Two-Agent Conflict Resolution with Assumption-Based Argumentation

Xiuyi Fan, Francesca Toni, Adil Hussain
Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK

Abstract. Conflicts exist in multi-agent systems. Agents have different interests and desires. Agents also hold different beliefs and may make different assumptions. To resolve conflicts, agents need to better convey information between each other and facilitate fair negotiations that yield jointly agreeable outcomes. In this paper, we present a two-agent conflict resolution scheme developed under Assumption-Based Argumentation (ABA). Agents represent their beliefs and desires in ABA. Conflicts are resolved by merging conflicting arguments. We also discuss the notion of fairness and the use of argumentation dialogue in conflict resolution.

Keywords. Multi-Agent Systems, Argumentation, Conflict Resolution

1. Introduction

Complex multi-agent systems are composed of heterogeneous agents with different beliefs and desires. Agents usually perform tasks in a joint manner so as to promote high common welfare. In this paper, we consider conflicts arising for two reasons during a collaboration among two agents. Firstly, agents reason with different assumptions to fill gaps in their beliefs. Since some assumptions may be incorrect, agents may be misinformed and decide on incompatible actions that lead to conflict. Secondly, even if agents share the same information, they may still disagree if they have different desires.

We use Assumption-Based Argumentation (ABA) [BTK93,BDKT97,DKT09] for conflict resolution, as beliefs and desires can be represented in ABA. In this setting, to resolve conflicts between two agents is to merge two ABA frameworks because the merge eliminates misunderstanding between agents and considers desires from both agents. To eliminate misunderstanding, an agent’s assumptions are updated by considering beliefs from the other agent. Therefore, incorrect assumptions are invalidated. To satisfy desires from both agents, we propose the mechanism of concatenation to merge rules. Upon a successful concatenation merge, both agents’ desires may be satisfied. We also briefly consider cases where a concatenation merge is not possible, and define fair compromise for these cases.

Research merits of this paper are: (1) the partition of rules in ABA in belief rules and desire rules, (2) the definition of a merge operator for rules about desires to resolve conflicts between agents, (3) the consideration of fairness for cases where not all agents’ desires are satisfiable, and (4) the exploration of the use of argumentation dialogues to resolve conflicts.
The paper is organized as follows. Section 2 reviews the ABA framework, serving as the ground for our work. Section 3 presents a number of examples that motivate our approach. Section 4 describes the merge operator. Section 5 explains how conflict resolution can be performed in a distributed and progressive manner. It further explores the concept of fairness as making similar compromise between agents. Section 6 reviews some related work in conflict resolution among agents. We conclude at section 7 with a summary and a discussion about possible future directions.

2. Assumption-Based Argumentation (ABA)

An ABA framework \([\mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C}]\) is a tuple \(\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle\) where

- \(\langle \mathcal{L}, \mathcal{R} \rangle\) is a deductive system, with a language \(\mathcal{L}\) and a set of inference rules \(\mathcal{R}\) of the form \(s_0 \leftarrow s_1, \ldots, s_m (m \geq 0)\),
- \(\mathcal{A} \subseteq \mathcal{L}\) is a (non-empty) set, whose elements are referred to as assumptions,
- \(\mathcal{C}\) is a total mapping from \(\mathcal{A}\) into \(2^\mathcal{L}\), where each \(c \in \mathcal{C}(\alpha)\) is a contrary of \(\alpha\).

Given a rule \(s_0 \leftarrow s_1, \ldots, s_m, s_0\) is referred as the head and \(s_1, \ldots, s_m\) as the body of the rule. We will use the following notation: \(\text{Head}(s_0 \leftarrow s_1, \ldots, s_m) = s_0\) and \(\text{Body}(s_0 \leftarrow s_1, \ldots, s_m) = s_1, \ldots, s_m\).

Basically, ABA frameworks can be defined for any logic specified by means of inference rules. Sentences in the underlying language include assumptions. Arguments are deductions of claims supported by sets of assumptions. Attacks are directed at the assumptions in the support of arguments.

In an ABA framework \(\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle\), an argument can be formally defined in terms of \(\mathcal{R}\) and \(\mathcal{A}\) in several ways \([BDKT97, DKT06, DKT09]\). Here, we adopt the definition of \([DKT09]\): an argument for (the claim) \(c \in \mathcal{L}\) supported by \(S \subseteq \mathcal{A}\) (\(S \vdash c\) in short) is a tree with nodes labelled by sentences in \(\mathcal{L}\) or by the symbol \(\tau\) such that

1. the root is labelled by \(c\);
2. for every node \(N\)
   - if \(N\) is a leaf then \(N\) is labelled either by an assumption or by \(\tau\);
   - if \(N\) is not a leaf and \(l_N\) is the label of \(N\), then there is an inference rule \(l_N \leftarrow b_1, \ldots, b_m (m \geq 0)\) and either \(m = 0\) and the child of \(N\) is \(\tau\) or \(m > 0\) and \(N\) has \(m\) children, labelled by \(b_1, \ldots, b_m\) (respectively);
3. \(S\) is the set of all assumptions labelling the leaves.

The notion of attack in ABA is formally defined as follows:

- an argument \(S_1 \vdash c_1\) attacks an argument \(S_2 \vdash c_2\) iff the claim \(c_1\) of the first argument is a contrary of one of the assumptions in the support \(S_2\) of the second argument \(\exists \alpha \in S_2\) such that \(c_1 \in \mathcal{C}(\alpha)\);
- a set of arguments \(\text{Arg}_1\) attacks a set of arguments \(\text{Arg}_2\) iff an argument in \(\text{Arg}_1\) attacks an argument in \(\text{Arg}_2\).

\(^1\)Here, as in \([GT08]\), we define the contrary of an assumption as a total mapping from an assumption to a set of sentences, instead of a mapping from an assumption to a sentence as in the original ABA \([BDKT97]\).

\(^2\)Here, as in \([DKT09]\), \(\tau \notin \mathcal{L}\) intuitively represents “true”.
Attacks between (sets of) arguments correspond in ABA to attacks between sets of assumptions:

- A set of assumptions $A$ attacks a set of assumptions $B$ iff an argument supported by a subset of $A$ attacks an argument supported by a subset of $B$.

With argument and attack defined, commonly used argumentation semantics such as conflict-free, admissible, complete, grounded, preferred, ideal and stable semantics can be applied in ABA [BTK93,BDKT97,DKT06,DMT07]. In this work, we focus on the conflict-free semantics, defined as follows:

- A set of assumptions is conflict-free iff it does not attack itself;
- A set $\text{Args}$ of arguments is conflict-free iff the union of all sets of assumptions that support arguments in $\text{Args}$ does not attack itself.

Throughout the paper, we use the following terminology: a sentence $s$ belongs to a conflict-free extension with respect to an argument $\text{arg}$ iff $s$ is the claim of $\text{arg}$ and $\text{arg}$ belongs to a conflict-free set of arguments.

3. Examples

Imagine a scenario such as the following. Two agents, Jenny and Amy, are planning a film night together. They want to decide the movie to watch.

**Example 3.1.** Jenny wants to pick a movie that is entertaining. She finds action movies entertaining. Spider Man and Terminator are both screening at the moment. Jenny believes Terminator is an action movie. She does not have much information about Spider Man. Hence she believes it is not an action movie. Jenny then concludes she wants to watch Terminator. Represented in ABA, Jenny’s beliefs and desires are\(^3\):

**Rules:**

\begin{align*}
\text{watchMovie}(X) & \leftarrow \text{selectMovie}(X), \text{entertainingMovie}(X) \\
\text{entertainingMovie}(X) & \leftarrow \text{actionMovie}(X) \\
\text{actionMovie}(\text{Terminator}) &
\end{align*}

**Assumptions:**

\begin{align*}
\text{selectMovie}(X) & \\
\neg \text{actionMovie}(X) &
\end{align*}

**Contraries:**

\begin{align*}
C(\text{selectMovie}(X)) &= \{ \neg \text{selectMovie}(X), \text{selectMovie}(Y) | Y \neq X \} \\
C(\neg \text{actionMovie}(X)) &= \{ \text{actionMovie}(X) \}
\end{align*}

Here, Jenny’s goal $\text{watch}(\text{Terminator})$ is supported by the argument in Figure 1 and belongs to a conflict-free extension.

\(^3\)In presenting ABA frameworks we use the following conventions:

- constants are in *italic*;
- $X, Y$ etc. are variables; inference rule/assumptions/contrary schemata (with variables) are used to stand for the set of all their ground instances with respect to constants. E.g., $\text{selectMovie}(X)$ stands for $\text{selectMovie}(\text{Terminator})$ and $\text{selectMovie}(\text{Spider Man})$;
- $s \leftarrow$ is represented simply as $s$, for any sentence $s \in \mathcal{L}$. 


Amy also wants to watch an action movie that is entertaining. However, she has watched *Terminator* before hence does not want to watch it again. Amy has watched the trailer of *Spider Man* and believes it also is an action movie. Amy concludes she wants to watch *Spider Man*. After exchanging their preferences and reasoning, Amy realized that Jenny mistakenly held an incorrect assumption, as *Spider Man* is an action movie. Jenny hence learns this new information and agrees to watch *Spider Man* with Amy. In ABA, Amy’s beliefs and desires are:

**Rules:**
- watchMovie(X) ← selectMovie(X), entertainingMovie(X)
- entertainingMovie(X) ← actionMovie(X)
- ¬selectMovie(Terminator)
- actionMovie(Spider Man)

**Assumptions and Contraries:**
as for Jenny

In this example, agreement can be reached simply by information exchange of actionMovie(Spider Man), resulting in a merged ABA framework with:

- watchMovie(Spider Man) ← selectMovie(Spider Man), entertainingMovie(Spider Man)
- actionMovie(Spider Man) ← actionMovie(Spider Man)
- actionMovie(Spider Man)

**Example 3.2.** Similarly to the previous example, Jenny wants to watch an action movie that is entertaining. She believes two films, *Terminator* and *Harry Potter*, are screening at the moment. She believes *Harry Potter* is a fantasy movie and *Terminator* is an action movie. She then decides to watch *Terminator*. In ABA, Jenny’s beliefs and desires are:

**Rules:**
- watchMovie(X) ← selectMovie(X), entertainingMovie(X)
- entertainingMovie(X) ← actionMovie(X)
- actionMovie(Terminator)
- fantasyMovie(Harry Potter)

**Assumptions:**
- selectMovie(X)
- ¬actionMovie(X)
\neg \text{fantasyMovie}(X)

**Contraries:**

\[ C(\neg \text{selectMovie}(X)) = \{ \neg \text{selectMovie}(X), \text{selectMovie}(Y)\mid Y \neq X \} \]

\[ C(\neg \text{actionMovie}(X)) = \{ \text{actionMovie}(X) \} \]

\[ C(\neg \text{fantasyMovie}(X)) = \{ \text{fantasyMovie}(X) \} \]

Instead of watching an action movie, Amy wants to watch an entertaining fantasy movie. In addition to the two movies Jenny knows of, Amy also knows that *Lord of the Rings* (LoR in short), an action and a fantasy movie, is screening. She is uncertain between *Harry Potter* and *LoR*. After exchanging information, Jenny becomes aware of the extra piece of information about *LoR* and both agree to watch this movie. In ABA, Amy’s beliefs and desires are:

**Rules:**

\[
\text{watchMovie}(X) \leftarrow \text{selectMovie}(X), \text{entertainingMovie}(X) \\
\text{entertainingMovie}(X) \leftarrow \text{fantasyMovie}(X) \\
\text{actionMovie}(\text{Terminator}) \\
\text{fantasyMovie}(\text{Harry Potter}) \\
\text{fantasyMovie}(\text{LoR}) \\
\text{actionMovie}(\text{LoR})
\]

**Assumptions and Contraries:**

as for Jenny

In this example, agreement is reached by information exchange of actionMovie(LoR) and the creation of a new rule that a movie is entertaining if it is both an action movie and a fantasy movie. The resulting merged ABA framework contains:

\[
\text{watchMovie}(\text{LoR}) \leftarrow \text{selectMovie}(\text{LoR}), \text{entertainingMovie}(\text{LoR}) \\
\text{entertainingMovie}(\text{LoR}) \leftarrow \text{actionMovie}(\text{LoR}), \text{fantasyMovie}(\text{LoR}) \\
\text{actionMovie}(\text{LoR}) \\
\text{fantasyMovie}(\text{LoR})
\]

Figure 2 gives the argument for watching *LoR* in the merged framework.

These examples demonstrate the two main aspects of our approach: (1) beliefs can be communicated to update incorrect assumptions; (2) when there is a conflict between agents’ desires, we try to reach an agreement satisfying both agents by merging rules.
4. Conflict Resolution as Merge of Beliefs

Inspired by the examples presented in section 3, we resolve conflicts between two agents with a merge of conflicting arguments/ABA frameworks. Conceptually, conflict resolution can be understood as follows. Two agents share the same high level goal, \( G \). Each agent has its own mean, \( \delta_i \), to realize \( G \). Hence \( G \) is realizable by each agent independently. Conflict arises as the two means, \( \delta_1 \) and \( \delta_2 \), of deriving \( G \) are different. Resolving conflict between two agents means detecting a commonly acceptable mean, \( \delta \), to realize \( G \).

As an illustration, in example 3.1 \( G \) is watchMovie(X) (where X is implicitly existentially quantified), \( \delta_1 = \{X/\text{Terminator}\} \), \( \delta_2 = \{X/\text{Spider Man}\} \) and \( \delta = \{X/\text{Spider Man}\} \) (namely, realizations are variable instantiations). Our approach is to construct a joint ABA framework by extracting information from both agents’ ABA frameworks. The joint ABA framework supports a conflict-free realization \( G \delta \) of \( G \).

4.1. Belief Rules vs. Desire Rules

Before we formally define conflict resolution, there is one classification we need to make. When describing agents with an ABA framework \( \langle L, R, A, C \rangle \), there are two types of rules in \( R \): (1) belief rules, \( BR \), and (2) desire rules, \( DR \). In previous examples,

\[
\begin{align*}
\text{actionMovie(Terminator)} \quad & \text{belief rules} \\
\text{fantasyMovie(Harry Potter)} \quad & \text{belief rules}
\end{align*}
\]

are belief rules. Agents with different desires can be modelled by different desire rules. In previous examples,

\[
\begin{align*}
\text{entertainingMovie(X) \leftarrow actionMovie(X)} \quad & \text{desire rules} \\
\text{entertainingMovie(X) \leftarrow fantasyMovie(X)} \quad & \text{desire rules}
\end{align*}
\]

are desire rules. Intuitively, desire rules are directly relevant to goals and can be used to build arguments for goals. Instead, belief rules may contribute to undermining arguments for goals.

Note that rules may be defeasible (e.g. an agent desire to select and watch a movie that is entertaining) or not (e.g. an agent believes an action movie is entertaining). Defeasibility, as conventional in ABA, is given by the presence of assumptions [DKT09].

4.2. Formal Definition of Conflict and Conflict Resolution

Formally, we have \( Agent_1 \) with ABA framework \( AF_1 = \langle L, R_1, A_1, C_1 \rangle \) and \( Agent_2 \) with ABA framework \( AF_2 = \langle L, R_2, A_2, C_2 \rangle \). In each framework, the set of rules \( R_i \) is composed of belief rules \( BR_i \) and desire rules \( DR_i \), such that \( R_i = BR_i \cup DR_i \), and \( BR_i \cap DR_i = \emptyset \). The agents share the same language \( L \) while having potentially different rules, assumptions and contraries of assumptions. The agents share the same goal \( G \) while having different instantiations \( G \delta_1 \) and \( G \delta_2 \) thereof.

**Definition 4.1.** Conflict. A conflict between \( Agent_1 \) and \( Agent_2 \) with ABA frameworks \( AF_1 \) and \( AF_2 \) (respectively) with respect to a goal \( G \) is a pair of goals \( (G \delta_1, G \delta_2) \) such that \( G \delta_i \in L \), \( G \delta_1 \) belongs to a conflict-free extension of \( AF_i \) (\( i = 1, 2 \)) and \( G \delta_1 \neq G \delta_2 \).
Definition 4.2. Conflict Resolution. A conflict resolution \( G \delta \) for a conflict \((G \delta_1, G \delta_2)\) with respect to \(G\) is such that \(G \delta\) belongs to conflict-free extensions of \(AF'_1\) and \(AF'_2\) defined as \(AF'_1 = \langle L, R_1 \cup BR_2, A, C \rangle\) and \(AF'_2 = \langle L, R_2 \cup BR_1, A, C \rangle\), where

- \(A = A_1 \cup A_2\);
- \(C\) is defined as \(C(\alpha) = C_1(\alpha) \cup C_2(\alpha)\) (for any \(\alpha \in A\)).

4.3. The Merge Operator

The merge of two ABA frameworks, \(AF_1\) and \(AF_2\), can be reduced to the merge of the two sets of rules, \(R_1\) and \(R_2\), sets of assumptions, \(A_1\) and \(A_2\), and contrary mappings, \(C_1\) and \(C_2\). We define the merge of two sets of assumptions as their union. Since ABA does not have any requirement on assumptions being consistent, the union of two sets of assumptions to produce another set is always a valid operation.

Then we define the merge of two contrary mappings as the union of the two individual contrary mappings. This can be justified as follows: if any of the agents knows a sentence is contrary with an assumption, then jointly both agents know the sentence is contrary with the assumption.

The merge of the rules goes one step beyond a simple union. Recall Example 3.2, where Agent 1 has the rule:

entertainingMovie(X) ← actionMovie(X)

whereas Agent 2 has a different rule with the same head:

entertainingMovie(X) ← fantasyMovie(X)

The final merged rule is:

entertainingMovie(X) ← actionMovie(X), fantasyMovie(X)

This merge corresponds with our intuition. It suggests that when two rules are used to describe agents’ desires, the resolution is to create a win-win situation by satisfying both agents’ desires at the same time. We define this as the concatenation merge.

Definition 4.3. Concatenation Merge. The merge operator between two ABA frameworks, \(AF_1 = \langle L, R_1, A_1, C_1 \rangle\) and \(AF_2 = \langle L, R_2, A_2, C_2 \rangle\), is defined as \(AF_1 \oplus AF_2 = AF = \langle L, R, A, C \rangle\), such that:

- \(A = A_1 \cup A_2\)
- \(C(\alpha) = C_1(\alpha) \cup C_2(\alpha)\)
- \(R = R_1 \oplus_R R_2\), where \(\oplus_R\) is defined as follows, given that \(DR_1 = \{r_1^1, r_1^2, \ldots, r_1^n\}\) and \(DR_2 = \{r_2^1, r_2^2, \ldots, r_2^m\}\) (\(n, m \geq 0\)):
  - \(R = BR \cup DR\) where
    - \(BR = BR_1 \cup BR_2\)
    - \(DR\) is such that, for \(i = 1, \ldots, n, j = 1, \ldots, m:\)
      - if \(Head(r_1^i) \neq Head(r_2^j)\) then \(r_1^i, r_2^j \in DR\);
      - if \(r_1^i = r_2^j = r\) then \(r \in DR\);
      - if \(Head(r_1^i) = Head(r_2^j) = h\) then \(r \in DR\), where \(Head(r) = h\) and \(Body(r) = Body(r_1^i), Body(r_2^j)\);
We refer to \( AF \) as the merged ABA framework (resulting from \( AF_1 \) and \( AF_2 \)).

**Theorem 4.1.** When there exists a conflict \( (\mathcal{G}\delta_1, \mathcal{G}\delta_2) \) between \( Agent_1 \) and \( Agent_2 \) with respect to some goal \( \mathcal{G} \), if

- \( \mathcal{G}\delta \) belongs to a conflict-extension of \( AF = AF_1 \ominus AF_2 \) with respect to an argument \( S_i \vdash \mathcal{G}\delta \), and
- there exist arguments \( S_1 \vdash \mathcal{G}\delta, S_2 \vdash \mathcal{G}\delta \) in \( AF'_1 \) and \( AF'_2 \), respectively, such that \( S_i \subseteq S \) \((i = 1, 2)\)

then \( \mathcal{G}\delta \) is a conflict resolution for \( (\mathcal{G}\delta_1, \mathcal{G}\delta_2) \) with respect to \( \mathcal{G} \).

**Proof.** We know that \( \mathcal{G}\delta \) is the claim of \( S_1 \vdash \mathcal{G}\delta \) in \( AF'_1 \) hence \( \mathcal{G}\delta \) is supported by \( S_1 \) in \( AF'_1 \). We also know \( S_i \subseteq S \) and \( S \) is conflict-free in \( AF \). We need to prove that \( S_i \) is conflict-free in \( AF'_i \). By contradiction, suppose \( S_i \) is not conflict-free in \( AF'_i \), and there exists \( A \subseteq S_i \) and \( A \vdash \alpha \) with respect to \( AF'_1 \) such that \( c = C(\alpha) \) for some \( \alpha \in S_i \). This implies that \( A \vdash \alpha \) attacks \( \alpha \) in \( AF \), since \( AF \) contains the union of all belief rules in \( AF'_1 \) and \( AF'_2 \). Therefore \( S \) is not conflict-free in \( AF \): contradiction.

It is worth noticing that the reverse of this theorem does not hold. Hence, if a conflict resolution \( \mathcal{G}\delta \) exists with respect to \( \mathcal{G} \), it may not the case that \( \mathcal{G}\delta \) belongs to a conflict-free extension of the merged \( AF \), as demonstrated by the following example.

**Example 4.1.** Assume the conflict is \((p(2), p(3))\), for some constants 2 and 3. The ABA frameworks are given below (here, all rules are desire rules):

<table>
<thead>
<tr>
<th>Rules: ( AF_1 ):</th>
<th>( AF_2 ):</th>
<th>( AF \ = AF_1 \ominus AF_2 ):</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(X) \leftarrow a(X) )</td>
<td>( p(X) \leftarrow q(X) )</td>
<td>( p(X) \leftarrow a(X), q(X) )</td>
</tr>
<tr>
<td>( q(X) \leftarrow b(X) )</td>
<td>( q(X) \leftarrow b(X) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assumptions:</th>
<th>( a(1); a(2) )</th>
<th>( b(1); b(3) )</th>
<th>( a(1); a(2) b(1); b(3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contraries:</td>
<td>( C(a(X)) = {q(X)} )</td>
<td>( C(b(X)) = {r(X)} )</td>
<td>( C(a(X)) = {q(X)} )</td>
</tr>
<tr>
<td>( C(b(X)) = {r(X)} )</td>
<td>( C(a(X)) = {q(X)} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Then, \( p(1) \) is a conflict resolution as \([a(1)]\) is conflict-free and supports \( p(1) \) in \( AF_1 = AF'_1 \) and \([b(1)]\) is conflict-free and supports \( p(1) \) in \( AF_2 = AF'_2 \). However, \( p(1) \) does not belong to a conflict-free extension in \( AF \), since \( p(X) \leftarrow a(X), q(X) \) is in \( AF \) and \( q(X) \) is the contrary of \( a(X) \). So \( p(X) \) cannot belong to a conflict-free extension in \( AF \).

It is also worth noticing that Theorem 4.1 holds under the condition that \( \mathcal{G}\delta \) has to exist in both \( AF'_1 \) and \( AF'_2 \). The follow counter-example justifies this condition.

**Example 4.2.** Assume \((\mathcal{G}\delta_1, \mathcal{G}\delta_2)\) is the conflict and (again, all rules are desire rules):

<table>
<thead>
<tr>
<th>Rules:</th>
<th>( AF_1 ):</th>
<th>( AF_2 ):</th>
<th>( AF \ = AF_1 \ominus AF_2 ):</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{G}\delta \leftarrow a )</td>
<td>( \mathcal{G}\delta \leftarrow c )</td>
<td>( \mathcal{G}\delta \leftarrow a; \mathcal{G}\delta \leftarrow b )</td>
<td></td>
</tr>
<tr>
<td>( \mathcal{G}\delta_1 \leftarrow b )</td>
<td>( \mathcal{G}\delta_2 \leftarrow c )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assumptions:</th>
<th>( a; b )</th>
<th>( c )</th>
<th>( a; b; c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contraries:</td>
<td>( C(a) = {w} )</td>
<td>( C(c) = {w} )</td>
<td>( C(a) = {w}; C(b) = {w} )</td>
</tr>
<tr>
<td>( C(b) = {w} )</td>
<td>( C(c) = {w} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Then, \([a]\), which supports \( \mathcal{G}\delta \), belongs to a conflict-free extension of \( AF \). However, since there is no argument for \( \mathcal{G}\delta \) in \( AF_1 = AF'_1 \), there is no conflict-free set of assumptions that supports \( \mathcal{G}\delta \). Hence, \( \mathcal{G}\delta \) is not a conflict resolution for \((\mathcal{G}\delta_1, \mathcal{G}\delta_2)\).
5. Discussion

5.1. Relevance of Arguments and Interactive Argumentation

The merge operation presented in Section 4 is a generic solution. It solves conflicts between two argumentation frameworks by statically merging them. While performing conflict resolution dynamically between two agents, we can adopt a goal-driven approach by performing the merge interactively and progressively from the claim. Conflict resolution with ABA can be performed interactively. Unlike some other argumentation frameworks, ABA ensures all arguments that support a claim are relevant [DKT09]. While resolving conflicts, each agent only discloses beliefs and desires that are relevant to the claim.

A distributed dispute derivation can be conducted by agents presenting beliefs and desires that support the claim interactively and progressively. Such derivation can be carried out in a top-down fashion. For instance, Example 3.2 can be modelled as follows:

**Step 1:**

**Jenny:** watchMovie(X) ← selectMovie(X), entertainingMovie(X)

**Amy:** watchMovie(X) ← selectMovie(X), entertainingMovie(X)

**Joint:** watchMovie(X) ← selectMovie(X), entertainingMovie(X)

**Step 2:**

**Jenny:** entertainingMovie(X) ← actionMovie(X)

**Amy:** entertainingMovie(X) ← fantasyMovie(X)

**Joint:** watchMovie(X) ← selectMovie(X), entertainingMovie(X)

entertainingMovie(X) ← actionMovie(X), fantasyMovie(X)

**Step 3:**

**Jenny:** actionMovie(Terminator)

fantasyMovie(Harry Potter)

**Amy:** actionMovie(Terminator)

fantasyMovie(Harry Potter)

fantasyMovie(LoR)

actionMovie(LoR)

**Joint:** watchMovie(X) ← selectMovie(X), entertainingMovie(X)

entertainingMovie(X) ← actionMovie(X), fantasyMovie(X)

actionMovie(Terminator)

fantasyMovie(Harry Potter)

fantasyMovie(LoR)

actionMovie(LoR)

This example shows exchange of rules. Assumptions and contraries of assumptions can be exchanged in a similar manner. Communicating information progressively has the advantage that only necessary and sufficient information is disclosed. It hence avoids possible unnecessary disclosure and computuation. Agents have the freedom to control the type and content of the information they disclose. This feature is useful to address privacy concerns that agents may have.

5.2. On Failure Cases of the Concatenation Merge

The concatenation merge fails in certain cases. In particular, the body of rules to be concatenated may be in conflict with each other (as in Example 4.2). Furthermore, there may
be no jointly satisfactory argument (as in Example 3.2 without the constant \(LoR\)). In these cases, only “biased” solution may exist, e.g., either watch Harry Potter or Terminator. These solutions impose a compromise on one of the agents.

When compromises are inevitable, and when there are multiple compromises to make, fairness can be interpreted to mean that both agents make an equivalent amount of compromise. An example scenario is the following. Jenny and Amy are planning for a film night, and also want to have a dinner together. Two movies are screening, Terminator and Harry Potter. Based on their individual reasoning, Jenny wants to watch Terminator and Amy wants to watch Harry Potter. As for the choice of food, Jenny wants to have Italian food whereas Amy wants to have Thai food. Due to budget constrains, they can only watch one movie and have one dinner.

Intuitively, the fair outcome in this example is that Jenny and Amy either watch Terminator and have Thai food or watch Harry Potter and have Italian food. Indeed, any other settlement, e.g. Terminator and Italian food or Harry Potter and Thai food, favors one agent while compromising the other agent. To map the intuition into a formal mechanism, we can take the following approach.

**Step 1:** Employ a score keeping system such that if a desire is fulfilled for an agent, the agent gets a score of 1; if a desire is not fulfilled for an agent, it gets a score of 0.

**Step 2:** The fair selections are ones that maximize the total score of all agents and minimize the difference between scores obtained by each individual agent.

The maximization is used to ensure agents’ desires are satisfied as much as possible. The minimization is used to ensure both agents make a similar amount of compromise. This approach assumes that all desires have the same value and values of desires are perceived equally by all agents. Alternatively, we can consider desires having different values, e.g. as in value-based argumentation [BC03]. As a further alternative, we can consider desires of agents as resources. Hence, satisfying desires becomes an issue of maximizing social welfare in the context of resources allocation. In this view, well-defined utility functions and social welfare criteria, such as Utilitarian, Egalitarian, Envy-freeness [CDE'06] can be used.

6. Related Work

Multi-agent conflict resolution has been a much studied area in AI. Tessier et. al [TCM01] presents a collection of papers that study various aspects of conflicts between agents, such as the definition and categorization of conflicts [TLFC01], conflicts in the view of sociology [Han01,MW01], and conflicts among collaborative agents [JT01,Cha01]. At a high level view, [TLFC01] categorizes conflicts as physical conflicts and knowledge conflicts. Physical conflicts are resource conflicts, where agents’ interests are hindered by insufficient resources. Effective resource sharing or operational coordination in multi-agent systems have hence been studied [TPS98,RH03,LS08]. Knowledge conflicts are epistemic conflicts, where agents have different views towards the environment and their own desires. To resolve knowledge conflicts, agents can merge potentially conflicting beliefs. Research in this area includes [Rev93,Cho98,KP98,BDKP02]. Our work is within the realm of resolving knowledge conflicts.

More recent development in argumentation [CaMS06,BH08,RS09,BCPS09] have demonstrated the versatility of various argumentation frameworks for conflict resolution.
Amgoud et al. [AP02, AK05, AK07] have explored how argumentation dialogues can be used as a process for resolving conflicts between agents. In their approach, conflicts are potentially conflicting arguments. As stated in [AK07], argumentation has the unique advantage that knowledge bases from different agents do not need to be merged statically. Rather, an interactive and progressive procedure is taken by agents interchangeably uttering their beliefs. Our approach is to aim at merging “relevant” part of the belief base of agents, defined as argumentation frameworks. We plan to further study how this “relevant” merge can be supported by dialogue.

In the context of belief revision in argumentation, [FKIS09] has surveyed a number of works that investigate the relation between belief revision and argumentation, thus setting the stage for the research presented in this paper. [CDLS08] has presented a study for revising a Dung-style abstract argumentation system by adding a new argument that interacts with one previous argument. The authors have studied how a single operation may affect various extensions of a set of arguments. [CMDK05, CMDK +07] have presented a framework for merging argumentation systems from Dung’s theory of argumentation. Their approach is composed of two steps. Firstly, all argumentation systems are expanded so every single argument is known by all argumentation systems. Then a voting mechanism is used to determine attacks that are recognized by all argumentation systems. [PC04] has presented a comparison between argumentation and belief revision. Our merge is based on focusing solely on the source of conflict (a goal) and does not result in a belief revision except for joining up all beliefs.

7. Conclusion and Future Work

In this paper, we have presented a two-agent conflict resolution mechanism based on merging ABA frameworks. We recognize two sources of conflicts between agents: agents being misinformed or bearing incomplete information, and agents having different desires. We have explored the merge operator, that takes two ABA frameworks and produces a single one. Information is shared during the merge and the resulting joint arguments satisfy both agents’ desires as much as possible. We have considered the issue of fairness when resolving a conflict. We have also discussed how to resolve conflicts with argumentation dialogues, in the ABA context.

Future work includes further investigating properties of the merge operator proposed in this paper. This includes investigation of the merge operation with respect to some other argumentation semantics and investigation on performing the merge via dialogues. We also plan to further explore strategies for resolving conflicts when no mutually agreeable solution exists and a fair solution needs to be found.

References


[Cha01] F. Chantemargue. Conflicts in collective robotics. In Tessier et al. [TCM01], pages 203–222.


[Han01] M. Hannebauer. Their problems are my problems. In Tessier et al. [TCM01], pages 63–100.


