On Explanations for Non-Acceptable Arguments

Xiuyi Fan and Francesca Toni

Imperial College London, London, United Kingdom,
{xf309,ft}@imperial.ac.uk

Abstract. Argumentation has the unique advantage of giving explanations to reasoning processes and results. Recent work studied how to give explanations for arguments that are acceptable, in terms of arguments defending it. This paper studies the counterpart of this problem by formalising explanations for arguments that are not acceptable. We give two different views (an argument-view and an attack-view) in explaining the non-acceptability of an argument and show the computation of explanations with debate trees.

1 Introduction

Argumentation (see e.g. [18, 20] for an overview) can be viewed as a process of generating *explanations*. Indeed, an arguing process transparently explains the procedure and the results of reasoning. Given a topic, the process of arguing can be viewed as identifying *related* information and generating an *explanation* for the topic, usually through some fictitious *proponent* and *opponent* debate game. Hence, arguing for an argument can be deemed to explain it.

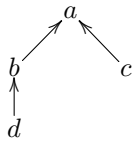
Recent work [14, 15] has proposed explaining the acceptability of an argument a as a set of arguments defending a. However, this approach fails to address the case when a is not acceptable; as, intuitively, an argument is not acceptable because it lacks appropriate defences against some attackers.

We propose two alternative views for explaining why some argument a is not acceptable. In the *argument-view*, we view an explanation for a with a defending set S as a set of argument A attacking S such that if A is removed, then a becomes acceptable. In the *attack-view*, we see an explanation for a as a set of attacks such that, if removed, a becomes acceptable. We analyse the relations between these two views of explanations.

We develop our notions of explanations in the context of Abstract Argumentation (AA) [9] as AA is arguably the most widely used argumentation framework with great simplicity. Also, the main approach we use in this work relies on a proof theory developed for AA, namely, dispute trees [11, 10]. Moreover, most other argumentation frameworks, e.g. Assumption-based Argumentation [11, 25] and ASPIC+ [17], are instances of AA; hence results obtained in AA apply to those frameworks as well. We will focus our discussion on the admissibility semantics thus equate arguments' acceptability with admissibility.

We motivate our approach with the following example on argumentation-based decision making, adapted from [13]:

Example 1. An agent needs to decide on accommodation in London, amongst three options: Imperial College Student Accommodation (ic), the John Howard Hotel (jh), and the Ritz Hotel (ritz). The main decision criterion is whether the accommodation is quiet. The agent believes that both ic and ritz are quiet, but jh is not. The decision to not choose ic can be represented by the following AA framework $\langle \mathcal{A}, \mathcal{R} \rangle$ (as conventional, represented as a directed graph with nodes being arguments in $\mathcal{A}$ and arcs being attacks in $\mathcal{R}$):



a: Choose ic.
b: Why not jh?
d: Because it is not quiet.
c: Why not ritz?

Here, the argument a is not acceptable as it cannot defend against argument c, even with the help of other arguments. Although b also attacks a, this attack is countered by d. Thus, one may conclude that either the *argument* c or the *attack* from c to a explains the non-acceptability of a, as *by removing either the argument or the attack* a *is acceptable.* Identifying the source of non-acceptability can then help repairing the AA framework to ensure a's acceptability, e.g. by adding an attack against c. In this paper, we focus on characterising explanations, not repairing AA frameworks.

The reminder of this paper is organised as follows. Section 2 reviews some background on AA and dispute trees. Section 3 introduces the two views of explanations. We will see that they do not in general coincide (although they do in Example 1). Section 4 gives the procedures of computing explanations with dispute trees. Section 5 discusses several issues with the difference between the two forms explanations and some other possible types of explanations. Section 6 reviews some related works. Section 7 concludes.

2 Background

Abstract Argumentation (AA) *frameworks* [9] are pairs $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, consisting of a set of *arguments*, $\mathcal{A}$, and a binary *attack* relation, $\mathcal{R}$. For any attack $(a, b) \in \mathcal{R}$, a is the *attacking argument*.

Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, an *extension* $A \subseteq \mathcal{A}$ is *admissible* (in AF) if and only if $\forall a, b \in A$, there is no $(a, b) \in \mathcal{R}$ (A is *conflict-free*) and for any $a \in A$, if $(c, a) \in \mathcal{R}$, then there exists some $b \in A$ such that $(b, c) \in \mathcal{R}$.

Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, we say that an argument a is *in* AF if and only if $a \in \mathcal{A}$; we also say that an attack (a, b) is *in* AF or that a *attacks* b in AF if and only if $(a, b) \in \mathcal{R}$. Finally, we say that an argument a is *admissible* if and only if a is in some admissible extension.

Dispute Trees [11, 10] are used to compute our explanations. Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, we will use the following version of dispute trees. A *dispute tree for* $a \in \mathcal{A}$ is a (possibly infinite) tree $\mathcal{T}$, such that:

1. every node of $\mathcal{T}$ is of the form $[\mathtt{L}{:}x]$, labelled by an argument x (in AF) and assigned the status of either *proponent (*$\mathtt{P}$*)* or *opponent (*$\mathtt{O}$*)* (thus $\mathtt{L} \in \{\mathtt{P}, \mathtt{O}\}$), but not both;
2. the root of $\mathcal{T}$ is $[\mathtt{P}{:}a]$;
3. for every node n of the form $[\mathtt{P}:b]$, for every argument c that attacks b in AF, there exists a child of n of the form $[\mathtt{O}{:}c]$;
4. for every node n of the form $[\mathtt{O}{:}b]$, there exists at most one child of n of the form $[\mathtt{P}{:}c]$ such that c attacks b in AF;
5. there are no other nodes in $\mathcal{T}$ except those given by 1-4.

We say that a node of the form $[\mathtt{L} : x]$ is a $\mathtt{L}$ node. The set of all arguments labelling $\mathtt{P}$ nodes in $\mathcal{T}$ is called the *defence set* of $\mathcal{T}$, denoted by $\mathcal{D}(\mathcal{T})$. A dispute tree $\mathcal{T}$ is an *admissible dispute tree* if and only if:

1. every $\mathtt{O}$ node in $\mathcal{T}$ has a child, and
2. no argument in $\mathcal{T}$ labels both a $\mathtt{P}$ and an $\mathtt{O}$ node.

Theorem 3.2 in [12] states the following, given an AF and an argument a in AF:

1. If $\mathcal{T}$ is an admissible dispute tree for a, then $\mathcal{D}(\mathcal{T})$ is admissible (in AF).
2. If $a \in A$ where $A \subseteq \mathcal{A}$ is an admissible extension (in AF) then there exists an admissible dispute tree for a with $\mathcal{D}(\mathcal{T}) = A'$ such that $A' \subseteq A$ and A' is admissible (in AF).

3 Two Different Notions of Explanation

We start with introducing the *pruning operator*, $\setminus$, as follows.

Definition 1. *Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ and a set of argument $A \subseteq \mathcal{A}$, the* pruning operator*, $\setminus$, is defined as $AF \setminus A = \langle \mathcal{A}', \mathcal{R}' \rangle$, where*

- $\mathcal{A}' = \mathcal{A} \setminus A$,
- $\mathcal{R}' = \{(x, y) | (x, y) \in \mathcal{R}$ *and* $x \in \mathcal{A}', y \in \mathcal{A}'\}$.

Note that in this work we overload the operator $\setminus$ in several ways. Indeed, this operator is also used for the standard set difference operator and as defined later in Definitions 5 and 7. In all cases, it removes the second input from the first input.

We first introduce *arg-explanations*, giving explanations in the "argument-view".

Definition 2. *Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $a \in \mathcal{A}$ be such that a is not admissible in AF. Then, if there exists some $A \subseteq \mathcal{A}$, such that:*

1. *a is admissible in $AF \setminus A$, and*
2. *there is no $A' \subset A$ such that a is admissible in $AF \setminus A'$,*

then A is an arg-explanation *of a. Otherwise, $\{a\}$ is the* arg-explanation *of a.*

Given an arg-explanation A of some argument a, we say that a is the topic *argument for A.*

The intuition behind Definition 2 is that an arg-explanation of a non-admissible argument a is a minimal (with respect to set inclusion) set of arguments A such that if A is removed, then a becomes admissible. However, such A may not always exist. In such case, we take the view that the reason for a being not admissible is a itself. It is easy to see that this happens if and only if a attacks itself.

Proposition 1. *Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $a \in \mathcal{A}$. The arg-explanation of a is $\{a\}$ if and only if $(a, a) \in \mathcal{R}$.*

A non-admissible argument can have multiple arg-explanations, as illustrated in the following example.

Example 2. Given the AA framework in Figure 1, there are two arg-explanations $\{b\}$ and $\{e\}$ for argument a. Indeed, removing either $\{b\}$ or $\{e\}$ from this AA framework makes the argument a admissible.

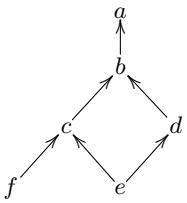


Fig. 1. AA framework for Example 2. Here the argument a has two arg-explanations ($\{b\}$ and $\{e\}$).

Proposition 2. *For any argument a in an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, if a is not admissible, then there is a non-empty arg-explanation of a.*

Proof. (Sketch.) By Proposition 1, if $(a, a) \in \mathcal{R}$, then the arg-explanation of a is $\{a\}$. Otherwise, since a is trivially admissible in the AA framework $AF' = \langle \{a\}, \{\} \rangle$, AF can always be reduced to AF' by removing arguments in $\mathcal{A}$.

We can see that arguments in an arg-explanation are always "related to" the argument being explained, formally given as follows.

Definition 3. *Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $x, y \in \mathcal{A}$. Then, x* is related to *y (in AF) if and only if:*

1. *$x = y$; or*
2. *$(x, y) \in \mathcal{R}$; or*

3. $\exists z \in \mathcal{A}$*, such that* $(x, z) \in \mathcal{R}$ *and* z *is related to* y*.*

Definition 3 is given recursively with (1) and (2) the base cases. Note that each argument is related to itself (by (1)). Note also that if there is no attack against an argument then the only argument related to it is the argument itself.

Proposition 3. *Let* A *be an arg-explanation for some non-admissible argument* a *in some AA framework* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$*. For all* $b \in A$*,* b *is related to* a *(in* AF*).*

Proof. (Sketch.) This proposition holds by the observation that for any (non-admissible) argument a, arguments not related to a do not affect its admissibility.

We now turn our attention to *att-explanations*, which give explanations in the "attack-view".

Definition 4. *Given an AA framework* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ *let* $a \in \mathcal{A}$ *be such that* a *is not admissible in* AF*. Then an* att-explanation *of* a *is a set of attacks* $R \subseteq \mathcal{R}$*, such that*

1. a *is admissible in* $\langle \mathcal{A}, \mathcal{R} \setminus R \rangle$*;*
2. *there is no* $R' \subset R$ *such that* a *is admissible in* $\langle \mathcal{A}, \mathcal{R} \setminus R' \rangle$*.*

Given an att-explanation R *of* a*, we say that* a *is the* topic *argument for* R*.*

The intuition behind Definition 4 is that the att-explanation of an argument a is a minimal (with respect to set inclusion) set of attacks such that a becomes admissible if these attacks are removed. Note that such R always exists, as shown by the following proposition.

Proposition 4. *For any argument* a *in an AA framework, if* a *is not admissible, then there is an att-explanation of* a *and every att-explanation of* a *is non-empty.*

Proof. (Sketch.) Trivially, as if a is not attacked, then a is admissible. Thus, we can always construct an att-explanation of a by including attacks of the form $(\_, a)$.[1]

Similarly to Proposition 3, the following holds.

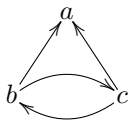
Proposition 5. *Let* R *be an att-explanation for* a *and* $(x, y) \in R$*. Then both* x *and* y *are related to* a*.*

Proof. (Sketch) This proposition holds as (1) if y is not related to a, then removing (x, y) does not affect the admissibility of a; and (2) if y is related to a, then x is.

One may hypothesise that arg-explanations and att-explanations of any argument a always coincide in the sense that the set formed by the attacking arguments in an att-explanation is an arg-explanation for a. The following example illustrates that this is not the case in general.

[1] Here and after, _ denotes an anonymous variable as in Prolog.

Example 3. We illustrate the difference between arg-explanations and attacking arguments in att-explanations. Consider the following AA framework AF:

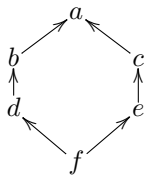


Here, a is not admissible as it is attacked by both b and c. To make a admissible, we can either remove **both** b and c (as removing only one of them is insufficient) hence the arg-explanation for a is $\{b, c\}$; or we can remove **either** the attack (b, a) **or** the attack (c, a). Thus, the attacking arguments in att-explanations are either b or c.

One interpretation of this example is that both b and c are at odds with a; and b and c are in mutual conflict. To make a admissible, we can either eliminate both b and c (arg-explanation); or we can ally a with either b or c (att-explanation).

One may also hypothesise that for any argument a, its att-explanations are always "more compact" than its arg-explanations in the sense that *the set of arguments formed by attacking arguments in an att-explanation is no bigger than any arg-explanations*, as in the case of Example 3. This is not true in general, as illustrated below.

Example 4. Consider the following AA framework AF:



Here, a is not admissible and $\{(b, a), (f, e)\}$ is an att-explanation for a. We can see that removing any of these two attacks alone from AF is insufficient to render a admissible. The attacking arguments in this att-explanation are b and f. However, it is easy to see that the set formed by f alone is an arg-explanation. Thus, removing f alone from AF renders a admissible.

The following proposition gives a formal link between arg-explanation and att-explanation.

Proposition 6. *Let A be an arg-explanation for some argument a in an AA framework $\langle \mathcal{A}, \mathcal{R} \rangle$ and $S = \{(x, y) \in \mathcal{R} | x \in A\}$. Then, there exists some att-explanation R for a such that $R \subseteq S$.*

Proof. (Sketch.) Trivially, as since A is an arg-explanation, removing A gives the same effect, as far as a is concerned, as removing all attacks from A.

4 Obtaining Explanations from Dispute Trees

Both arg-explanations and att-explanations can be obtained from dispute trees. Since admissible arguments correspond to admissible dispute trees (see Section 2), given a non-admissible argument a, no dispute tree with root argument a is admissible. We pose the question:

> *How do we turn a non-admissible dispute tree into an admissible dispute tree by removing (some of) its nodes?*

Answering this question effectively gives us arg-explanations for a. We provide an answer with *pruned trees*.

In this section we assume as given a general AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$.

Definition 5. *Given a dispute tree $\mathcal{T}$, the* pruned tree $\mathcal{T}'$ *(of $\mathcal{T}$) with respect to a set of arguments $A \subseteq \mathcal{A}$ (denoted with $\mathcal{T}' = \mathcal{T} \setminus A$) is a dispute tree such that a node $n = [\mathtt{L}{:}x]$ is in $\mathcal{T}'$ if and only if the following three conditions hold:*

1. *n is in $\mathcal{T}$; and*
2. *$x \notin A$; and*
3. *let $E = \{y | [\_{:}y]$ is an ancestor of n in $\mathcal{T}\}$; then $E \cap A = \{\}$.*

The intuition behind Definition 5 is that given a dispute tree $\mathcal{T}$ and a set of arguments A, pruning $\mathcal{T}$ with respect to A yields another tree $\mathcal{T}'$ such that $\mathcal{T}'$ does not contain any node labelled by arguments in A or nodes that "hung below" nodes labelled by arguments in A.

Later we will refer to some pruned tree $\mathcal{T}' = \mathcal{T} \setminus A$ as (non-)admissible without specifying in which AA framework. Implicitly, we will assume that this framework is $AF \setminus A$.

Example 5. (Example 3 continued.) An input dispute tree $\mathcal{T}$ for argument a is shown in Figure 2 (left). Two pruned trees $\mathcal{T}' = \mathcal{T} \setminus \{c\}$ and $\mathcal{T}'' = \mathcal{T} \setminus \{b, c\}$ are shown in the same figure (in the middle and on the right, respectively). We can see that neither $\mathcal{T}$ nor $\mathcal{T}'$ are admissible dispute trees (in AF and in $AF \setminus \{c\}$, respectively). However, $\mathcal{T}''$ is an admissible dispute tree (in $AF \setminus \{b, c\}$).

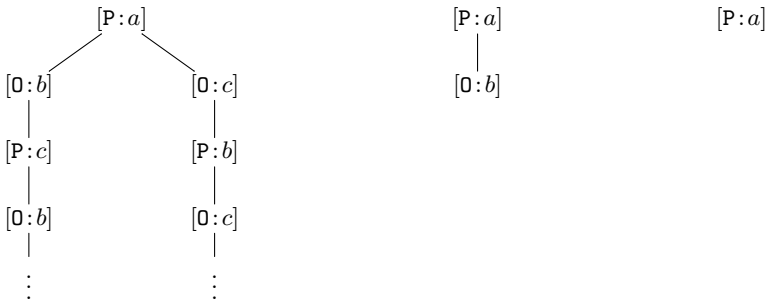


Fig. 2. Dispute tree $\mathcal{T}$ for argument a in Example 5 (left); a pruned tree $\mathcal{T}' = \mathcal{T} \setminus \{c\}$ (middle); a pruned tree $\mathcal{T}'' = \mathcal{T} \setminus \{b, c\}$ (right).

Trivially, the following proposition holds, stating that pruning a dispute tree with an empty set returns the same dispute tree.

Proposition 7. *Let $\mathcal{T}$ be a dispute tree. Then $\mathcal{T} \setminus \{\} = \mathcal{T}$.*

With pruned tree defined, we can identify arguments making a given dispute tree non-admissible, as follows.

Definition 6. *Given some argument a in AF, let $\mathcal{T}$ be a dispute tree for a. A* tree-arg-explanation *(with respect to $\mathcal{T}$) is a set of arguments A such that*

1. *$\mathcal{T} \setminus A$ is an admissible dispute tree (in $AF \setminus A$); and*
2. *there is no $A' \subset A$ such that $\mathcal{T} \setminus A'$ is an admissible dispute tree (in $AF \setminus A'$).*

Intuitively, given a dispute tree $\mathcal{T}$, a tree-arg-explanation is a minimal set of arguments such that the pruned tree $\mathcal{T} \setminus A$ is admissible. For the dispute tree $\mathcal{T}$ in Example 5, $\{b, c\}$ is the only tree-arg-explanation for $\mathcal{T}$.

Note that, since we require arg-explanations to be minimal (see Definition 2, condition 2), in general, tree-arg-explanations are not arg-explanations, as illustrated in the following example.

Example 6. Given the AA framework shown in Figure 1, consider the two dispute trees, $\mathcal{T}_1$ and $\mathcal{T}_2$, for the argument a, shown respectively in the left and the right in Figure 3. Although $\mathcal{T}_1 \setminus \{e, f\}$ and $\mathcal{T}_1 \setminus \{b\}$ are admissible, $\mathcal{T}_1 \setminus \{e\}$ is not. Indeed, both $\{e, f\}$ and $\{b\}$ are tree-arg-explanations with respect to $\mathcal{T}_1$.

Also, both $\mathcal{T}_2 \setminus \{e\}$ and $\mathcal{T}_2 \setminus \{b\}$ are admissible. Thus, both $\{e\}$ and $\{b\}$ are tree-arg-explanations with respect to $\mathcal{T}_2$.

By Definition 2, $\{e\}$ and $\{b\}$ are arg-explanations for a and $\{e, f\}$ is not an arg-explanation, although it is a tree-arg-explanation with respect to $\mathcal{T}_1$.

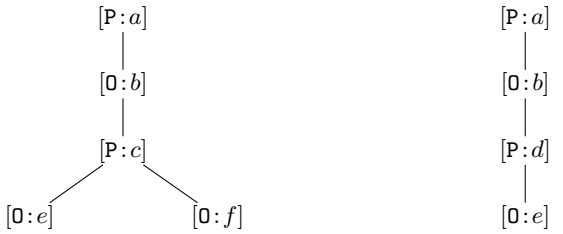


Fig. 3. Two dispute trees $\mathcal{T}_1$ (left) and $\mathcal{T}_2$ (right) for a in the AA framework in Figure 1.

Proposition 8. *Given an argument $a \in \mathcal{A}$, for any $A \subseteq \mathcal{A}$, if a is admissible in $AF \setminus A$, then there is a dispute tree $\mathcal{T}$ for a in AF such that $\mathcal{T}' = \mathcal{T} \setminus A$ is an admissible dispute tree for a in $AF \setminus A$.*

Proof. If a is admissible in AF, then we let $A = \{\}$, by Proposition 7 and Theorem 3.2 in [12], this proposition holds.

If a is not admissible in AF, we need to show that $\mathcal{T}$ can be constructed from $\mathcal{T}'$. We let $\mathcal{T}$ be the limit of the sequence $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ constructed as follows:

1. $\mathcal{T}_1 = \mathcal{T}'$;
2. $\mathcal{T}_{i+1}$ is $\mathcal{T}_i$ with a new node $[\mathtt{L}:x]$ as the child of some node $[\_:y]$ such that
   (a) $x \in A$;

(b) (x, y) is in AF;
(c) $\mathcal{T}_{i+1}$ is a dispute tree.

With this construction, we know that $\mathcal{T}_n$ is a dispute tree for a as $\mathcal{T}_1$ is a dispute tree for a. We can see that $\mathcal{T}_n \setminus A = \mathcal{T}'$ as the specified construction "reverses" the pruning. Hence the proposition holds.

Proposition 8 sanctions that dispute trees give a "complete" approach for computing arg-explanations. In other words, if a set of arguments is an arg-explanation for some argument a (in some AA framework AF), then it will not be missed by looking at dispute trees for a (in AF).

With Proposition 8, we are ready to show the main result for computing arg-explanations with dispute trees, as follows.

Theorem 1. *Given an argument a in AF, let $TT = \{\mathcal{T}_1, \ldots, \mathcal{T}_n, \ldots\}$ be the set of all dispute trees for a and $S = \{A | A$ is a tree-arg-explanation with respect to $\mathcal{T}_i$, for any $\mathcal{T}_i \in TT\}$. For all $A \in S$, if there is no $A' \in S$ such that $A' \subset A$, then A is an arg-explanation for a.*

Proof. To show that A is an arg-explanation for a is to show

1. a is admissible in $AF \setminus A$; and
2. A is a minimal set (with respect to $\subseteq$) satisfying 1.

Condition 1 holds as, since $A \in S$, A is a tree-arg-explanation. Thus, there is some dispute tree $\mathcal{T}_i \in TT$ for a such that $\mathcal{T}_i \setminus A$ is an admissible dispute tree. By Theorem 3.2 in [12], a is admissible in $AF \setminus A$.

Condition 2 holds as there is no $A' \in S$ such that $A' \subset A$; and by Proposition 8, there is no other set of arguments A^* such that both of the following two conditions hold:

1. a is admissible in $AF \setminus A^*$; and
2. there does not exist $A_i \in S$ for which $A_i \subseteq A^*$.

As both conditions hold, the theorem holds.

Thus far, we have shown how arg-explanations can be computed with dispute trees (namely dispute trees are a "sound" mechanism for obtaining arg-explanations).In the rest of this section, we study obtaining att-explanations from dispute trees. We start with defining pruned trees with respect to attacks, as follows.

Definition 7. *Given a dispute tree $\mathcal{T}$, the* pruned tree $\mathcal{T}'$ *(of $\mathcal{T}$) with respect to a set of attacks R is a dispute tree (denoted with $\mathcal{T}' = \mathcal{T} \setminus R$) such that a node $n = [\mathtt{L}{:}x]$ ($\mathtt{L} \in \{\mathtt{P}, \mathtt{O}\}$) is in $\mathcal{T}'$ if and only if the following three conditions hold:*

1. *n is in $\mathcal{T}$; and*
2. *if n is a child of $n' = [\_{:}y]$ in $\mathcal{T}$, then $(x, y) \notin R$; and*
3. *let $S = \{n' | n'$ is an ancestor of n in $\mathcal{T}\}$; then for all $n_1 = [\_ : w] \in S$, $n_2 = [\_{:}z] \in S$ such that n_1 is a child of n_2, we have $(w, z) \notin R$.*

The intuition behind Definition 7 is that given a dispute tree $\mathcal{T}$ and a set of attacks R, pruning $\mathcal{T}$ with respect to R yields another tree $\mathcal{T}'$ such that $\mathcal{T}'$ does not contain any branch rooted at x with y the parent of x, where $(x, y) \in R$.

Example 7. (Example 5 continued.) Given the dispute tree $\mathcal{T}$ for argument a shown in Figure 2 (left), the pruned tree $\mathcal{T}^* = \mathcal{T} \setminus \{(c, a)\}$ is shown in Figure 4. $\mathcal{T}^*$ is an admissible dispute tree.

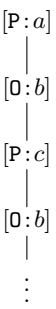


Fig. 4. A pruned tree $\mathcal{T}^* = \mathcal{T} \setminus \{(c, a)\}$ for Example 7.

Following the same idea behind Definition 6, we define *tree-att-explanation* as follows.

Definition 8. *Given a dispute tree $\mathcal{T}$ for some argument a, a* tree-att-explanation *(with respect to $\mathcal{T}$) is a set of attacks $R \subseteq \mathcal{R}$ such that*

1. *$\mathcal{T} \setminus R$ is an admissible dispute tree; and*
2. *there is no set of attacks $R' \subset R$ such that $\mathcal{T} \setminus R'$ is admissible.*

In the same way that tree-arg-explanations are not always arg-explanations, tree-att-explanations are not always att-explanations, as illustrated in the following example.

Example 8. Given the AA framework shown in Figure 5 (left), there are two dispute trees, $\mathcal{T}_1$ and $\mathcal{T}_2$, for argument a (shown respectively in the middle and the right in Figure 5).

From $\mathcal{T}_1$, we see that $\{(h, c), (g, f)\}$ is a tree-att-explanation. Yet, from $\mathcal{T}_2$ we see that $\{(g, f)\}$ alone is also a tree-att-explanation. Thus, the former tree-att-explanation is not an att-explanation and the latter is.

Similarly to Proposition 8, the following proposition for att-explanations holds, sanctioning a form of "completeness" for obtaining att-explanations.

Proposition 9. *Given an argument a in AF, for any set of attacks R in AF, if a is admissible in $AF \setminus R$, then there is a dispute tree $\mathcal{T}$ for a in AF such that $\mathcal{T} \setminus R$ is an admissible dispute tree.*

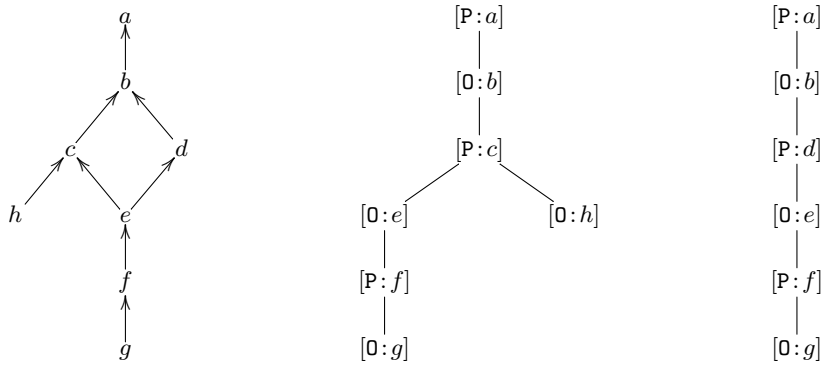


Fig. 5. AA framework in Example 8 (left); a dispute tree $\mathcal{T}_1$ for argument a (middle); another dispute tree $\mathcal{T}_2$ for a (right).

Finally, we are ready to show the main result for obtaining att-explanations with dispute trees, as follows.

Theorem 2. *Given an argument a in AF, let $TT = \{\mathcal{T}_1, \ldots, \mathcal{T}_n, \ldots\}$ be the set of all dispute trees for a and $S = \{A | A$ is a tree-att-explanation with respect to $\mathcal{T}_i$, for any $\mathcal{T}_i \in TT\}$. For all $R \in S$, if there is no $R' \in S$ such that $R' \subset R$, then R is an att-explanation for a.*

The proof of Theorem 2 is similar to the one of Theorem 1.

5 Discussion

In this paper, we have given two different notions of explanations, the "argument-view" and the "attack-view". Comparing the two, the following observations can be made.

Firstly, arg-explanations are more suitable for identifying "fixes" for arguments not being admissible. For instance, given an arg-explanation, to make the topic argument admissible, one can just add new attacks to all arguments in the arg-explanation. Thus, for dialectical applications such as *persuasion* in multi-agent systems (e.g. see [19, 26]), identifying arg-explanations helps agents know effective attacking points, i.e. arguments to attack to render the topic admissible. It is easy to see that att-explanations do not grant this ability, as inserting new arguments attacking the attacking arguments in an att-explanation does not necessarily change the admissibility of the topic. For instance, inserting a new argument d attacking c in Example 3 does not make a admissible, though $\{(c, a)\}$ is an att-explanation and c is the attacking argument in (c, a).

Secondly, we have enforced minimality while defining both arg-explanations and att-explanations in Definitions 2 and 4, respectively. As a consequence, as illustrated in Examples 6 and 8, computing both arg-explanations and att-explanations requires the construction of all dispute trees for the topic argument.

Constructing all dispute trees for an argument might be deemed to be too expensive computationally for certain applications. For both arg-explanations and att-explanations, in addition to tree-arg/att-explanations, we can consider *rel-arg/att-explanations* as alternatives, briefly discussed below.

The second conditions in Definition 2 and 4, where minimality is required, can be relaxed to *relatedness*, i.e. for arg-explanations, informally:

> A *rel-arg-explanation* for some non-admissible topic argument a is a set of arguments A such that: (1) if A is removed, then a becomes admissible; and (2) every argument in A is related to a as in Definition 3.

By Proposition 3, arg-explanations are rel-arg-explanations. Moreover, it is easy to see that tree-arg-explanations are also rel-arg-explanations. We observe that rel-arg-explanations are easy to obtain, e.g. the set of arguments labelling opponent nodes in a dispute tree gives a rel-arg-explanation. However, it can be viewed that such oversimplification renders rel-arg-explanation less useful for the purpose of recognising the "true source" that triggers the non-admissibility of the topic. Similar reasoning can be applied for att-explanations.

With rel-arg/att-explanations and arg/att-explanations at two extremes, one may think that tree-arg/att-explanations give a good compromise between the usefulness of such explanations and their computation complexity, i.e. obtaining a tree-arg-explanation requires computing a dispute tree with a minimal set of opponent arguments within the tree. Thus, for applications where computing arg/att-explanations is too expensive to be affordable, computing tree-arg/att-explanations could be a suitable alternative for understanding why the topic argument is not admissible.

Thirdly, in this work, we made no distinction between different arg/att-explanations. As illustrated in Example 2, in general, there are multiple arg-explanations for a single topic argument. In Example 2, one may argue that $\{e\}$ is a more reasonable explanation for a as it is the "root of the cause" whereas b is less suitable as it already has two "immediate responses", arguments c and d. However, such reasoning itself is unconvincing as it could be equally well argued that "*if the problem at b is addressed, then there is no need to worry about anything else*". Similar reasoning can be applied to att-explanations as well. Thus, we take the view that making further distinction between arg/att-explanations is difficult and possibly application-dependent.

6 Related Work

[14, 15] have introduced the *related admissibility* argumentation semantics to capture explanations for admissible arguments in both AA and Assumption-based Argumentation. Given an admissible argument as the topic, they model its explanations as a set of arguments defending the topic. They also use dispute trees to compute explanations. Roughly speaking, arguments in proponent nodes from an admissible dispute tree are an explanation for the argument in the root

of the tree. They have not studied explanations for non-admissible arguments or explanations in the "attack-view".

[3] have studied revising AA frameworks by adding new arguments which may interact with existing arguments. They have studied the behaviour of the extensions of the augmented argumentation frameworks, taking also into account possible changes of the underlying semantics. Our work is orthogonal to theirs. We are interested in finding explanations for non-admissible arguments and the forms of explanations we study in this paper are concerned with removing arguments or attacks.

[22] have introduced dynamic argumentation frameworks and allowed various revision operators being applied. Their work is performed at a "meta-level" in the sense that both the underlying logic for arguments and argumentative semantics are left unspecified. Their work is focused on understanding dynamic changes represented in argumentation frameworks and defining operators modelling these changes. Our work differs from theirs as we focus on abstract argumentation and generating explanations.

[16] have studied the minimal changes needed to make some arguments acceptable in an argumentation framework. They have considered two types of changes: adding or removing attacks. Their work is motivated by agents in persuasion. However, in their setting, the set of arguments in the argumentation framework is fixed and only certain attacks can be added or removed. Our study of att-explanations is closely related to their work. However, we have relied on different approaches (with dispute trees) for finding att-explanations whereas they have used a set of rewriting rules.

[6] have studied the impact of adding a new argument to an AA framework, particularly on the set of its extensions. The authors have studied several properties for this type of changes under the grounded and preferred semantics. They are not concerned with giving explanations to the (non-) acceptability of arguments. Comparing with their work, ours is not about revising AA frameworks, but identifying arguments and attacks that affect the non-acceptability of arguments.

[4] have studied the impact of removing a single argument from an AA framework on the set of extensions. Their work is situated in a legal context. Our work is different as we are not concerned with changes to all extensions when a particular argument is removed. Rather, half of our paper concerns which arguments are responsible for the non-admissibility of arguments.

[2] have studied different types of *expansions*, that is, different ways to modify an existing AA framework. In their work, they allow the addition of new arguments, as well as the addition/removal of attacks. The problem studied there is: given an argumentation system and a "goal set" E, find a minimal expansion such that E belongs to at least one extension of the modified system. Though related, they are clearly solving a different problem as we are not concerned with adding arguments or attacks.

[5] use the notion of *explanation dialogues* to represent dialogical proof procedures for abductive argumentation framework. Their notion of explanations

is closer to the ones introduced in [14, 15], i.e. focus on explanations for sentences (arguments) that are in certain extensions (acceptable), instead of non-acceptable arguments.

[23] also study explanations of arguments as two sets of arguments, a "removal set" and an "addition set". Roughly speaking, an argument can be made acceptable by removing arguments from the removal set and inserting arguments from the addition set. Though their notion of explanation is similiar to our arg-explanations, their computation is not based on debate trees or forests. They have not considered att-explanations.

[1] present a work on explanation for failure query in inconsistent knowledge base with argumentation. Their work focuses on using argumentation dialogue to explain a single type of query whereas ours aims at introducing a general theory of explanation for unacceptable arguments.

The literature on human-computer interaction includes a considerable amount of work on explanation in various contexts, e.g. for recommender systems [24], and on evaluating empirically various explanatory tools according to various criteria such as effectiveness and transparency [24]. We have focused on defining various notions of explanation for abstract argumentation, and in particular for non-membership of arguments in admissible extensions. It would be interesting in the future to evaluate empirically our techniques, and in particular the relative merits of the various notions of explanation we have defined according to criteria identified in the HCI literature.

We have defined tree-att-explanations (see Definition 8) in terms of a pruning operator over dispute trees. Other forms of pruning have been defined in the literature, e.g. in [8], but for different tasks and frameworks, e.g., in the case of [8], for improving query answering in Possibilistic Defeasible Logic Programming. It would be nonetheless interesting to study whether other forms of pruning could provide other notions of explanation for abstract argumentation, and whether our form of pruning could serve the purpose of defining explanatory methods in other frameworks.

7 Conclusion

Argumentation has its unique advantage in explaining the process and results of its computation. To fully exploit this advantage, [14, 15] study explanations for admissible arguments. In short, that work considers explanations for an admissible argument as arguments defending it. In this work, we shift our focus to explanations for arguments that are not admissible. We aim to be able to explain *why some argument is not admissible.* We take the view that an argument a is not admissible because of the presence of some arguments A or attacks R, such that if A or R are removed, then a becomes admissible. Thus, an explanation in the "argument-view" (arg-explanation) of a is A and an explanation in the "attack-view" (att-explanation) of a is R.

We have shown that, although exhibiting similarities, arg-explanations and att-explanations for the same argument do not always coincide. We have used dispute trees for obtaining both forms of explanations.

Explanations studied in this work are based on the admissibility semantics in abstract argumentation. In the future, we would like to explore explanations with other semantics and other argumentation formalisms. Note that in this paper we have already implicitly addressed explanations for arguments not belonging to any preferred extension [9], since preferred extensions are maximally admissible and every admissible extension is contained in some preferred extension; therefore, if an argument does not belong to any preferred extension then it does not belong to any admissible extension either. In addition, we plan to study other types of explanations, e.g. based on relatedness rather than minimality. The computation approach introduced in this work is based on dispute trees. It will be interesting to see if other approaches, e.g. labelling-based, can be developed. Moreover, explanations are studied in this work from a theoretical viewpoint. It would be very useful if experiments of our notions of explanations could be conducted with real users from a human-computer interaction perspective, e.g. along the lines of [21, 24, 7]. Finally, it would be interesting to study the various notions of explanation we have defined from a computational complexity perspective, to determine their computational viability.

Acknowledgements

This research was supported by the EPSRC TRaDAr project *Transparent Rational Decisions by Argumentation*: EP/J020915/1.

References

1. A. Arioua, N. Tamani, M. Croitoru, and P. Buche. Query failure explanation in inconsistent knowledge bases using argumentation. In *COMMA Proc.*, pages 101–108, 2014.
2. R. Baumann. What does it take to enforce an argument? minimal change in abstract argumentation. In *Proc. ECAI*, pages 127–132, 2012.
3. R. Baumann and G. Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010.*, pages 75–86, 2010.
4. P. Bisquert, C. Cayrol, F. D. Saint-Cyr, and M. Lagasquie-Schiex. Change in argumentation systems: Exploring the interest of removing an argument. In *Scalable Uncertainty Management - 5th International Conference, SUM 2011, Dayton, OH, USA, October 10-13, 2011. Proceedings*, pages 275–288, 2011.
5. R. Booth, D. M. Gabbay, S. Kaci, T. Rienstra, and L. V. D. Torre. Abduction and dialogical proof in argumentation and logic programming. In *Proc. ECAI*, pages 117–122, 2014.
6. C. Cayrol, F. D. Saint-Cyr, and M. Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *JAIR*, 38:49–84, 2010.

7. Federico Cerutti, Nava Tintarev, and Nir Oren. Formal arguments, preferences, and natural language interfaces to humans: an empirical evaluation. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 207–212, 2014.
8. Carlos Iván Chesñevar, Guillermo Ricardo Simari, and Lluis Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Logic Programming and Nonmonotonic Reasoning, 8th International Conference, LPNMR 2005, Diamante, Italy, September 5-8, 2005, Proceedings*, pages 158–171, 2005.
9. P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AIJ*, 77(2):321–357, 1995.
10. P.M. Dung, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *AIJ*, 170:114–159, 2006.
11. P.M. Dung, R.A. Kowalski, and F. Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218. Springer, 2009.
12. P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *AIJ*, 171(10–15):642–674, 2007.
13. X. Fan and F. Toni. Decision making with assumption-based argumentation. In *Proc. TAFA*, pages 127–142. Springer, 2013.
14. X. Fan and F. Toni. On computing explanation in abstract argumentation. In *Proc. ECAI*, 2014.
15. X. Fan and F. Toni. On computing explanations in argumentation. In *Proc. of AAAI*, 2015.
16. D. Kontarinis, E. Bonzon, N. Maudet, A. Perotti, L. v. D. Torre, and S. Villata. Rewriting rules for the computation of goal-oriented changes in an argumentation system. In *Proc. CLIMA*, pages 51–68, 2013.
17. S. Modgil and H. Prakken. The $ASPIC^+$ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.
18. S Modgil, F Toni, F Bex, I Bratko, C Chesñevar, W Dvořák, M Falappa, X Fan, S Gaggl, A García, M González, T. Gordon, J Leite, M Možina, C Reed, G Simari, S Szeider, P Torroni, and S Woltran. The added value of argumentation. In *Agreement Technologies*, volume 8, pages 357–403. Springer, 2013.
19. H. Prakken. Formal systems for persuasion dialogue. *Knowledge Engineering Review*, 21(2):163–188, 2006.
20. I. Rahwan and G. R. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.
21. Iyad Rahwan, Mohammed Iqbal Madakkatel, Jean-François Bonnefon, Ruqiyabi Naz Awan, and Sherief Abdallah. Behavioral experiments for assessing the abstract argumentation semantics of reinstatement. *Cognitive Science*, 34(8):1483–1502, 2010.
22. N. D. Rotstein, M. O. Moguillansky, M. A. Falappa, A. J. García, and G. R. Simari. Argument theory change: Revision upon warrant. In *Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30, 2008.*, pages 336–347, 2008.
23. C. Sakama. Abduction in argumentation frameworks and its use in debate games. In *LNCS 8417*, pages 285–303, 2013.
24. Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems - methodological issues and empirical studies on the

impact of personalization. *User Model. User-Adapt. Interact.*, 22(4-5):399–439, 2012.
25. F. Toni. A tutorial on assumption-based argumentation. *Argument & Computation, Special Issue: Tutorials on Structured Argumentation*, 5(1):89–117, 2014.
26. D. Walton and E. Krabbe. *Commitment in Dialogue: Basic concept of interpersonal reasoning*. State University of New York Press, 1995.