquad \wedge (att(c,a) \Rightarrow \neg c) \wedge (att(d,a) \Rightarrow \neg d))) \\
& \qquad\qquad \wedge (b \Leftrightarrow ((att(a,b) \Rightarrow \neg a) \wedge (att(b,b) \Rightarrow \neg b) \\
& \qquad\qquad\qquad\qquad \wedge (att(c,b) \Rightarrow \neg c) \wedge (att(d,b) \Rightarrow \neg d))) \\
& \qquad\qquad \wedge (c \Leftrightarrow ((att(a,c) \Rightarrow \neg a) \wedge (att(b,c) \Rightarrow \neg b) \\
& \qquad\qquad\qquad\qquad \wedge (att(c,c) \Rightarrow \neg c) \wedge (att(d,c) \Rightarrow \neg d)))
\end{aligned}
$$

Example

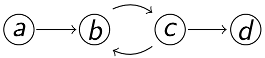
$F_1 =$ 

$$
\begin{aligned}
&th_{st}(A) = \\
&acc(a) \Leftrightarrow \quad \forall a, b, c, d, [[(a \Leftrightarrow ((att(a,a) \Rightarrow \neg a) \wedge (att(b,a) \Rightarrow \neg b) \\
&\qquad\qquad\qquad \wedge (att(c,a) \Rightarrow \neg c) \wedge (att(d,a) \Rightarrow \neg d))) \\
&\qquad\qquad \wedge (b \Leftrightarrow ((att(a,b) \Rightarrow \neg a) \wedge (att(b,b) \Rightarrow \neg b) \\
&\qquad\qquad\qquad \wedge (att(c,b) \Rightarrow \neg c) \wedge (att(d,b) \Rightarrow \neg d))) \\
&\qquad\qquad \wedge (c \Leftrightarrow ((att(a,c) \Rightarrow \neg a) \wedge (att(b,c) \Rightarrow \neg b) \\
&\qquad\qquad\qquad \wedge (att(c,c) \Rightarrow \neg c) \wedge (att(d,c) \Rightarrow \neg d))) \\
&\qquad\qquad \wedge (d \Leftrightarrow ((att(a,d) \Rightarrow \neg a) \wedge (att(b,d) \Rightarrow \neg b) \\
&\qquad\qquad\qquad \wedge (att(c,d) \Rightarrow \neg c) \wedge (att(d,d) \Rightarrow \neg d)))] \Rightarrow a]
\end{aligned}
$$

AMANDE

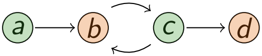
UNIVERSITÉ D'ARTOIS

Example

$F_1 =$ 

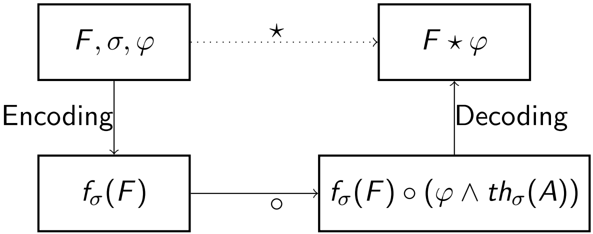
$$
\begin{aligned}
& th_{st}(A) = \\
& acc(a) \Leftrightarrow \quad \forall a, b, c, d, [[(a \Leftrightarrow ((att(a,a) \Rightarrow \neg a) \wedge (att(b,a) \Rightarrow \neg b) \\
& \qquad\qquad \wedge (att(c,a) \Rightarrow \neg c) \wedge (att(d,a) \Rightarrow \neg d))) \\
& \qquad \wedge (b \Leftrightarrow ((att(a,b) \Rightarrow \neg a) \wedge (att(b,b) \Rightarrow \neg b) \\
& \qquad\qquad \wedge (att(c,b) \Rightarrow \neg c) \wedge (att(d,b) \Rightarrow \neg d))) \\
& \qquad \wedge (c \Leftrightarrow ((att(a,c) \Rightarrow \neg a) \wedge (att(b,c) \Rightarrow \neg b) \\
& \qquad\qquad \wedge (att(c,c) \Rightarrow \neg c) \wedge (att(d,c) \Rightarrow \neg d))) \\
& \qquad \wedge (d \Leftrightarrow ((att(a,d) \Rightarrow \neg a) \wedge (att(b,d) \Rightarrow \neg b) \\
& \qquad\qquad \wedge (att(c,d) \Rightarrow \neg c) \wedge (att(d,d) \Rightarrow \neg d)))] \Rightarrow a] \\
\wedge \quad & acc(b) \Leftrightarrow \quad \forall a, b, c, d, [\cdots \Rightarrow b] \\
\wedge \quad & acc(c) \Leftrightarrow \quad \forall a, b, c, d, [\cdots \Rightarrow c] \\
\wedge \quad & acc(d) \Leftrightarrow \quad \forall a, b, c, d, [\cdots \Rightarrow d]
\end{aligned}
$$

Example

$F_1 =$ 

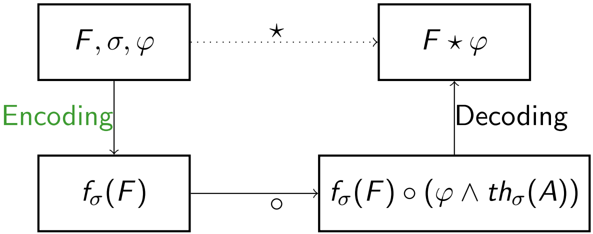
- $f_{st}(F_1) = th_{st}(A) \wedge \bigwedge_{(a,b)\in R} att(a,b) \wedge \bigwedge_{(a,b)\notin R} \neg att(a,b)$
- Propagating the values of the $att(x,y)$ variables, we get the values of the $acc(x)$:
  $acc(a) = acc(c) = true$ and $acc(b) = acc(d) = false$
  So the arguments accepted by F_1 are: $\{a, c\}$

Reminder

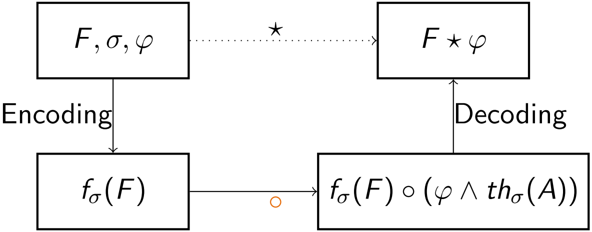


F, σ, φ
$\star$
$F \star \varphi$
Encoding
Decoding
$f_\sigma(F)$
$\circ$
$f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))$

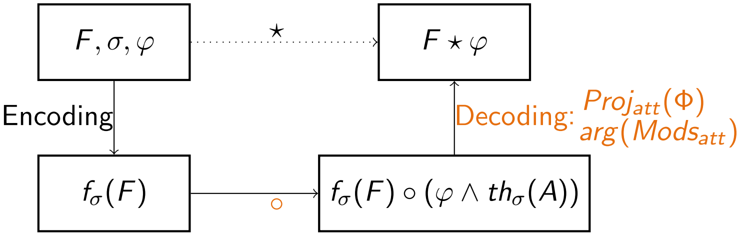
Reminder


F, σ, φ
$\star$
$F \star \varphi$
Encoding
Decoding
$f_\sigma(F)$
$\circ$
$f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))$

Reminder


F, σ, φ
$\star$
$F \star \varphi$
Encoding
Decoding
$f_\sigma(F)$
$\circ$
$f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))$

Reminder



F, σ, φ
$\star$
$F \star \varphi$
Encoding
Decoding: $Proj_{att}(\Phi)$ $arg(Mods_{att})$
$f_\sigma(F)$
$\circ$
$f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))$

Decoding Tools

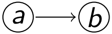
- $Proj_{att}(\Phi)$: projection of the models of Φ on the variables $att(x, y)$
- $arg(Mods_{att})$: generation of AFs from models projected on $att(x, y)$

Decoding Tools

- $Proj_{att}(\Phi)$: projection of the models of Φ on the variables $att(x, y)$
- $arg(Mods_{att})$: generation of AFs from models projected on $att(x, y)$

Example of decoding

With $A = \{a, b\}$, the revised models could be:
$Mod(\Phi) = \{\{acc(a), \neg acc(b), \neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$.
So, $Proj_{att}(\Phi) = \{\{\neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$ and
$arg(Proj_{att}(\Phi)) = \{F\}$ with F the AF below:



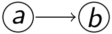
AMANDE

Decoding Tools

- $Proj_{att}(\Phi)$: projection of the models of Φ on the variables $att(x, y)$
- $arg(Mods_{att})$: generation of AFs from models projected on $att(x, y)$

Example of decoding

With $A = \{a, b\}$, the revised models could be:
$Mod(\Phi) = \{\{acc(a), \neg acc(b), \neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$.
So, $Proj_{att}(\Phi) = \{\{\neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$ and
$arg(Proj_{att}(\Phi)) = \{F\}$ with F the AF below:



AMANDE

Decoding Tools

- $Proj_{att}(\Phi)$: projection of the models of Φ on the variables $att(x, y)$
- $arg(Mods_{att})$: generation of AFs from models projected on $att(x, y)$

Example of decoding

With $A = \{a, b\}$, the revised models could be:
$Mod(\Phi) = \{\{acc(a), \neg acc(b), \neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$.
So, $Proj_{att}(\Phi) = \{\{\neg att(a, a), att(a, b), \neg att(b, a), \neg att(b, b)\}\}$ and
$arg(Proj_{att}(\Phi)) = \{F\}$ with F the AF below:

AMANDE

Translation-based Revision Operator

Translation-based Revision

Let $\circ$ be a KM revision operator. For every semantics σ, every AF $F = \langle A, R \rangle$ and every formula $\varphi \in \mathcal{L}_A$, the associated *translation-based revision operator* $\star$ is defined by:

$$F \star \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))))$$

AMANDE

UNIVERSITÉ D'ARTOIS

Translation-based Revision Operator

Translation-based Revision

Let $\circ$ be a KM revision operator. For every semantics σ, every AF $F = \langle A, R \rangle$ and every formula $\varphi \in \mathcal{L}_A$, the associated *translation-based revision operator* $\star$ is defined by:

$$F \star \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A))))$$

AMANDE

UNIVERSITÉ D'ARTOIS

Distance-based Revision

Let d be a distance between interpretations on $\mathcal{L}_A$. Given a formula $\psi \in \mathcal{L}_A$, the pre-order $\leq_\psi$ is defined by:

$$\omega \leq_\psi \omega' \text{ iff } d(\omega, Mod(\psi)) \leq d(\omega', Mod(\psi))$$

AMANDE

UNIVERSITÉ D'ARTOIS

Distance-based Revision

Let d be a distance between interpretations on $\mathcal{L}_A$. Given a formula $\psi \in \mathcal{L}_A$, the pre-order $\leq_\psi$ is defined by:

$$\omega \leq_\psi \omega' \text{ iff } d(\omega, Mod(\psi)) \leq d(\omega', Mod(\psi))$$

The KM revision operator $\circ_d$ based on d is defined by:

$$Mod(\psi \circ_d \alpha) = \min(Mod(\alpha), \leq_\psi)$$

Distance-based Revision

Let d be a distance between interpretations on $\mathcal{L}_A$. Given a formula $\psi \in \mathcal{L}_A$, the pre-order $\leq_\psi$ is defined by:

$$\omega \leq_\psi \omega' \text{ iff } d(\omega, Mod(\psi)) \leq d(\omega', Mod(\psi))$$

The KM revision operator $\circ_d$ based on d is defined by:

$$Mod(\psi \circ_d \alpha) = \min(Mod(\alpha), \leq_\psi)$$

The AF revision operator $\star_d$ *based on distance* d is defined by:

$$F \star_d \varphi = arg(Proj_{att}(f_\sigma(F) \circ_d (\varphi \wedge th_\sigma(A))))$$

Distances and Minimal Change

Priority to Minimal Change on Acceptance Statuses

Let A be a set of arguments, and $N = |A|^2 + 1$.

$$d_H^{acc}(\omega, \omega') = \sum_{a \in A}(\omega(acc(a)) \oplus \omega'(acc(a))) + \sum_{a,b \in A}(\omega(att(a,b)) \oplus \omega'(att(a,b)))$$

Distances and Minimal Change

Priority to Minimal Change on Acceptance Statuses

Let A be a set of arguments, and $N = |A|^2 + 1$.

$$d_H^{acc}(\omega, \omega') = N \times \sum_{a \in A}(\omega(acc(a)) \oplus \omega'(acc(a))) + \sum_{a,b \in A}(\omega(att(a,b)) \oplus \omega'(att(a,b)))$$

Distances and Minimal Change

Priority to Minimal Change on Acceptance Statuses

Let A be a set of arguments, and $N = |A|^2 + 1$.

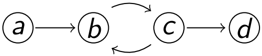
$$\begin{aligned} d_H^{acc}(\omega, \omega') = \; & N \times \sum_{a \in A} (\omega(acc(a)) \oplus \omega'(acc(a))) \\ & + \sum_{a,b \in A} (\omega(att(a,b)) \oplus \omega'(att(a,b))) \end{aligned}$$

Priority to Minimal Change on the Attack Relation

Let A be a set of arguments, and $N = |A| + 1$.

$$\begin{aligned} d_H^{att}(\omega, \omega') = \; & \sum_{a \in A} (\omega(acc(a)) \oplus \omega'(acc(a))) \\ & + N \times \sum_{a,b \in A} (\omega(att(a,b)) \oplus \omega'(att(a,b))) \end{aligned}$$

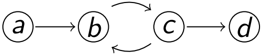
Example

$F_1 =$ 

Accepted arguments: $\{a, c\}$

Revision by $\varphi = acc(a) \land \neg att(a, b)$ with priority to minimal change on...

Example

$F_1 =$ 

Accepted arguments: $\{a, c\}$

Revision by $\varphi = acc(a) \wedge \neg att(a, b)$ with priority to minimal change on...

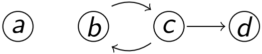
...the attack relation:

1 acceptance status change

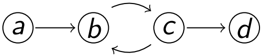
- $\{a\}$

1 attack change

- removal of (a, b)

$F_2 =$ 

Example

$F_1 =$ 

Accepted arguments: $\{a, c\}$

Revision by $\varphi = acc(a) \wedge \neg att(a, b)$ with priority to minimal change on...

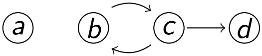
...the attack relation:

1 acceptance status change

- $\{a\}$

1 attack change

- removal of (a, b)

$F_2 =$ 

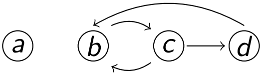
...acceptance statuses:

0 status change

- $\{a, c\}$

2 attack changes

- removal of (a, b)
- addition of (d, b)

$F_3 =$ 

Restriction to Acceptance Formulae

Postulates adapted from the standard belief revision [Katsuno et Mendelzon 1991]

(AS1) $Sc_\sigma(F \star \varphi) \subseteq \mathcal{S}(\varphi)$

(AS2) If $Sc_\sigma(F) \cap \mathcal{S}(\varphi) \neq \emptyset$, then $Sc_\sigma(F \star \varphi) = Sc_\sigma(F) \cap \mathcal{S}(\varphi)$

(AS3) If φ is *acc*-consistent, then $Sc_\sigma(F \star \varphi) \neq \emptyset$

(AS4) If $\varphi \equiv_{acc} \psi$, then $Sc_\sigma(F \star \varphi) = Sc_\sigma(F \star \psi)$

(AS5) $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \subseteq Sc_\sigma(F \star (\varphi \wedge \psi))$

(AS6) If $Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi) \neq \emptyset$, then $Sc_\sigma(F \star (\varphi \wedge \psi)) \subseteq Sc_\sigma(F \star \varphi) \cap \mathcal{S}(\psi)$

AF Revision based on a Distance

Proposition

Given a pseudo-distance d between sets of arguments, and an AF F, $\leq_F^d$ is the total pre-order between sets of arguments defined by:

$$\varepsilon_1 \leq_F^d \varepsilon_2 \text{ iff } d(\varepsilon_1, Sc_\sigma(F)) \leq d(\varepsilon_2, Sc_\sigma(F)).$$

Given φ and *acc*-formula, *the revision operator based on the pseudo-distance d* $\star_d$ which satisfies

$$Sc_\sigma(F \star_d \varphi) = \min(\mathcal{S}(\varphi), \leq_F^d)$$

satisfies **(AS1)** - **(AS6)**.

Coming Back to Minimal Change on Arguments Statuses

Proposition

The revision operator with priority to minimal change on the arguments statuses, restricted to acceptance formulae, satisfies postulates **(AS1)-(AS6)**.

Intuition: can be proved by reducing this operator to another one based on a distance between sets of arguments.

Possible Extensions of the Approach

Open World Revision

Given $F = \langle A, R \rangle$ an AF, B a non-empty set of arguments such that $A \cap B = \emptyset$, $\varphi \in \mathcal{L}_{A \cup B}$ a formula and $\circ$ a KM revision operator, the associated *open world revision operator* $\star_B$ is defined by:

$$F \star_B \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A \cup B))))$$

AMANDE

Possible Extensions of the Approach

Open World Revision

Given $F = \langle A, R \rangle$ an AF, B a non-empty set of arguments such that $A \cap B = \emptyset$, $\varphi \in \mathcal{L}_{A \cup B}$ a formula and $\circ$ a KM revision operator, the associated *open world revision operator* $\star_B$ is defined by:

$$F \star_B \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A \cup B))))$$

Constrained Revision

Given $F = \langle A, R \rangle$ an AF, $\varphi, \mu \in \mathcal{L}_A$ two formulae and $\circ$ a KM revision operator, the associated μ-*constrained revision operator* is defined by:

$$F \star_\mu \varphi = arg(Proj_{att}(f_\sigma(F) \circ (\varphi \wedge th_\sigma(A) \wedge \mu)))$$

Summary of this Work

- Revision method for AFs by translation into propositional logic
  - Generic method: works for any semantics
  - Extended to open world revision and constrained revision
  - Use of a rich language: $acc(x)$ and $att(x, y)$
- Encoding skeptical acceptance for stable and complete semantics
- Definition of a family of revision operators: distance-based revision operators
  - Operators adapted to both kind of minimal change: acceptance statuses and attacks
- Characterization of revision operators restricted to the *acc* case

Future Work

- Characterization of other semantics and credulous acceptance
- Rationality postulates for other kinds of operators:
  - Minimal change on the attack relation
  - Formulae on attacks/Unrestricted formulae
  - Open world revision, constrained revision
- Implementations of revision operators: SAT solvers

AMANDE

UNIVERSITÉ D'ARTOIS

Future Work

- Characterization of other semantics and credulous acceptance
- Rationality postulates for other kinds of operators:
  - Minimal change on the attack relation
  - Formulae on attacks/Unrestricted formulae
  - Open world revision, constrained revision
- Implementations of revision operators: SAT solvers

Thank you for your attention!

AMANDE