

On Monotonicity of Dispute Trees as Explanations for Case-Based Reasoning with Abstract Argumentation

Guilherme Paulino-Passos¹, Francesca Toni¹

¹Imperial College London, Department of Computing, London, United Kingdom

Abstract

Recent work on explainability raises the question of what different types of explanations actually mean. One idea is that explanations can reveal information about the behaviour of the model on a subset of the input space. When this way of interpreting explanations is thought as an interactive process, inferences from explanations can be seen as a form of reasoning. In the case of case-based reasoning with abstract argumentation (AA-CBR), previous work has used arbitrated dispute trees as a methodology for explanation. Those are dispute trees where nodes are seen as losing or winning depending on the outcome for the new case under consideration. In this work we show how arbitrated dispute trees can be readapted for different inputs, which allows a broader interpretation of them, capturing more of the input-output behaviour of the model. We show this readaptation is correct by construction, and thus the resulting reasoning based on this reuse is monotonic and thus necessarily a faithful explanation.

Keywords

Explainable AI, Interactive Explanations, Argumentation, Case-Based Reasoning

1. Introduction

Recent work on explainability raises the question of what different types of explanations actually mean. In particular, one general idea is that explanations can reveal information about model behaviour on at least some other inputs. That is, an explanation can, at least in principle, suggest some input-output behaviour of the model. Previous work has modelled this as a form of reasoning, that can be possibly seen as non-monotonic when this usage of the explanation for extrapolating model behaviour can be revised by further interactions [1].

Dispute trees have been proposed as explanations for argumentation methods, in particular for classification with *AA-CBR*, a model based on abstract argumentation (AA) for case-based reasoning (CBR) [2, 3]. In the more recent version, *arbitrated* dispute trees (ADTs) are shown as trees reflecting whether an argument is winning or losing, depending on the new case being classified [3].

Here we show how an ADT for classifying a new case can possibly be reused for other new cases. This shows that an ADT can be seen as “committing” to outcomes for other possible cases, instead of simply the case to be explained (and previous cases occurring in the ADT itself).

1st International Workshop on Argumentation for eXplainable AI (ArgXAI, co-located with COMMA '22), September 12, 2022, Cardiff, UK


✉ g.passos18@imperial.ac.uk (G. Paulino-Passos); f.toni@imperial.ac.uk (F. Toni)

🌐 <https://clarg.doc.ic.ac.uk> (G. Paulino-Passos); <https://www.doc.ic.ac.uk/~ft/> (F. Toni)

🆔 0000-0003-3089-1660 (G. Paulino-Passos); 0000-0001-8194-1459 (F. Toni)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Indeed, this reuse yields another admissible dispute tree, and is therefore a faithful explanation, by construction.

In this work we instantiate our previous model of interactive explanations to dispute trees in *AA-CBR*, via a method of readapting a dispute tree providing explanation for a prediction. Such method is not always applicable but when it is it results in a provably faithful explanation. When modelling this as reasoning, we show this will result in a form of monotonic inference.

2. Background

2.1. Abstract Argumentation

An *abstract argumentation framework (AF)* [4] is a pair $(Args, \rightsquigarrow)$, where $Args$ is a set (of arguments) and $\rightsquigarrow \subseteq Args \times Args$ is a binary relation on $Args$. For $\alpha, \beta \in Args$, if $\alpha \rightsquigarrow \beta$, then we say that α *attacks* β and that α is an *attacker* of β . For a set of arguments $E \subseteq Args$ and an argument $\alpha \in Args$, E *defends* α if for all $\beta \rightsquigarrow \alpha$ there exists $\gamma \in E$ such that $\gamma \rightsquigarrow \beta$. Then, the *grounded extension* of $(Args, \rightsquigarrow)$ can be constructed as $\mathbb{G} = \bigcup_{i \geq 0} G_i$, where G_0 is the set of all unattacked arguments, and $\forall i \geq 0, G_{i+1}$ is the set of arguments that G_i defends. For any $(Args, \rightsquigarrow)$, the grounded extension \mathbb{G} always exists and is unique and, if $(Args, \rightsquigarrow)$ is well-founded [4], extensions under other semantics (e.g. stable extensions) are equal to \mathbb{G} . For finite AFs, $(Args, \rightsquigarrow)$ is well-founded iff it is acyclic. Given $(Args, \rightsquigarrow)$, we will sometimes use $\alpha \in (Args, \rightsquigarrow)$ to stand for $\alpha \in Args$.

2.2. Abstract Argumentation for Case-Based Reasoning

We will present *AA-CBR* following the presentation of Paulino-Passos and Toni [5], referring to the model as *AA-CBR_≽*. *AA-CBR_≽* maps a *dataset* D of *examples* labelled with an *outcome* and an *unlabelled example* (with unknown outcome) into an AF. The dataset may be understood as a *casebase*, the labelled examples as *past cases* and the unlabelled example as a *new case*: we will use these terminologies interchangeably throughout. Cases have a characterisation, and outcomes are chosen from two available ones, one of which is selected up-front as the *default outcome*. Finally we assume that the set of characterisations of (past and new) cases is equipped with a partial order \preceq (whereby $\alpha \prec \beta$ holds if $\alpha \preceq \beta$ and $\alpha \neq \beta$ and is read “ α is less *specific* than β ”) and with a relation $\not\sim$ (whereby $\alpha \not\sim \beta$ is read as “ β is *irrelevant* to α ”). Previous literature shows examples of partial orders for specific applications [6]. Formally:

Definition 1. Let X be a set of characterisations, equipped with partial order \prec and binary relation $\not\sim$. Let $Y = \{\delta_o, \bar{\delta}_o\}$ be the set of (all possible) outcomes, with δ_o the default outcome. Then, a casebase D is a finite set such that $D \subseteq X \times Y$ (thus a past case $\alpha \in D$ is of the form (α_C, α_o) for $\alpha_C \in X, \alpha_o \in Y$) and a new case is of the form $(N_C, ?)$ for $N_C \in X$. We also discriminate a particular element $\delta_C \in X$ and define the default argument $(\delta_C, \delta_o) \in X \times Y$.

A casebase D is coherent iff there are no two cases $(\alpha_C, \alpha_o), (\beta_C, \beta_o) \in D$ such that $\alpha_C = \beta_C$ but $\alpha_o \neq \beta_o$, and it is incoherent otherwise.

For simplicity of notation, we sometimes extend the definition of \preceq and $\not\sim$ to $X \times Y$, by setting $(\alpha_c, \alpha_o) \preceq (\beta_c, \beta_o)$ iff $\alpha_c \preceq \beta_c$.

Definition 2. The AF mined from a dataset D and a new case $(N_C, ?)$ is $(Args, \rightsquigarrow)$, in which:

- $Args = D \cup \{(\delta_C, \delta_o)\} \cup \{(N_C, ?)\}$;
- for $(\alpha_C, \alpha_o), (\beta_C, \beta_o) \in D \cup \{(\delta_C, \delta_o)\}$, it holds that $(\alpha_C, \alpha_o) \rightsquigarrow (\beta_C, \beta_o)$ iff
 1. $\alpha_o \neq \beta_o$,
 2. $\alpha_C \succeq \beta_C$, and
 3. $\nexists (\gamma_C, \gamma_o) \in D \cup \{(\delta_C, \delta_o)\}$ with $\alpha_C \succ \gamma_C \succ \beta_C$ and $\gamma_o = \alpha_o$;
- for $(\beta_C, \beta_o) \in D \cup \{(\delta_C, \delta_o)\}$, it holds that $(N_C, ?) \rightsquigarrow (\beta_C, \beta_o)$ iff $(N_C, ?) \not\rightsquigarrow (\beta_C, \beta_o)$.

The AF mined from a dataset D alone is $(Args', \rightsquigarrow')$, with $Args' = Args \setminus \{(N_C, ?)\}$ and $\rightsquigarrow' = \rightsquigarrow \cap (Args' \times Args')$.

Definition 3. Let \mathbb{G} be the grounded extension of the AF mined from D and $(N_C, ?)$, with default argument (δ_C, δ_o) . The **outcome for** N_C is δ_o if (δ_C, δ_o) is in \mathbb{G} , and $\bar{\delta}_o$ otherwise.

Definition 4. The AF mined from D alone and the AF mined from D and $(N_C, ?)$, with default argument (δ_C, δ_o) , are **regular** when the following holds:

1. the irrelevance relation $\not\sim$ is defined as: $x_1 \not\sim x_2$ iff $x_1 \not\preceq x_2$, and
2. δ_C is the least element of X .¹

In this paper we will restrict our attention to regular mined AFs. We will refer to the (regular) AF mined from D and $(N_C, ?)$, with default argument (δ_C, δ_o) , as $AF_{\succeq}(D, N_C)$, and to the (regular) AF mined from D alone as $AF_{\succeq}(D)$.

Example 1. For concreteness, let us consider a simple example of credit application. Applicants will be described by two features: age and income. System developers decided from background knowledge and previous data that the default case of an applicant is of $\{\text{age}:30; \text{income}:25\}$ (age in years, income in £1000 per year) and by the default it is denied ($y = 0$). The casebase is formed of the following cases:

$$D = \{(\{\text{age}:20; \text{income}:100\}, 1), (\{\text{age}:40; \text{income}:23\}, 1), (\{\text{age}:32; \text{income}:40\}, 1), (\{\text{age}:60; \text{income}:42\}, 0), (\{\text{age}:86; \text{income}:150\}, 1)\}$$

More formally, we could say inputs are elements of $X = \mathbb{R}^2$, and outputs of $Y = \{0, 1\}$. We will interpret the partial order in the following way: $\{\text{age}:a; \text{income}:b\} \preceq \{\text{age}:a2; \text{income}:b2\}$ iff i) $30 \leq a \leq a2$ or $a2 \leq a \leq 30$; and ii) $25000 \leq b \leq b2$ or $b2 \leq b \leq 25000$. Essentially, a case α is smaller than β when, for both dimensions (features), the value of the dimension for α is between β and the default value. Notice this means that the notion of atypicality, represented by the partial order \preceq , is neither being too big or being too small. The corresponding framework is presented in Figure 1.

¹Indeed this is not a strong condition, since it can be proved that if $x_\alpha \not\preceq \delta_C$ then all cases (x_α, y_α) in the casebase could be removed, as they would never change an outcome. On the other hand, assuming also the first condition in Definition 4, if $(x_\alpha, ?)$ is the new case and $x_\alpha \not\preceq \delta_C$, then the outcome is $\bar{\delta}_o$ necessarily.

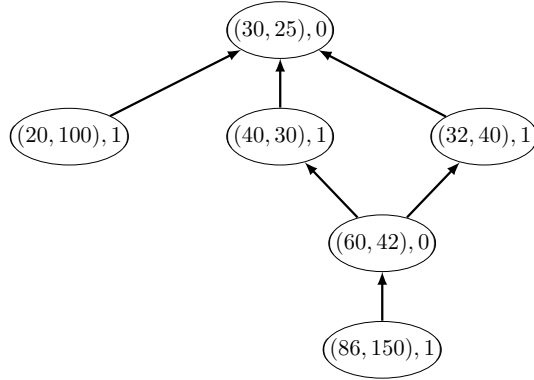


Figure 1: Case base D from Example 1 structured as an AA framework (i.e. without a focus case).

3. The interactive process

A more in-depth discussion of interactive explanations as reasoning is in previous literature [1]. Here, we make a more concise presentation, so the key concepts can be applied later.

3.1. Intuition

The motivation for interactive explanations is scenarios where a user evaluates the behaviour of a model via multiple, sequential, queries. They may ask for the output for a specific input, and ask as well for an explanation. The explanation may motivate then a new query, resulting in a new output and explanation. One may thus ask what should we a priori expect of such process, or such sequence of explanations. We thus consider an interactive process (as overviewed in Fig. 2), where the user queries for an output and explanation thereof, given an input, the system returns them, and then the user may query again. We see the AI system as including a *classifier* and an *explainer*, with both considered black-boxes by our conceptual model.

We assume that the explanation method (although not the classifier) can keep track of the *history* of inputs it received and their outputs. This allows supporting two scenarios: i) of an explainer that tries to improve its explanations by knowing what it explained before; and ii) of a malicious explainer that, trying to manipulate or mislead the user and to avoid being detected, keeps track of what was explained before.

Histories give a snapshot of the process by finite sequences of inputs and their outputs. Note that we assume the explainer to be a function, thus ignoring randomness of the explanation method itself. This assumption implies that no information about the computed explanations needs to be stored in the history, as awareness of earlier inputs and outputs already gives all information needed to infer what explanations were also previously returned.

Example 2. *Let us illustrate the framework with a model of counterfactual explanations. We will continue Example 1 and assume the classifier is a regular $AA-CBR_{\succeq}$ model mined from D . For illustrative reasons, we will assume this classifier is treated as a black-box model. As for the counterfactual explanations, we will not assume a particular algorithm, but we will assume it*

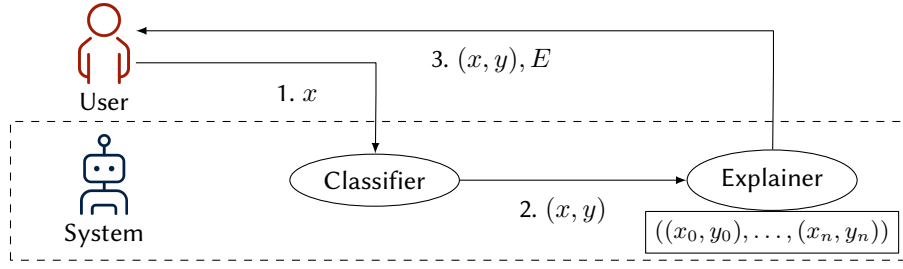


Figure 2: Overview of the *interactive explanation* process. 1. The user queries the system with an input x , which goes to the classifier. 2. The classifier produces output y , and sends the pair (x, y) to the explainer. 3. The explainer produces an explanation E for (x, y) , using information from the history of inputs/outputs $((x_0, y_0), \dots, (x_n, y_n))$, and sends (x, y) and E back to the user, who may then stop or further query the system.

finds a counterfactual by heuristical search [7, 8]. Besides, we will interpret it with a minimality assumption: a counterfactual is taken to be minimally different from the input, corresponding to the idea that it is as close as possible [7].

This way, for a single input-output pair (x, y) , the method returns a new input-output pair (x_{cf}, y_{cf}) . For describing the intended interpretation is: let $(\Delta_a, \Delta_b) = x - x_{cf}$, and (Δ'_a, Δ'_b) such that $0 \leq \Delta'_a \leq \Delta_a$, and $0 \leq \Delta'_b \leq \Delta_b$. For any such (Δ'_a, Δ'_b) , $x' = x + (\Delta'_a, \Delta'_b)$ is intuitively a smaller change to x than x_{cf} , and thus is expected to be classified as y , and not as y_{cf} .

For example, for a case $x_0 = \{\text{age}:50; \text{income}:50\}$ the output is $y_0 = 1$, if the counterfactual is $e_0 = (\{\text{age}:75; \text{income}:43\}, 0)$, then for input $x_1 = \{\text{age}:65; \text{income}:45\}$ the expected output would still be 1.

Now suppose a user interacts with the system querying for x_0 , receives y_0 as output and e_0 as an explanation. What would be expected if the user queries for x_1 , but instead receives $y_1 = 0$ and $e_1 = (\{\text{age}:59; \text{income}:43\}, 1)$? Is this an inconsistency of the method? Could it be solved?

3.2. A formal model of the interactive process

We assume an input set X and an output set Y as well as a set of possible explanations \mathcal{E} , that we keep abstract. Regarding notation, for a set S , we denote the set of finite sequences of elements of S as $Seq(S)$, i.e., $Seq(S) = \bigcup_{i \in \mathbb{N}} S^i$ for S^i a sequence of i elements of S . Given $n \in \mathbb{N}$, we use the notation $[n] = \{m \in \mathbb{N} \mid m \leq n\}$. Thus a sequence $(s_0, s_1, \dots, s_n) \in Seq(S)$ can be written as $(s_i)_{i \in [n]}$.

We consider that the system is composed of a classifier $\mathbb{C} : X \rightarrow Y$ and an explanation method $\mathbb{E} : Seq(X \times Y) \rightarrow Seq(\mathcal{E})$, mapping from a sequence of input-output pairs $(x_i, y_i)_{i \in [n]}$ to a sequence of explanations $(E_i)_{i \in [n]}$, of the same length as the sequence of pairs. Notice that the explainer uses information on the entire past of inputs-outputs, as we discussed in Section 3.1. We can think of this sequence of pairs as a *history*. We consider that at each time step $t \in \mathbb{N}$, the user queries for an input $x_t \in X$, which receives a classification $\mathbb{C}(x_t) = y_t$ and an explanation E_t . In this way, the explainer provides an explanation motivated by a specific input-output, while considering the history $((x_i, y_i))_{i \in [t-1]}$.

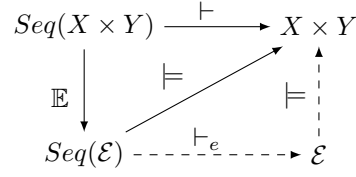


Figure 3: Diagram illustrating the relations between sequences of input-output pairs, sequences of explanations, single pairs and single explanations. This diagram does not always commute (indicated by dashed lines) since for a sequence of explanations \vdash -entailing an input-output pair, there may not exist a single explanation \vdash_e -entailed by this sequence which \models -entails this pair.

An important particular case is when there is a function $\mathbb{E}_\bullet : X \times Y \rightarrow \mathcal{E}$, mapping from a single example (x, y) to an explanation E . In this case, the explainer function \mathbb{E} can be defined as applying \mathbb{E}_\bullet to each element of the sequence: formally, $\mathbb{E}((x_i, y_i)_{i \in [n]}) = (\mathbb{E}_\bullet(x_i, y_i))_{i \in [n]}$. In this particular case the history is disregarded when explanations are computed.

This view of the interactive explanation process does not enforce that previously exhibited explanations are kept, that is, that E_i is unchanged for all $i \in [t]$ when x_{t+1} is queried. Past explanations being unretractable, in the sense that the system cannot replace the explanation of any previous query, is captured by the following property:

Definition 5 (Interaction-stability). *An explainer \mathbb{E} is said to be interaction-stable whenever, for every sequence of input-output pairs $(x_i, y_i)_{i \in [n]}$ and for every $m < n$, if $(E_i)_{i \in [n]} = \mathbb{E}((x_i, y_i)_{i \in [n]})$ and $(E'_i)_{i \in [m]} = \mathbb{E}((x_i, y_i)_{i \in [m]})$ then $E_i = E'_i$ for any $i \in [m]$.*

That is, an interaction-stable explainer will always keep the explanation E_i associated to the pair (x_i, y_i) , even as the interaction moves on. It is straightforward to see that an explainer \mathbb{E} derived from a function \mathbb{E}_\bullet is always interaction-stable.

With this setup, inference can be defined. We assume that a sequence of explanations $(E_i)_{i \in [n]}$ “commits” to some model behaviour. We model this by an entailment relation \models between $Seq(\mathcal{E})$ and $X \times Y$, in such a way that $(E_i)_{i \in [n]} \models (x, y)$ means that $(E_i)_{i \in [n]}$ “commits” to the outcome y , given the input x . We will abuse notation and define $E \models (x, y)$ to mean $(E) \models (x, y)$ (for (E) the sequence with just one explanation, E). This entailment relation we keep abstract and application-dependent. What is important is that it captures how a user would interpret the explanation or plausible inferences therefrom, including as regards the input-output being explained. One example is explanations as sufficient reasons: any sufficient reason is exactly a rule that guarantees the output for a part of the input space, including the given input. An important particular case of entailment is when it does not depend on the order of the elements of the sequence. In this case, a set-based representation would be enough, and it is in this sense that sequences generalise sets.

From this core notion of \models , relating explanations to input-output, we can derive two “homogeneous” notions of “entailment”, that is, from sequences of elements of a set to elements of the same set. This makes that notion more analogous to the notion of entailment in logic, which is defined from sets of formulas to a single formula. One such notion is at input-output level, and the other at explanation level. For the former, we say $(x_i, y_i)_{i \in [n]} \vdash (x, y)$ iff

$\mathbb{E}((x_i, y_i)_{i \in [n]}) \models (x, y)$. For the latter, $(E_i)_{i \in [n]} \vdash_e E$ iff $\forall (x, y) \in X \times Y$, if $E \models (x, y)$ then $(E_i)_{i \in [n]} \models (x, y)$ (summarised in Fig. 3).

3.3. Consistency and non-monotonicity

We shall now define properties of explainers, in particular of *consistency* and *non-monotonicity* for the relations associated with explainers.

Definition 6 (Consistency). *A sequence of explanations $(E_i)_{i \in [n]}$ is said to be consistent iff there does not exist $x \in X, y, y' \in Y$, with $y \neq y'$, such that $(E_i)_{i \in [n]} \models (x, y)$ and $(E_i)_{i \in [n]} \models (x, y')$. An entailment relation \models is said to be consistent iff every sequence of explanations is consistent. A relation \vdash is said to be consistent iff there does not exist $x \in X, y, y' \in Y$, with $y \neq y'$, and $((x_i, y_i))_{i \in [n]}$ such that $((x_i, y_i))_{i \in [n]} \models (x, y)$ and $((x_i, y_i))_{i \in [n]} \models (x, y')$.*

Since the relations \vdash and \vdash_e , derived from the base notion of \models , are “homogeneous”, we can define properties borrowed from the literature on non-monotonic reasoning [9, 10], what would not be possible for the relation \models . We only generalise them to sequences, instead of sets (as typical). Formally, some properties are:

Definition 7 (Non-monotonicity). *The relation \models is said to be non-monotonic iff there is $(E_i)_{i \in [n]}, E_{n+1}$ and (x, y) such that $(E_i)_{i \in [n]} \models (x, y)$ and $(E_i)_{i \in [n+1]} \not\models (x, y)$.*

Also, given a set S , a relation \vdash' from $Seq(S)$ to S , and $s, s_i \in S$, for $i \in \mathbb{N}$, the relation \vdash' is said to satisfy non-monotonicity iff there is $(s_i)_{i \in [n]}, s_{n+1}, s$ s.t. $(s_i)_{i \in [n]} \vdash' s$ and $(s_i)_{i \in [n+1]} \not\vdash' s$.

Which concrete methods satisfy which of those properties and what is the impact on user experience are open questions.

Example 2 (continued). *An instantiation for counterfactuals as presented in this example is that an explanation is a tuple of original input, output, and counterfactual and counterfactual output (x, x_{cf}, y_{cf}) . Thus the explainer \mathbb{E} is interaction-stable, based on a \mathbb{E}_\bullet that, given a single input-output pair, returns this tuple. The entailment relation is then defined as: for a single explanation $E = (x, y, x_{cf}, y_{cf})$, $E \models (x', y')$ iff:*

1. $x' = x_{cf}$ and $y' = y_{cf}$;
2. $x' = x$ and $y' = y$; or
3. with $(\Delta_a, \Delta_b) = x - x_{cf}$ and $(\Delta'_a, \Delta'_b) = x - x'$, $0 \leq \Delta'_a \leq \Delta_a$, and $0 \leq \Delta'_b \leq \Delta_b$ and $y' = y'$.

For a sequence of explanations, a naïve aggregation can be used: $(E_i)_{i \in [n]} \models (x', y')$ iff there is i such that $E_i \models (x', y')$. Using this definition, the behaviour in Example 2 is inconsistent.

An alternative method could aggregate restricting by a specificity criterion: say E covers x' iff there is y' such that $E \models (x', y')$. We say that E_1 is more specific than E_2 if the set of inputs covered by E_1 is a subset of the set of inputs covered by E_2 . Then redefine for sequences in the following way: $(E_i)_{i \in [n]} \models (x', y')$ iff there is i such that $E_i \models (x', y')$ and there is no E_j such that $E_j \models (x', y'')$, where $y' \neq y''$. By this definition, even if $e_0 \models (x_1, 1)$, $(e_0, e_1) \models (x_1, 0)$ and $(e_0, e_1) \not\models (x_1, 1)$. This would make such relation consistent and non-monotonic.

4. Arbitrated dispute trees

Dispute trees [11, 12] have been defined as explanations to *AA-CBR* since Čyras et al. [2], and further developed in the form of *arbitrated* dispute trees for an extended version of it in Čyras et al. [3]. This last form is the one which will interest us, due to its symmetry regarding default and non-default outcomes. For compatibility with our presentation, we will redefine the main concepts, but omit results proven originally, since the adaptations are minimal.

Definition 8. Let $AF_{\succeq}(D, x) = (Args, \rightsquigarrow)$. An **arbitrated dispute tree** (ADT) is a tree \mathcal{DT} such that:

1. every node of \mathcal{DT} is of the form $[N:\alpha]$ for $N \in \{W, L\}$ and $\alpha \in Args$, any such node being called an N -node labelled by argument α ;
2. the root of \mathcal{DT} is labelled by the default argument (δ_C, δ_o) and is a W -node, if $AA-CBR_{\succeq}(D, x) = \delta_o$; and a L -node otherwise;
3. for every W -node n labelled $\alpha \in Args$ and for every β attacker of α in $(Args, \rightsquigarrow)$, there is a child m of n such that m is a L -node labelled by β ;
4. for every L -node n labelled by $\alpha \in Args$, there is exactly one child which is a W -node labelled by some β attacker of α ; and
5. there are no other nodes in \mathcal{DT} .

We will also refer to \mathcal{DT} as an *arbitrated dispute tree for (the prediction for) x* .

An example of a dispute tree is in Figure 4a. Dispute trees have been originally defined as being possibly infinite, while Čyras et al. [3] assumes casebases are coherent, and one can thus prove every arbitrated dispute tree in that context is finite². While we make no such assumption, we consider infinite dispute trees to be inappropriate for explaining, so we retain the original definition but explicitly define ADT explanations to limit ourselves to the finite case.

Definition 9. An **arbitrated dispute tree explanation** is a finite arbitrated dispute tree.

5. Arbitrated dispute trees as interactive explanations

While arbitrated dispute trees have been used as explanations in previous literature [3], here we instantiate our framework for interactive explanations as ADTs. Our main question is: given an ADT, to what model behaviour does it “commit”? What should the \models relation be for them?

ADTs are way of presenting and explaining the behaviour of *AA-CBR* as a more succinct structure than the entire argumentation framework. Our idea here is simply that such presentation can be generalised to other cases: intuitively, an ADT for a new case x may show a set of sufficient reasons for justifying a decision for another new case x_2 . In a sense, x_2 does not need more information than was already presented in that ADT in order to be decided, therefore the rest of the AF is unnecessary. This can be verified simply by the structure of the decision tree and by checking for relevance. Let us first define this formally.

²This not being the case, some results from the original presentation are inapplicable, such as [3, Prop. 3.4].

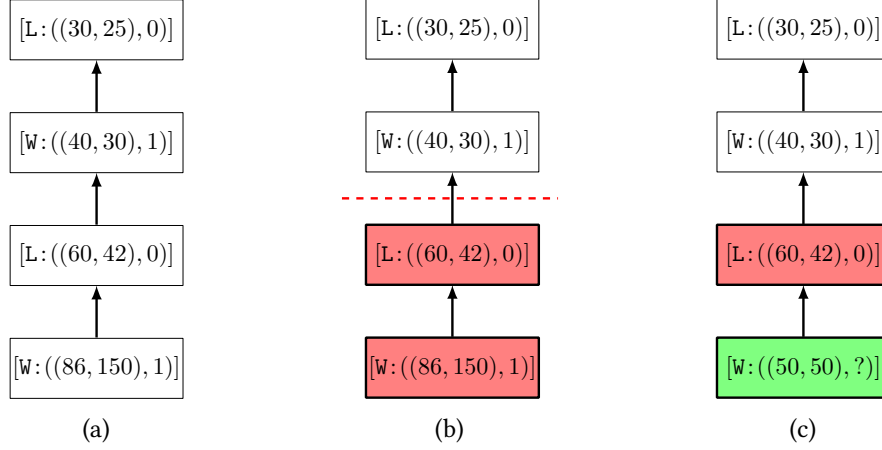


Figure 4: Example of ADT explanation in Example 4. Figure 4a shows the ADT explanation for input $x_2 = \{\text{age}:90; \text{income}:200\}$. Figures 4b and 4c show the process of adapting it to the new input $x_0 = \{\text{age}:50; \text{income}:50\}$. In Figure 4b, in red are the nodes labelled by arguments which are irrelevant to x_0 . There is a single leaf, which is a W node. Therefore this ADT can be adapted. This is done in Figure 4c, with the (single) attacker of the leaf added back, but now attacked by the new case.

5.1. Readaptation of ADTs

Let D be a dataset, and $(Args, \rightsquigarrow) = AF_{\geq}(D)$ be its corresponding argumentation framework. Let x be a new case, and \mathcal{DT} be an ADT explanation for its prediction. Now let x_2 be a second new case. Depending on the partial order relation, on the ADT and on the new cases, it might be possible to reuse the ADT \mathcal{DT} , without looking at $(Args, \rightsquigarrow)$ directly, to decide the outcome for x_2 . Indeed, in that case another ADT would be generated as well.

The two core ideas are: 1) if a past case is relevant to the new case, then every case smaller than it is also relevant; 2) every attacker of a W-node is in an ADT (as a L-node).

Theorem 10. *Let \mathcal{DT}' be \mathcal{DT} with all nodes labelled by x_1 removed. For every leaf l of \mathcal{DT}' , let max_l be the node in the path from the root to l which is maximally relevant to x_2 (that is, there is no node in the path greater than it such that it is also relevant to l).*

If all max_l are W-nodes, then the predicted outcome for x_2 is the same as the predicted outcome for x . Besides, let \mathcal{DT}_{x_2} be the tree constructed by the following process: start with the subtree of \mathcal{DT}' containing only max_l and their ancestors, add all L-nodes which are children of max_l in \mathcal{DT}' , and, finally, for each L-node added in this way, add as a child a new W-node labelled by x_2 . Then \mathcal{DT}_{x_2} is an ADT for the prediction on x_2 .

When this is the case for new cases x and x_2 , we will say \mathcal{DT} can be **readapted** to x_2 . Before proving the theorem, let us illustrate the process:

Example 3. *Given our casebase D , suppose we would like to have the prediction for $x_2 = \{\text{age}:90; \text{income}:200\}$ and an ADT explanation. One would notice that the outcome is $y_2 = 1$ with the ADT in Figure 4a. The idea is, if one would like to predict for $x_0 = \{\text{age}:50; \text{income}:50\}$, this ADT could be readapted in a straightforward way. The processed is illustrated in the remaining of Figure 4.*

Proof of Theorem 10. For every l , max_l exists: by regularity, the default case (thus the root) is relevant to x_2 . Let us see that \mathcal{DT}_{x_2} is an ADT. Clearly the condition 1 is satisfied. Let us check conditions 3 and 4 for each node. For every branch, every node until max_l satisfies the conditions in Definition 8, since \mathcal{DT} is an ADT. Now consider max_l . By assumption, every max_l is a W-node. Again, since \mathcal{DT} is an ADT for x , for every max_l , each child of it is included in \mathcal{DT} as a L-node child. Since those are also in \mathcal{DT}_{x_2} , the condition is satisfied for each max_l . Next, each of those attackers require exactly one W-node as a child, attacking it. This is satisfied by every added W-node labelled by x_2 , which is an attacker since the arguments which label such L-nodes are irrelevant (otherwise max_l would not be maximally relevant). Finally, since x_2 has no attackers in $AF_{\succeq}(D, x_2)$, it satisfies the conditions of a leaf. Condition 5 is clearly satisfied.

The last requirement to check is whether condition 2 is satisfied. Indeed it is, since, given that the other conditions are satisfied, then the set of arguments labelling W-nodes is in the grounded extension of $AF_{\succeq}(D, x_2)$, which can be verified by induction [3, Prop. 3.3]. Therefore if the root is a W-node, it is in the grounded extension and thus the prediction is δ_o . Otherwise, it is a L-node and then it has as a child which is a W-node, that is, the default argument is attacked by the grounded extension (and thus not in it, since it is conflict-free) and so the prediction of x_2 is the $\bar{\delta}_o$. \square

5.2. The \models relation for dispute trees

Now we can instantiate our framework with dispute trees. We will define \mathbb{E}_{\bullet} as a function that receives a new case x and its predicted outcome $AA-CBR_{\succeq}(D, x) = y$, and returns an ADT explanation \mathcal{DT} for it. While the choice of the decision tree could matter for practical applications, for our purposes our concern is only that some decision tree is returned. Regarding existence, it requires the argumentation frameworks to be acyclic. This can be achieved by restricting ourselves to coherent casebases, or instead applying $cAA-CBR_{\succeq}$, a variation of the original method which guarantees acyclic AFs [5]. For the rest of the section we will assume those conditions.

We can then easily instantiate the remaining of the framework, extending for sequences as previously suggested $\mathbb{E}((x_i, y_i)_{i \in [n]}) = (\mathbb{E}_{\bullet}(x_i, y_i))_{i \in [n]}$, and defining \models as: $(\mathcal{DT}_i)_{i \in [n]} \models (x, y)$ iff there is i such that \mathcal{DT}_i can be readapted to x . Indeed, this readapted tree \mathcal{DT}_x is precisely an explanation such that $\mathcal{DT}_i \vdash_e \mathcal{DT}_x$.

The intuition for this relation is a possible answer to the question: what does an ADT explanation say about other inputs, beyond the originally one being explained? Surely it also reveals how the cases which label the appearing nodes are decided, but would that be all? What we show from Theorem 10 is precisely that, for a new input case, if, when filtering the tree for only nodes labelled by relevant arguments, all leaves are W-nodes, then not only we can guarantee what the output is, but also acquire an explanation for it via a transformation. Also notice that this all is done without running the prediction algorithm entirely from scratch. How could this be used to accelerate inference time in practice is outside the scope of this paper.

Another interesting aspect to notice is that, if there is a single L-node as a maximally relevant node, then nothing can be said. This can be seen from Figure 4. For a new case $\{\text{age} : 19; \text{income} : 110\}$, the only relevant case is the root (the default case), which is a L-case.

However the prediction for it is still 1 since the case $(\{\text{age} : 20; \text{income} : 100\}, 1)$ is relevant, and would attack the default case in the grounded extension. The slightly different new case $\{\text{age} : 21; \text{income} : 110\}$ would have no such attacker.

We can then state the following consequences:

Theorem 11. *Let $\mathbb{E}((x_i, y_i)_{i \in [n]}) = (E_i)_{i \in [n]}$, and $(x, y) \in X \times Y$ Then:*

1. \mathbb{E} is interaction-stable;
2. $(E_i)_{i \in [n]} \models (x, y)$ iff there is an explanation E such that $(E_i)_{i \in [n]} \vdash_e E$ and $E \models (x, y)$;
3. (faithfulness) if $(E_i)_{i \in [n]} \models (x, y)$, then $AA-CBR_{\succeq}(D, x) = y$.
4. \models is consistent;
5. \models is monotonic.

Those are straightforward from \models being defined from existence of an explanation in the sequence and from the fact that all arguments labelling W-nodes are contained in the grounded extension [3, Prop. 3.3].

ADT explanations are simple, but provably faithful explanations for *AA-CBR*. This should not be surprising, since this inference relation is conservative: a single L-node as a maximally relevant node in the tree makes the entire ADT unusable.

6. Discussion and conclusion

A possible discussion is whether the entire set of ADTs for a input could not be returned instead. While this is a possibility, likely for a practical application all this information would be used, but still adapted and filtered in a way not to avoid overwhelming the final user. This is especially the case since many different dispute trees could have overlaps, being partly redundant. Depending on context, this could be less informative than having the entire argumentation framework, since although not a tree, there is no repetition of cases in there. Thus having a single ADT seems a more effective strategy. The condition is, as previously mentioned, that either the casebase is coherent, or that *cAA-CBR_⊆* is used [5], since there is also a correspondence in that given an outcome, there is always a dispute for explaining that decision [3, Prop.3.4].

We leave for future work an exploration of how would ADTs be for real applications of *AA-CBR*, including how often ADTs can be reused, and whether this can create gains in inference time. Another aspect left for future work is user studies for evaluating the effectiveness of ADT as explanations. An interesting question is whether ADTs with maximally relevant L-nodes could be readapted in some way, even if fallible. An idea would be considering those *prima facie* explanations, resulting in a non-monotonic entailment relation from sequences of explanations.

Acknowledgments

GPP was supported by Capes (Brazil, Ph.D. Scholarship 88881.174481/2018-01). FT was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101020934) and by J.P. Morgan

and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme. Any views or opinions expressed herein are solely those of the authors.

References

- [1] G. Paulino-Passos, F. Toni, On interactive explanations as non-monotonic reasoning, in: Workshop on Explainable Artificial Intelligence (XAI) at IJCAI 2022, 2022.
- [2] K. Čyras, K. Satoh, F. Toni, Explanation for case-based reasoning via abstract argumentation, in: P. Baroni, T. F. Gordon, T. Scheffler, M. Stede (Eds.), Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016, volume 287 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2016, pp. 243–254. doi:10.3233/978-1-61499-686-6-243.
- [3] K. Čyras, D. Birch, Y. Guo, F. Toni, R. Dulay, S. Turvey, D. Greenberg, T. Hapuarachchi, Explanations by arbitrated argumentative dispute, *Expert Syst. Appl.* 127 (2019) 141–156. doi:10.1016/j.eswa.2019.03.012.
- [4] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995) 321 – 357.
- [5] G. Paulino-Passos, F. Toni, Monotonicity and noise-tolerance in case-based reasoning with abstract argumentation, in: M. Bienvenu, G. Lakemeyer, E. Erdem (Eds.), Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021, 2021, pp. 508–518. doi:10.24963/kr.2021/48.
- [6] O. Cocarascu, A. Stylianou, K. Čyras, F. Toni, Data-empowered argumentation for dialectically explainable predictions, in: ECAI 2020 - 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 10-12 June 2020, 2020.
- [7] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, *Harvard Journal of Law & Technology* 31 (2018).
- [8] M. Hildebrandt, C. Castillo, L. E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (Eds.), FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, ACM, 2020. doi:10.1145/3351095.
- [9] D. Makinson, General patterns in nonmonotonic reasoning, in: D. M. Gabbay, C. J. Hogger, J. A. Robinson (Eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming - Volume 3 - Nonmonotonic Reasoning and Uncertain Reasoning*, Oxford University Press, 1994, pp. 35–110.
- [10] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, *Artif. Intell.* 44 (1990) 167–207. doi:10.1016/0004-3702(90)90101-5.
- [11] P. M. Dung, R. A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, *Artif. Intell.* 170 (2006) 114–159. doi:10.1016/j.artint.2005.07.002.
- [12] P. M. Dung, P. Mancarella, F. Toni, Computing ideal sceptical argumentation, *Artif. Intell.* 171 (2007) 642–674. doi:10.1016/j.artint.2007.05.003.