Intertranslatability of Abstract Argumentation Frameworks

Polberg Sylwia
University College London

Cardiff Argumentation Meeting
July 2016
Roadmap

1 Introduction

2 Background
   - Dung’s Framework
   - Framework with Sets of Attacking Arguments
   - Argumentation Framework with Recursive Attacks
   - Extended Argumentation Frameworks
   - Bipolar Argumentation Frameworks
   - Argumentation Framework with Necessities
   - Evidential Argumentation System
   - Abstract Dialectical Framework

3 Translations
   - Basic Translation
   - Coalition Translation
   - Attack Propagation Translation
   - Defender Translation

4 Summary

References
So let us assume you want to use argumentation in your project and know what sort of features you will need...

### Scenario 1

- You find two argumentation frameworks that are "almost" good, but each one would have to be extended with a missing feature that is present in the other structure
- What do you do?  
  - Create another framework joining the two?  
  - Perhaps find an easy way to simulate the missing feature?
Scenario 2

- You find two argumentation frameworks that have what you want, but you have problem choosing between them...
- ...and what you keep finding are interesting observations about their differences, but no hard facts that you can really use to defend your choice
- What do you do?
  - Go with the general opinion?
  - Throw a dice?
  - Or see what it would take for one framework to emulate the behaviour of the other?
Scenario 3

- You find an argumentation framework that is just right.
- ...but it’s computational complexity is not analyzed and it does not have an implementation.
- What do you do?
  - Find a different framework?
  - Fill in the research gaps yourself?
  - ...or just use a translation?
Scenario 4

- You find an argumentation framework that is just right and work with it
- ...but then a reviewer complains your choice was unnecessary and that “with a bit of effort the Dung’s framework could have handled it”
- What do you do?
  - Talk about your preferences? How the framework is easier to use in your application than Dung’s? Hope he/she will get that?
  - ...or, if it is helpful, show him actual translations, their computational complexity, the impossibility proofs, straight facts that he or she cannot deny?
Scenario 5

- You had to create a new framework to handle what you want...
- ...and now need to explain how it is related to other works in the field
- It would be awesome if you could create:
  - Scenarios handled differently between the frameworks
  - A way for your framework to handle the existing ones
  - The effort it would take for other structures to emulate yours
- How can you come up with such things?
Our Work I

Our motivation

Intertranslatability research can be used in:

- Designing argumentation–based software
- Widening the application and instantiation range of a given framework
- Research of framework dedicated solvers
- Comparing expressive power of given frameworks
- Studying the meaning and the “added value” of framework components
- Establishing the connections between different framework components
Argumentation Frameworks

Abstract argumentation is more than Dung’s framework. There exist many different types (BPW14), including:

- **Attack frameworks:**
  - Dung’s Frameworks (AF) (Dun95)
  - Framework with Sets of Attacking Arguments (SETAF) (NP07)
  - Framework with Recursive Attack (AFRA) (BCGG11)
  - Extended Argumentation Framework (EAF) (Mod09)

- **Support frameworks:**
  - Bipolar Argumentation Framework (BAF) (CLS09; CLS13)
  - Argumentation Framework with Necessities (AFN) (Nou13)
  - Evidential Argumentation System (EAS) (ORL10; PO14)
  - Abstract Dialectical Framework (ADF) (BW10; BES+13; Pol15)
Contributions

New translations

Intertranslatability of Abstract Argumentation Frameworks
Dung’s Abstract Argumentation Framework (AF) is a pair \((A, R)\), where \(A\) is a set of arguments and \(R \subseteq A \times A\) represents the attack relation.
A **Framework with Sets of Attacking Arguments** (SETAF) is a pair \((A, R)\), where \(A\) is a set of arguments and \(R \subseteq (2^A \setminus \emptyset) \times A\) represents the attack relation.

**Example**

![Diagram of argumentation framework with attacks](image)
An argumentation framework with recursive attacks (AFRA) is a pair \((A, R)\) where \(A\) is a set of arguments and \(R\) is a set of attacks, namely pairs \((a, X)\) s.t. \(a \in A\) and \(X \in A \cup R\).
Extended Argumentation Framework (MP10)

**Extended Argumentation Framework**

The extended argumentation framework (EAF) is a tuple \((A, R, D)\), where \(A\) is a set of arguments, \(R \subseteq A \times A\) is the attack relation, \(D \subseteq A \times R\) is the defense attack relation.

**Example**

![Diagram of an extended argumentation framework with nodes a, b, c, d, e, f, g and connections between them representing attacks and defenses.](image-url)
Bipolar Argumentation Frameworks (CLS13)

Bipolar Argumentation Framework

The bipolar argumentation framework (BAF) is a tuple \((A, R, S)\), where \(A\) is a set of arguments, \(R \subseteq A \times A\) represents the attack relation and \(S \subseteq A \times A\) the support relation.

Example

![Diagram of a bipolar argumentation framework with nodes a, b, c, d, and e connected by arrows indicating the attack and support relations.](image-url)
Argumentation Framework with Necessities (Nou13) I

**Argumentation Framework with Necessities**

An **argumentation framework with necessities** is a tuple \((A, R, N)\), where \(A\) is the set of **arguments**, \(R \subseteq A \times A\) represents (binary) **attacks**, and \(N \subseteq (2^A \setminus \emptyset) \times A\) is the **necessity relation**.

**Example**

![Argumentation Framework with Necessities Diagram](image-url)
An **argumentation framework with necessities** is a tuple \((A, R, N)\), where \(A\) is the set of **arguments**, \(R \subseteq A \times A\) represents (binary) **attacks**, and \(N \subseteq (2^A \setminus \emptyset) \times A\) is the **necessity relation**.
An evidential argumentation system (EAS) is a tuple \((A, R, E)\) where \(A\) is a set of arguments, \(R \subseteq (2^A \setminus \emptyset) \times A\) is the attack relation, and \(E \subseteq (2^A \setminus \emptyset) \times A\) is the support relation. We distinguish a special argument \(\eta \in A\) s.t. \(\not\exists (X, y) \in R\) where \(\eta \in X\); and \(\not\exists X\) where \((X, \eta) \in R\) or \((X, \eta) \in E\).
Evidential Argumentation System (ON08; ORL10; PO14)

**Evidential Argumentation System**

An **evidential argumentation system** (EAS) is a tuple \((A, R, E)\) where \(A\) is a set of arguments, \(R \subseteq (2^A \setminus \emptyset) \times A\) is the **attack** relation, and \(E \subseteq (2^A \setminus \emptyset) \times A\) is the **support** relation. We distinguish a special argument \(\eta \in A\) s.t. \(\not\exists (X, y) \in R\) where \(\eta \in X\); and \(\not\exists X\) where \((X, \eta) \in R\) or \((X, \eta) \in E\).

**Example**

![Evidential Argumentation System Diagram](image-url)
Abstract Dialectical Framework (BES$^{+}$13; Pol15) I

**Definition**

An **abstract dialectical framework** (ADF) is a tuple $(S, L, C)$, where:

- $S$ is a set of abstract **arguments** (nodes, statements),
- $L \subseteq S \times S$ is a set of **links** (edges) and
- $C = \{C_s\}_{s \in S}$ is a set of **acceptance conditions**, one condition per each argument.

Important: links now do not represent relations anymore; the precise nature of the interaction between arguments is specified by the acceptance conditions.

**Acceptance conditions**

- They represent the relation of an argument to its parents
- Can be represented as functions $C_s : 2^{par(s)} \rightarrow \{in, out\}$
- More commonly defined as propositional formulas
Abstract Dialectical Framework (BES$^{+}$13; Pol15) II

### Example

![Graph](image)

### Semantics

- Labeling-based (BES$^{+}$13): implemented in DIAMOND (ES13)
- Extension-based (Pol15): four families (AA, CC, AC, CA$_{1}$ and CA$_{2}$), to be implemented
Translation

**Intuition**

“...translation can be understood as a function Tr which maps theories from one formalism into another such that intended models of a theory $\Delta$ from the source formalism are in a certain relation to the intended models of $\text{Tr}(\Delta)$.” ((DW11, 1))

**Abstract Argumentation Translations**

Let $S$, $T$ be two framework types between which we want to translate and $\sigma$, $\delta$ source and target semantics. We distinguish:

- Semantics translations – same framework type, different semantics (DW11)
- **Framework translations** – different framework types, same semantics
- **Normal form translations** – same framework type, same semantics (CK14)
Properties of Translations I

Functional Properties

Look at a translation as a function:

- Can it handle any source framework, or just some subclass?
- Can any target framework be produced, or just some subclass?
- Does it produce same target framework for more than one source framework?
- If yes, what is the relation between the source frameworks?
Properties of Translations II

Complexity Properties

Look at how difficult the translation is:

- Is it purely structural, or does it require computing some basic semantics?
- Is it modular?
- What is the computation time?
- Does it cause any blow up in size of the target framework?

Syntactical Properties

Look at what it does to framework components:

- Does it change the type of arguments or not?
- Does it introduce auxiliary arguments and relations or not?
- Does it remove certain arguments and relations or not?
Properties of Translations III

Semantical Properties

Look at how the semantics of the frameworks behave:

- Is the translations specialized for a particular semantics, or is it generic?
- Is the semantics’ domain the same?
- How strong is the translation?
- Is the translation bijective?
- Does the translation introduce auxiliary arguments in the extensions?

Existing Notions

Typical translation properties include (Got95; Lib14; Jan99):

- Modularity
- Efficiency, polynomiality
- Exactness, faithfulness
Properties of Translations IV

Exactness and Faithfulness

- **Strong translation** – every target extension corresponds to a source one and vice versa
- **Semantics bijective translation** – it’s strong and there is a one to one relation between target and source extensions
- **Faithful translation** – it’s semantics bijective and the original extensions are retrieved by removing auxiliary arguments
- **Exact translation** – it’s semantics bijective and the target extensions are exactly the same as source ones
Possible Approaches

Some translations are **easy** and our target framework can handle everything that the source one does. Some however, **are not**. When one structure possesses a feature the other does not, we can:

- Hide it
- Simulate it
- Remove it
- Limit ourselves to cases in which it does not occur
Translation Approaches II

Translation Types

We can distinguish four main types of translations (BGvdTV09; MBC11; CLS13; CL15; ORL10; PO14; BGvdTV10):

- **Basic** – when going from less to more complex frameworks, usually target framework can handle all elements of the source one
- **Coalition** – from more to less complex structures, not handled elements are hidden away in argument structure
- **Attack Propagation** – from more to less complex structures, effect of not handled elements is simulated by handled ones
- **Defender** – from more to less complex structures, not handled elements are translated into handled ones with the use of auxiliary arguments
Basic Translation I

Basic Translation

- A simple translation, often a generalization
- Never semantical
- On average, it does not require too many auxiliary arguments
- Preserves the structure of the source framework
- Generic, usually preserves all standard semantics in at least faithful manner
Basic Translation II

Example

\[
\begin{align*}
\text{a} & \quad \text{b} \\
\text{b} & \quad \text{a}
\end{align*}
\]
Basic Translation III

Example

\[ a \rightarrow b \]

\[ a \rightarrow b \]

\[ \eta \]

\[ a \rightarrow \neg b \]

\[ a \rightarrow \neg b \]

\[ a \leftarrow \neg b \]

\[ a \leftarrow \neg b \]
Basic Translation IV

Example

- Basic Translation
- Coalition Translation
- Attack Propagation Translation
- Defender Translation

Polberg Sylwia
University College London

Intertranslatability of Abstract Argumentation Frameworks
Basic Translation V

Example

\[
\begin{align*}
\text{Basic Translation V} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
\text{Example} & \\
## Basic Translation: Summary

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>AFRA</th>
<th>SETAF</th>
<th>EAF</th>
<th>BAF</th>
<th>AFN</th>
<th>EAS</th>
<th>ADF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AFRA</td>
<td>✓, ✓</td>
<td>✗</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SETAF</td>
<td>✓, ✓</td>
<td>✗</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EAF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓, ✓</td>
<td>✓, ✓</td>
<td>✓, ✓</td>
</tr>
<tr>
<td>BAF</td>
<td>✓, ✓</td>
<td></td>
<td>✗</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AFN</td>
<td>✓</td>
<td></td>
<td></td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓, ✓</td>
<td>✓, ✓</td>
</tr>
<tr>
<td>EAS</td>
<td></td>
<td>✓</td>
<td>✓, ✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓, ✓</td>
</tr>
<tr>
<td>ADF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>

✓ – translation  
✓ – hybrid translation  
✓ – subclass translation
Coalition translation

- Arguments in the target framework are collections of source arguments that are tied by support or ability to carry out a group attack
- Almost always semantical
- Exponential increase in amount of required auxiliary arguments
- Translation is lossy, it removes arguments from the source framework
- Usually preserves most of the standard semantics in a strong to semantics bijective manner
Coalition Translation II

Example

\[
\begin{align*}
\{\eta, a\} & \quad \{\eta, a, b\} & \quad \{\eta, a, c, d\} \\
\{\eta\} & \quad \{\eta, a, c\} & \quad \{\eta, a, b, c, d\}
\end{align*}
\]
Coalition Translation III

Example

- \{\eta, a\}
- \{\eta, a, b\}
- \{\eta, a, c\}
- \{\eta, a, b, c\}
- \{\eta, a, b, c, d\}
- \{\eta, a, b, c\}
- \{\eta, a, b, c, d\}

Graph:

- Nodes: \eta, a, b, c, d, e
- Edges: \{\eta, a\} -> a, \{\eta, a, b\} -> b, \{\eta, a, c\} -> c, \{\eta, a, b, c\} -> d, \{\eta, a, b, c, d\} -> e

Polberg Sylwia
University College London

Intertranslatability of Abstract Argumentation Frameworks
## Coalition Translation: Summary

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>AFRA</th>
<th>SETAF</th>
<th>EAF</th>
<th>BAF</th>
<th>AFN</th>
<th>EAS</th>
<th>ADF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFRA</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETAF</td>
<td>✓</td>
<td></td>
<td>x</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EAF</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAF</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFN</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>✓, ✓, ✓, ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

✓ – translation  
✓ – hybrid translation
### Attack Propagation Translation 1

<table>
<thead>
<tr>
<th><strong>Attack propagation translation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Completes the source framework with various types of indirect attacks</td>
</tr>
<tr>
<td>Does not require auxiliary arguments</td>
</tr>
<tr>
<td>Removes some of the arguments in the source framework</td>
</tr>
<tr>
<td>In principle, the translation is semantical; can be structural only for particular normal forms</td>
</tr>
<tr>
<td>Preserves completeness–based semantics in an exact manner</td>
</tr>
</tbody>
</table>
**Example**

A graph showing relationships between nodes labeled a, b, c, d, e, f, and g. The nodes are connected by arrows indicating the direction of interaction or influence. The graph illustrates the concept of attack propagation within an abstract argumentation framework.
### Attack Propagation: Summary

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>AFRA</th>
<th>SETAF</th>
<th>EAF</th>
<th>BAF</th>
<th>AFN</th>
<th>EAS</th>
<th>ADF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFRA</td>
<td>✓</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETAF</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAF</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAF</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFN</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✗</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>

✓ – translation
✓ – hybrid translation
Defender Translation

Defender translation

- Exploit defense to e.g. simulate support or to connect arguments
- Does not remove arguments from the source framework
- For attack–based frameworks, the translation:
  - is structural
  - can require exponentially many auxiliary arguments
- For support–based frameworks:
  - it is semantical and can be structural only for particular normal forms
  - requires polynomially many auxiliary arguments
- Usually preserves semantics that are at least admissible in a strong to faithful manner
Defender Translation

Example

\[ (\{x_1, x_2, x_3\}, y) \rightarrow y \]
Defender Translation

Example

\[ (\{x_1, x_2, x_3\}, y) \]
Defender Translation

Example

```
   a
  / \  \\
 d   b   c  e
   \ /   \   \
    b   a'  \\
   /     /   \
 d   b   c  e

Intertranslatability of Abstract Argumentation Frameworks
```
Defender Translation

Example

![Diagram showing a before and after defender translation process with nodes a, b, c, d, and e. The diagram illustrates the intertranslatability of Abstract Argumentation Frameworks.](image-url)
Defender Translation

Example

\[ \text{Diagram showing relationships between nodes a, b, c, a', b', a, b} \]
Defender Translation

Example

- Basic Translation
- Coalition Translation
- Attack Propagation Translation
- Defender Translation

Intertranslatability of Abstract Argumentation Frameworks

Polberg Sylwia
University College London
Defender Translation

Example

[Diagram of abstract argumentation frameworks showing intertranslatability between basic, coalition, and attack propagation translations.]
### Defender Translation: Summary

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>AFRA</th>
<th>SETAF</th>
<th>EAF</th>
<th>BAF</th>
<th>AFN</th>
<th>EAS</th>
<th>ADF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFRA</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETAF</td>
<td>✓</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAF</td>
<td>✓</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAF</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFN</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>✓ ,✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

✓ – translation
✓ – hybrid translation
## Improving Translations

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>AFRA</th>
<th>SETAF</th>
<th>EAF</th>
<th>BAF</th>
<th>AFN</th>
<th>EAS</th>
<th>ADF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>AFRA</td>
<td>✓</td>
<td>x</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETAF</td>
<td>X</td>
<td>x</td>
<td></td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>EAF</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>BAF</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFN</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>x</td>
<td>X</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAS</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>X</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X?</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

✓ – is exact
✓ – exact might exist
X – exact most likely does not exist
this is how I finish a presentation:

happymonsters.tumblr.com

Soo...
Uhh
...yeah.


Guido Boella, Dov Gabbay, Leendert van der Torre, and Serena Villata.  
Support in abstract argumentation.  

Gerhard Brewka, Sylwia Polberg, and Stefan Woltran.  
Generalizations of Dung frameworks and their role in formal argumentation.  

Gerhard Brewka and Stefan Woltran.  
Abstract dialectical frameworks.  
Cosmina Croitoru and Timo Kötzing.
A normal form for argumentation frameworks.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex.
An axiomatic approach to support in argumentation.
Claudette Cayrol and Marie-Christine Lagasquie-Schiex.
Bipolar abstract argumentation systems.
In Guillermo Simari and Iyad Rahwan, editors, Argumentation in Artificial Intelligence, pages 65–84. 2009.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex.
Bipolarity in argumentation graphs: Towards a better understanding.

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

Wolfgang Dvořák and Stefan Woltran.
On the intertranslatability of argumentation semantics.
Stefan Ellmauthaler and Hannes Strass.  
The DIAMOND system for argumentation: Preliminary report.  
In Michael Fink and Yuliya Lierler, editors, *Proc. ASPOCP*, 2013.

Georg Gottlob.  
Translating default logic into standard autoepistemic logic.  

Tomi Janhunen.  
On the intertranslatability of non–monotonic logics.  

Paolo Liberatore.  
Bijective faithful translations among default logics.  
Sanjay Modgil and Trevor J. M. Bench-Capon.
Metalevel argumentation.

Sanjay Modgil.
Reasoning about preferences in argumentation frameworks.

Sanjay Modgil and Henry Prakken.
Reasoning about preferences in structured extended argumentation frameworks.
Farid Nouioua.
AFs with necessities: Further semantics and labelling characterization.

Søren Nielsen and Simon Parsons.
A generalization of Dung’s abstract framework for argumentation: Arguing with sets of attacking arguments.

Nir Oren and Timothy J. Norman.
Semantics for evidence-based argumentation.
Nir Oren, Chris Reed, and Michael Luck.  
Moving between argumentation frameworks.  

Sylwia Polberg and Nir Oren.  
Revisiting support in abstract argumentation systems.  

Sylwia Polberg.  
Revisiting extension–based semantics of abstract dialectical frameworks.  