# Merging of Abstract Argumentation Frameworks

Jérôme Delobelle\(^1\), Adrian Haret\(^2\), Sébastien Konieczny\(^1\)
Jean-Guy Mailly\(^2\), Julien Rossit\(^3\), Stefan Woltran\(^2\)

<table>
<thead>
<tr>
<th>1: CRIL – CNRS, France</th>
</tr>
</thead>
<tbody>
<tr>
<td>2: Institute of Information Systems – TU Wien, Austria</td>
</tr>
<tr>
<td>3: LIPADE – Université Paris Descartes, France</td>
</tr>
</tbody>
</table>

15th International Conference on Principles of Knowledge Representation and Reasoning
25 – 29 April 2016
Outline

Background Notions
  Dung’s AFs
  Revising Dung’s AFs

Merging Operators for AFs
  Extension-based Merging
  From Extensions to AFs
  Resolute Merging

Comparison with the Literature
  Fusion Operators vs Merging Postulates
  Merging Operators vs Aggregation Axioms
  Discussion: Attack-based vs Extension-based Merging

Conclusion
Outline

**Background Notions**
- Dung’s AFs
- Revising Dung’s AFs

**Merging Operators for AFs**
- Extension-based Merging
- From Extensions to AFs
- Resolute Merging

**Comparison with the Literature**
- Fusion Operators vs Merging Postulates
- Merging Operators vs Aggregation Axioms
- Discussion: Attack-based vs Extension-based Merging

**Conclusion**
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph $F = \langle A, R \rangle$, $A$ is the set of arguments and $R \subseteq A \times A$ is the attack relation.
- Evaluation of arguments: Many semantics to compute extensions
  - grounded, stable, preferred, complete, …
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph \( F = \langle A, R \rangle \), \( A \) is the set of arguments and \( R \subseteq A \times A \) is the attack relation.
- Evaluation of arguments: Many semantics to compute extensions.
  - grounded, stable, preferred, complete, ...
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph $F = \langle A, R \rangle$, $A$ is the set of arguments and $R \subseteq A \times A$ is the attack relation.
- Evaluation of arguments: Many semantics to compute extensions.
  - grounded, stable, preferred, complete, …

$Ext_{st}(F) = \{ \{a_1, a_4, a_6\} \}$
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph $F = \langle A, R \rangle$, $A$ is the set of arguments and $R \subseteq A \times A$ is the attack relation.
- Evaluation of arguments: Many semantics to compute extensions.
  - grounded, stable, preferred, complete, ...

$$Ext_{pr}(F) = \{ \{a_1, a_4, a_6\}, \{a_1, a_3\}\}$$
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph $F = \langle A, R \rangle$, $A$ is the set of arguments and $R \subseteq A \times A$ is the attack relation.
- Evaluation of arguments: Many semantics to compute extensions
  - grounded, stable, preferred, complete, ...

$$Ext_{pr}(F) = \{ \{a_1, a_4, a_6\}, \{a_1, a_3\} \}$$
Abstract AF [Dung, AIJ 1995]

- An AF is a digraph $F = \langle A, R \rangle$, $A$ is the set of arguments and $R \subseteq A \times A$ is the attack relation
- Evaluation of arguments: Many semantics to compute extensions
  - grounded, stable, preferred, complete, ...

$$Ext_{co}(F) = \{\{a_1, a_4, a_6\}, \{a_1, a_3\}, \{a_1\}\}$$
Revision of AFs

[Coste et al, KR 2014]

- Revision of an AF $F$ by a formula $\varphi$ which expresses conditions on extensions
- A two-step process:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>revised extensions</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>${F'_1, \ldots, F'_k}$</td>
</tr>
</tbody>
</table>

[Diller et al, IJCAI 2015]

Modification of rationality postulates: result is required to be a single AF

[Dunne et al, AIJ 2015]

Inputs $F$, $\varphi$} \quad \rightarrow \quad \text{revised extensions} \quad \rightarrow \quad \{F'_1, \ldots, F'_k\}$
Revision of AFs

[Coste et al, KR 2014]

- Revision of an AF $F$ by a formula $\varphi$ which expresses conditions on extensions
- A two-step process:
  
  $\begin{align*}
  \text{Inputs} & : F, \varphi \\
  \text{Outputs} & : \text{revised extensions} \rightarrow \{F'_1, \ldots, F'_k\}
  \end{align*}$

[Diller et al, IJCAI 2015]

- Modification of rationality postulates: result is required to be a single AF [Dunne et al, AIJ 2015]

$\begin{align*}
\text{Inputs} & : F, \varphi \\
\text{Outputs} & : \text{realizable revised extensions} \rightarrow F'
\end{align*}$
Outline

Background Notions
  Dung’s AFs
  Revising Dung’s AFs

Merging Operators for AFs
  Extension-based Merging
  From Extensions to AFs
  Resolute Merging

Comparison with the Literature
  Fusion Operators vs Merging Postulates
  Merging Operators vs Aggregation Axioms
  Discussion: Attack-based vs Extension-based Merging

Conclusion
Merging of a profile of AFs $\langle F_1, \ldots, F_n \rangle$, with an integrity constraint $\mu$ which expresses conditions on extensions

A two-step process:

Inputs $\langle F_1, \ldots, F_n \rangle \quad \mu \quad \Rightarrow \quad$ extensions $\quad \Rightarrow \quad \{ F'_1, \ldots, F'_k \}$

Questions:

- How to obtain the extensions?
- How to obtain the AFs?
Merging of a profile of AFs $\langle F_1, \ldots, F_n \rangle$, with an integrity constraint $\mu$ which expresses conditions on extensions

A two-step process:

\[
\begin{align*}
\text{Inputs} & \quad \langle F_1, \ldots, F_n \rangle \quad \mu \\
\text{extensions} & \quad \Rightarrow \\
\text{Outputs} & \quad \{ F'_1, \ldots, F'_k \}
\end{align*}
\]

Questions:

- How to obtain the extensions?
- How to obtain the AFs?
Merging of a profile of AFs $\langle F_1, \ldots, F_n \rangle$, with an integrity constraint $\mu$ which expresses conditions on extensions

- A two-step process:

  Inputs $\langle F_1, \ldots, F_n \rangle \quad \mu \quad \{ F_1', \ldots, F_k' \} \quad \text{extensions}$

Questions:

- How to obtain the extensions?
- How to obtain the AFs?
Rationality Postulates

Postulates adapted from propositional belief merging [Konieczny and Pino Pérez, JLC 2002]

(M0) \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F})) \subseteq \mathcal{A}_\mu^\sigma \)

(M1) If \( \mathcal{A}_\mu^\sigma \neq \emptyset \), then \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F})) \neq \emptyset \)

(M2) If \( \text{Ext}_\sigma(\bigwedge \mathcal{F}) \cap \mathcal{A}_\mu^\sigma \neq \emptyset \), then \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F})) = \text{Ext}_\sigma(\bigwedge \mathcal{F}) \cap \mathcal{A}_\mu^\sigma \)

(M3) If \( \mathcal{F}_1 \equiv \mathcal{F}_2 \) and \( \mu_1 \equiv_\sigma \mu_2 \), then \( \text{Ext}_\sigma(\Delta_{\mu_1}(\mathcal{F}_1)) = \text{Ext}_\sigma(\Delta_{\mu_2}(\mathcal{F}_1)) \)

(M4) If \( \text{Ext}_\sigma(F_1) \subseteq \mathcal{A}_\mu^\sigma \) and \( \text{Ext}_\sigma(F_2) \subseteq \mathcal{A}_\mu^\sigma \), then \( \text{Ext}_\sigma(\Delta_\mu(\langle F_1, F_2 \rangle)) \cap \text{Ext}_\sigma(F_1) \neq \emptyset \) implies \( \text{Ext}_\sigma(\Delta_\mu(\langle F_1, F_2 \rangle)) \cap \text{Ext}_\sigma(F_2) \neq \emptyset \)

(M5) \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_1)) \cap \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_2)) \subseteq \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_1 \cup \mathcal{F}_2)) \)

(M6) If \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_1)) \cap \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_2)) \neq \emptyset \), then \( \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_1 \cup \mathcal{F}_2)) \subseteq \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_1)) \cap \text{Ext}_\sigma(\Delta_\mu(\mathcal{F}_2)) \)

(M7) \( \text{Ext}_\sigma(\Delta_{\mu_1}(\mathcal{F})) \cap \mathcal{A}_{\mu_2}^\sigma \subseteq \text{Ext}_\sigma(\Delta_{\mu_1 \wedge \mu_2}(\mathcal{F})) \)

(M8) If \( \text{Ext}_\sigma(\Delta_{\mu_1}(\mathcal{F})) \cap \mathcal{A}_{\mu_2}^\sigma \neq \emptyset \), then \( \text{Ext}_\sigma(\Delta_{\mu_1 \wedge \mu_2}(\mathcal{F})) \subseteq \text{Ext}_\sigma(\Delta_{\mu_1}(\mathcal{F})) \cap \mathcal{A}_{\mu_2}^\sigma \)
Syncretic Assignment

Mapping from any profile $\mathcal{F}$ to a total pre-order on extensions $\leq_{\mathcal{F}}$ s.t.

1. If $c_1 \in Ext_\sigma(\bigwedge \mathcal{F})$, $c_2 \in Ext_\sigma(\bigwedge \mathcal{F})$, then $c_1 \simeq_{\mathcal{F}} c_2$
2. If $c_1 \in Ext_\sigma(\bigwedge \mathcal{F})$, $c_2 \notin Ext_\sigma(\bigwedge \mathcal{F})$, then $c_1 <_{\mathcal{F}} c_2$
3. $\forall c_1 \in Ext_\sigma(F_1), \exists c_2 \in Ext_\sigma(F_2)$ s.t. $c_2 \leq_{(F_1,F_2)} c_1$
4. If $c_1 \leq_{\mathcal{F}_1} c_2$ and $c_1 \leq_{\mathcal{F}_2} c_2$, then $c_1 \leq_{\mathcal{F}_1 \cup \mathcal{F}_2} c_2$
5. If $c_1 <_{\mathcal{F}_1} c_2$ and $c_1 \leq_{\mathcal{F}_2} c_2$, then $c_1 <_{\mathcal{F}_1 \cup \mathcal{F}_2} c_2$

Theorem

$\Delta$ satisfies (M0)-(M8) iff $Ext_\sigma(\Delta_\mu(\mathcal{F})) = \min(A_\mu^\sigma, \leq_{\mathcal{F}})$
Distance-based Merging

- $d$: distance between sets of arguments (e.g. Hamming distance)
- $\otimes$: aggregation function (e.g. sum)
- $\mathcal{F} \hookrightarrow \leq_{\otimes, d}^{\mathcal{F}}$: syncretic assignment defined by

$$c_1 \leq_{\otimes, d}^{\mathcal{F}} c_2 \text{ iff } \otimes_{F \in \mathcal{F}} d(c_1, \text{Ext}_\sigma(F)) \leq \otimes_{F \in \mathcal{F}} d(c_2, \text{Ext}_\sigma(F))$$
Example of Distance-based Merging

\[
\begin{align*}
&\text{Ext}_{st}(F_1) = \{\{a_1, a_3, a_4\}, \{a_2, a_3, a_4\}\} \\
&\text{Ext}_{st}(F_2) = \{\{a_2, a_4\}\} \\
&\text{Ext}_{st}(F_3) = \{\{a_1, a_2, a_4\}\}
\end{align*}
\]
Example of Distance-based Merging

$\text{Ext}_{st}(F_1) = \{\{a_1, a_3, a_4\}, \{a_2, a_3, a_4\}\}$

$\text{Ext}_{st}(F_2) = \{\{a_2, a_4\}\}$

$\text{Ext}_{st}(F_3) = \{\{a_1, a_2, a_4\}\}$

Goal: merging $\mathcal{F} = \langle F_1, F_2, F_3 \rangle$ with constraint

$\mu = a_2 \wedge a_4 \wedge (a_1 \vee a_3)$

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${a_1, a_2, a_4}$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>${a_2, a_3, a_4}$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>${a_1, a_2, a_3, a_4}$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Example of Distance-based Merging

\[ \text{Goal: merging } F = \langle F_1, F_2, F_3 \rangle \text{ with constraint } \mu = a_2 \land a_4 \land (a_1 \lor a_3) \]

\[ \begin{array}{c|ccc|c}
\mu & F_1 & F_2 & F_3 & \Sigma \\
\hline
\{a_1, a_2, a_4\} & 2 & 1 & 0 & 3 \\
\{a_2, a_3, a_4\} & 0 & 1 & 2 & 3 \\
\{a_1, a_2, a_3, a_4\} & 1 & 2 & 1 & 4 \\
\end{array} \]
Reminder: A Two-Step Process

Inputs \[ \langle F_1, \ldots, F_n \rangle \] \[ \mu \] \rightarrow \text{extensions} \rightarrow \{ F'_1, \ldots, F'_k \}

- Postulates, representation theorem: selection of extensions
- Generation operators: obtaining AFs
Reminder: A Two-Step Process

Inputs
\[ \langle F_1, \ldots, F_n \rangle \]  \[ \mu \]  \implies  \text{extensions}  \implies  \{ F'_1, \ldots, F'_k \}

- Postulates, representation theorem: selection of extensions
- Generation operators: obtaining AFs
Generation of AFs

- Mapping $\mathcal{AF}_\sigma$ from a set of extensions $\mathcal{C}$ to a set of AFs $\mathcal{F}$ s.t. $\text{Ext}_\sigma(\mathcal{F}) = \mathcal{C}$.
- Full merging operator: $\mathcal{AF}_\sigma(\min(\mathcal{A}_\mu^\sigma, \leq_\mathcal{F}))$
- Two policies to handle minimal change
Generation of AFs

- mapping $A\mathcal{F}_\sigma$ from a set of extensions $\mathcal{C}$ to a set of AFs $\mathcal{F}$ s.t. $\text{Ext}_\sigma(\mathcal{F}) = \mathcal{C}$.
- Full merging operator: $A\mathcal{F}_\sigma(\min(\mathcal{A}_\mu^\sigma, \leq_{\mathcal{F}}))$
- Two policies to handle minimal change

| Minimal change of attack, then minimal cardinality | Minimal cardinality, then minimal change of attack |
Example of Generation

Reminder: at the first step, we obtained

$$Ext_{st}(\Delta \sum_{\mu}^{dH}(\langle F_1, F_2, F_3 \rangle)) = \{\{a_1, a_2, a_4\}, \{a_2, a_3, a_4\}\}$$

![Diagram with nodes a1, a2, a3, a4 and edges showing attack, then cardinality and cardinality, then attack paths.]
Resolute Merging: Schematic Explanation

- Is it possible to represent the group’s beliefs by a single AF?
- A two-step process:

\[
\begin{align*}
\text{Inputs} & \quad \langle F_1, \ldots, F_n \rangle \quad \mu \\
\text{Outputs} & \quad \Rightarrow \text{realizable extensions} \quad \Rightarrow \quad F'
\end{align*}
\]

Question:
- Adaption of the first step to obtain realizable extensions?
Resolute Merging: Schematic Explanation

- Is it possible to represent the group’s beliefs by a single AF?
- A two-step process:

\[
\begin{align*}
\text{Inputs} & \quad \langle F_1, \ldots, F_n \rangle \, \mu \\
\Rightarrow & \quad \text{realizable extensions} \\
\Rightarrow & \quad F'
\end{align*}
\]

Question:

- Adaptation of the first step to obtain realizable extensions?
Resolute Merging

- σ-compliant assignment [Diller et al, IJCAI 2015]: pre-order ≤ s.t. for any formula μ, \(\min(\mathcal{A}^\sigma_\mu, \leq)\) is σ-realizable

Good News

A resolute merging operator satisfies the postulates iff there is a σ-compliant syncretic assignment s.t. \(\text{Ext}_\sigma(\Delta_\mu(\mathcal{F})) = \min(\mathcal{A}^\sigma_\mu, \leq)\)
Resolute Merging

- $\sigma$-compliant assignment [Diller et al, IJCAI 2015]: pre-order $\leq$ s.t. for any formula $\mu$, $\min(A^\sigma_\mu, \leq)$ is $\sigma$-realizable

**Good News**
A resolute merging operator satisfies the postulates iff there is a $\sigma$-compliant syncretic assignment s.t. $\Ext_\sigma(\Delta_\mu(F)) = \min(A^\sigma_\mu, \leq)$

**Bad News**
There are no resolute merging operators for stable, preferred, grounded and complete semantics.
Outline

Background Notions
Dung’s AFs
Revising Dung’s AFs

Merging Operators for AFs
Extension-based Merging
From Extensions to AFs
Resolute Merging

Comparison with the Literature
Fusion Operators vs Merging Postulates
Merging Operators vs Aggregation Axioms
Discussion: Attack-based vs Extension-based Merging

Conclusion
FUS\textsubscript{All}, FUS\textsubscript{All\textsc{NT}}, FUS\textsubscript{Maj\textsc{NT}} [Delobelle et al, IJCAI 2015]

\begin{align*}
\text{Inputs} & \quad \langle F_1, \ldots, F_n \rangle \quad \Longrightarrow \quad \text{Weighted AF} \quad \Longrightarrow \quad \text{Outputs} \\
\text{extensions} & \\
\end{align*}

\begin{itemize}
\item no integrity constraint (i.e. $\mu = \top$): (M0),(M7),(M8) trivially satisfied
\end{itemize}

\begin{table}
\centering
\begin{tabular}{|c|c|c|}
\hline
 & FUS\textsubscript{All} & FUS\textsubscript{All\textsc{NT}} & FUS\textsubscript{Maj\textsc{NT}} \\
\hline
(M1) & $\times$ & $\checkmark$ & $\checkmark$ \\
(M2) & $\times$ & $\times$ & $\times$ \\
(M3) & $\checkmark$ & $\checkmark$ & $\checkmark$ \\
(M4) & $\times$ & $\times$ & $\times$ \\
(M5) & $\times$ & $\times$ & $\times$ \\
(M6) & $\times$ & $\times$ & $\times$ \\
\hline
\end{tabular}
\end{table}
Aggregation Axioms [Dunne et al, COMMA 2012; Delobelle et al, IJCAI 2015]

- **Anonymity** aggregation is not sensible to permutations of the profile
- **Non-triviality** the result has at least one non-empty extension
- **Decisiveness** the result has exactly one non-empty extension
- **Unanimity** when agents agree on something, it belongs to the result
- **Majority** when most of the agents agree on something, it belongs to the result
- **Closure** everything in the result is in some part of the input
- **Identity** if all AFs are identical, the result is the initial AF
# Merging Operators vs Aggregation Axioms

<table>
<thead>
<tr>
<th>Properties</th>
<th>$\Sigma, \text{dg}$</th>
<th>$\Sigma, \text{card}$</th>
<th>$\text{Lex, dg}$</th>
<th>$\text{Lex, card}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANON</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\sigma$-SNT/$\sigma$-WNT</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$\sigma$-SD / $\sigma$-WD</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>UA</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$\sigma$-U / $s\sigma$-U</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$c\sigma$-U</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
</tr>
<tr>
<td>MAJ-A</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$\sigma$-MAJ / $c\sigma$-MAJ</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$s\sigma$-MAJ</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CLO / AC / $\sigma$-C</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$c\sigma$-C</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
</tr>
<tr>
<td>$s\sigma$-C</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
<td>✓ $gr$</td>
</tr>
<tr>
<td>ID</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Database and Artificial Intelligence Group**
Discussion: Two Different Philosophies of AF Merging

It is not surprising that

› $FUS_{\text{All}}, FUS_{\text{AllNT}}, FUS_{\text{MajNT}}$ do not satisfy many IC-merging postulates

› our merging operators do not satisfy many aggregation axioms

Both approaches follow different intuitions

<table>
<thead>
<tr>
<th>Operators</th>
<th>$FUS_{\text{All}}, FUS_{\text{AllNT}}, FUS_{\text{MajNT}}$</th>
<th>$\Delta_\mu$-family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Aggregation axioms</td>
<td>IC-Merging Postulates</td>
</tr>
<tr>
<td>Information</td>
<td>Attacks</td>
<td>Extensions</td>
</tr>
</tbody>
</table>
Example


- $\mathcal{F} = \langle F_1, F_2, F_3, F_4, F_5 \rangle$
- Only $c \rightarrow b$ belongs to all AFs
Example

- $\mathcal{F} = \langle F_1, F_2, F_3, F_4, F_5 \rangle$
- Only $c \rightarrow b$ belongs to all AFs
- Result of merging is $F_6$

\[ F_6 = \langle b \leftarrow c \leftarrow a_1, a_2 \rightarrow a_3 \rightarrow a_4 \rangle \]
Example
Extension-based Merging

- $\mathcal{F} = \langle F_1, F_2, F_3, F_4, F_5 \rangle$
- $\{a_1, a_2, a_3, a_4, b\}$ is the single extension for all AFs: must be selected at first step of merging

Since $F_1 \neq F_2$, $F_1$ is the AF closest to the profile.
Example
Extension-based Merging

- $\mathcal{F} = \langle F_1, F_2, F_3, F_4, F_5 \rangle$
- $\{a_1, a_2, a_3, a_4, b\}$ is the single extension for all AFs: must be selected at first step of merging
- Result of merging is $F_1$

Since $F_1 = F_2$, $F_1$ is the AF closest to the profile
Outline

Background Notions
- Dung’s AFs
- Revising Dung’s AFs

Merging Operators for AFs
- Extension-based Merging
- From Extensions to AFs
- Resolute Merging

Comparison with the Literature
- Fusion Operators vs Merging Postulates
- Merging Operators vs Aggregation Axioms
- Discussion: Attack-based vs Extension-based Merging

Conclusion
Conclusion

Summary

- New family of AF merging operators, inspired by extension-based revision [Coste et al, KR 2014]
  - Axiomatic characterization + representation theorem
  - Concrete operators: distance-based merging
- New philosophy of AF merging, orthogonal to attack-based merging

Future works

- Determine resolute merging operators similar to resolute revision operators [Diller et al, IJCAI 2015]
- Study other attack-based approaches [Coste-Marquis et al 2007, Tohmé et al 2008]
- Computational aspects and algorithms design