# Constructing and Explaining Case Models: A Case-based Argumentation Perspective

Wachara Fungwacharakorn[1[0000−0001−9294−3118]], Ken Satoh[1[0000−0002−9309−4602]], and Bart Verheij[2[0000−0001−8927−8751]]

[1] National Institute of Informatics and SOKENDAI, Tokyo, Japan
{wacharaf,ksatoh}@nii.ac.jp
[2] Artificial Intelligence, University of Groningen, Groningen, The Netherlands
bart.verheij@rug.nl

**Abstract.** In this paper, we investigate constructing and explaining on case models, which have been proposed as formal models for presumptive reasoning and evaluating arguments from cases. Recent research shows applications of case models and relationships between case models and other computational reasoning models. However, formal methods for constructing and explaining case models have not been investigated yet. Therefore, in this paper, we present methods for constructing and explaining case models based on the formalism of abstract argumentation for case-based reasoning (AA-CBR). The methods are illustrated in this paper with a legal example of paying penalties for a delivery company. We found that we can constructed case models can provide model-theoretic semantics equivalent to AA-CBR, and that explanations in case models can be made by dispute trees as in AA-CBR.

**Keywords:** case-based reasoning · argumentation frameworks · case models

## 1 Introduction

Artificial Intelligence and Law researchers are interested in explanations of reasons using cases. In order to explain reasons, early case-based legal reasoning systems, such as HYPO, use analogical reasoning [3]. Connections have been made to argumentation models, such as ASPIC+ [15], building on formal developments in argumentation theory [7, 12, 21]. Argumentation has been shown to be useful for explanation [2], and case-based argumentation has been shown to be useful not only for explaining from precedent cases [6], but also explaining the development of case law [10] as well as explaining legal theories based on hypothetical cases in statute law [16].

On the other hand, Artificial Intelligence and Law researchers are also interested in evaluations of arguments using cases. Case models [19] have been recently developed in order to formally evaluate arguments. Each case model consists of a set of consistent, mutually incompatible, and different logic formulas representing cases, and a total and transitive preference ordering over the cases. Case models evaluate arguments as incoherent, coherent, presumptively valid, and conclusive. Several applications of case models have been investigated, including evidential reasoning [11, 20] and ethical system design [18]. Formalizing case models for case-based reasoning has also been investigated [18, 19]. However, the questions of how to formally construct case models from a case-base and how to explain argument moves in case models have not been studied.

In order to address these questions, we investigate case models from a case-based argumentation perspective. As a representative of case-based argumentation, we consider abstract argumentation for case-based reasoning (AA-CBR) [5], which inspires explanations in precedential constraint [14, 22]. A case base in AA-CBR is a finite set of cases with outcome – including a default case with outcome, which is a pair of a default situation represented as the empty set and a predefined default outcome. Given a new situation, AA-CBR infers an outcome by forming a corresponding argumentation framework [7] with respect to the case base, and determining whether or not the default case with outcome is included in the grounded extension of the corresponding argumentation framework. AA-CBR explains the inference using dispute trees with respect to the default case with outcome. By exploring the relation between AA-CBR and case models, we present a method of constructing case models from an AA-CBR case base. Furthermore, we extend several concepts in case models to explain argument moves in case models using dispute trees as in AA-CBR. We show that dispute trees used in case models are homomorphic to dispute trees used in AA-CBR.

This paper is structured as follows. Section 2 describes abstract argumentation for case-based reasoning (AA-CBR), which is a representative of case-based argumentation formalism used in this paper. Section 3 describes case models. Section 4 presents the first contribution of formalizing a method for constructing case models from AA-CBR case bases. Then, Section 5 presents the second contribution of developing dispute trees for explanations in case models. Section 6 discusses connections with related research, and provides suggestions for future work. Finally, Section 7 provides the conclusion of this paper.

## 2 Abstract Argumentation for Case-based Reasoning

In this section, we focus on abstract argumentation for case-based reasoning (AA-CBR) [5], which is used as a representative of case-based argumentation in this paper. AA-CBR aims to formalize reasoning from consistent cases with outcome and a predefined default outcome using Dung's abstract argumentation frameworks [7]. Recently, AA-CBR has been developed for general representations of situations and preferences [4], but in this paper, we follow the original one [5] and represent the situation of a case as a finite set of attributes. The original paper calls those attributes *factors* but we use in this paper *fact propositions* in order to distinguish them from those in CATO. We define the set $\mathcal{F}$ of all possible fact propositions called the *fact-domain*. A case base in AA-CBR is then defined as follows [5].

**Definition 1 (Case-base in AA-CBR).** *Let $\mathcal{F}$ be the* fact-domain*. A case with outcome in AA-CBR is a pair $(X, o)$ where $X \subseteq \mathcal{F}$ representing the fact situation of the case and $o \in \{+, -\}$ representing the outcome of the case. We denote the opposite of $o$ by $\bar{o}$, namely $\bar{o} = +$ if $o = -$; and $\bar{o} = -$ if $o = +$. A* case-base*, denoted with $CB$, is a finite set of cases with outcome that assumes consistency, i.e. for $(X, o_x), (Y, o_y) \in CB$, if $X = Y$, then $o_x = o_y$, and contains a* default *case (with outcome) $(\emptyset, d)$ where $d \in \{+, -\}$ is a default outcome. We denote a set of all propositions occurring in $CB$ by $\mathcal{F}_{CB}$, i.e. $\mathcal{F}_{CB} = \bigcup_{(X,o) \in CB} X$.*

*Example 1.* To illustrate case-based argumentation, we adapt an example of penalties from a delivery company [1] with the following rules.

1. If there is no special situation, the delivery company does not have to pay a penalty.
2. If the items were delayed, the delivery company has to pay a penalty.
3. If the items were damaged, the delivery company has to pay a penalty.
4. If the items were damaged but they are fungible and the items were replaced, then the delivery company does not have to pay a penalty.

We represent facts with the following propositions.

- `delayed`: the items were delayed.
- `damaged`: the items were damaged.
- `fungible`: the items are fungible
- `replaced`: the items were replaced.
- `penalty`; the delivery company has to pay a penalty

Considering a conclusion of whether the delivery company has to pay a penalty (+ means the company has to pay a penalty; − otherwise), the working example can be represented as a case base $CB$, consisting of cases with outcome as follows.

1. $co_0 = (\emptyset, -)$
2. $co_1 = (\{\texttt{delayed}\}, +)$
3. $co_2 = (\{\texttt{damaged}\}, +)$
4. $co_3 = (\{\texttt{damaged}, \texttt{fungible}, \texttt{replaced}\}, -)$

To infer an outcome for a new fact situation $N$, AA-CBR forms an abstract argumentation framework [7] corresponding to $N$ and considers whether or not the default case $(\emptyset, d)$ is in the grounded extension of the argumentation framework.

**Definition 2 (Abstract argumentation in AA-CBR).** *The AA framework corresponding to a case base $CB$ with a default case $(\emptyset, d)$ and a new fact situation $N$ is $(AR, attacks)$ satisfying the following conditions:*

- $AR = \{(X, o) \in CB | X \subseteq N\}$ [3]

[3] The original AA-CBR [5] uses $(N, ?)$ that attacks all cases with outcome of which situations are not subsets of $N$, but, to simplify definitions in the rest of the present paper, we adapt this part of the definition following [1] instead.

– $(X, o_x)$ *attacks* $(Y, o_y)$ *for all pairs* $(X, o_x)$, $(Y, o_y) \in AR$ *such that*
  - $o_x \neq o_y$ , *and (different outcomes)*
  - $Y \subsetneq X$, *and (specificity)*
  - $\nexists (Z, o_x) \in AR$ *with* $Y \subsetneq Z \subsetneq X$ *(concision)*

*The* AA outcome *of the new fact situation* $N$ *is* $d$ *if* $(\varnothing, d)$ *is in the grounded extension of the argumentation framework, otherwise the* AA outcome *of the new fact situation* $N$ *is* $\bar{d}$. *We denote the AA framework in which arguments are all cases with outcome in* $CB$ *as* $(CB, attacks)$.

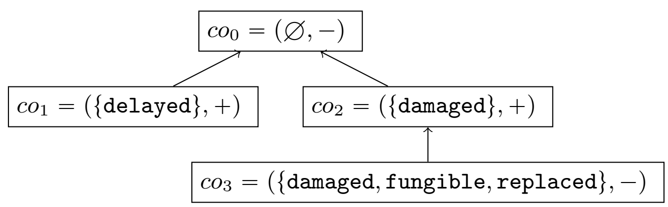


**Fig. 1.** Argumentation framework corresponding to the working example

Suppose $N_1 = \{\texttt{delayed}, \texttt{damaged}, \texttt{fungible}, \texttt{replaced}\}$, representing a case where items were damaged, the damaged items are fungible, the items were replaced, but they were delayed. We have that AA-CBR uses all cases in the case base to form an argumentation framework, as illustrated in Figure 1. We can see that the default case $co_0 = (\varnothing, -)$ is not in the grounded extension of the argumentation framework. Thus, the AA outcome of $N_1$ is $+$, i.e. the delivery company has to pay a penalty.

AA-CBR provides explanations using *admissible* and *maximal* dispute trees, which are extended from dispute trees used in dialectic proof procedures [8, 9]. The definition of dispute trees used in AA-CBR [5] is as follows.

**Definition 3 (Dispute Tree).** *Let* $(AR, attacks)$ *be an argumentation framework. A* dispute tree *for* $a \in AR$ *is a (possibly infinite) tree* $\mathcal{T}$ *such that:*

1. *every node of* $\mathcal{T}$ *is of the form* $[L : x]$, *with* $L \in \{P, O\}$ *and* $x \in AR$ *where* $L$ *indicates the status of proponent (*$P$*) or opponent (*$O$*);*
2. *the root of* $\mathcal{T}$ *is a* $P$ *node labelled by* $a$*;*
3. *for every* $P$ *node* $n$*, labelled by some* $b \in AR$*, and for every* $c \in AR$ *such that* $c$ *attacks* $b$*, there exists a child of* $n$*, which is an* $O$ *node labelled by* $c$*;*
4. *for every* $O$ *node* $n$*, labelled by some* $b \in AR$*, there exists at most one child of* $n$*, which is a* $P$ *node labelled by* $c \in AR$ *such that* $c$ *attacks* $b$*;*
5. *there are no other nodes in* $\mathcal{T}$ *except those given by 1-4.*

*A dispute tree* $\mathcal{T}$ *is an* admissible *dispute tree if and only if (i) every* $O$ *node in* $\mathcal{T}$ *has a child, and (ii) no argument in* $\mathcal{T}$ *labels both* $P$ *and* $O$ *nodes. A dispute tree* $\mathcal{T}$ *is a* maximal *dispute tree if and only if for all opponent nodes* $[O : x]$ *which are leaves in* $\mathcal{T}$ *there is no* $y \in AR$ *such that* $y$ *attacks* $x$.

Admissible dispute trees are maximal dispute trees but not vice versa [5] because admissible dispute trees are those maximal dispute trees without opponent leaves while maximal dispute trees with opponent leaves also exist. In other words, admissible dispute trees demonstrate argumentations where the proponent can attack all of the opponent's arguments but maximal dispute trees demonstrate argumentations where the proponent's burden is *complete*, i.e. either the proponent cannot attack some opponent's arguments or the proponent already attacks all of the opponent's arguments. Hence, AA-CBR uses dispute trees for explanations as follows.

**Definition 4 (AA-CBR explanation).** *Let $N$ be a fact situation and $d$ be a default outcome. An* explanation *for why the AA outcome of $N$ is $d$ is any admissible dispute tree for $(\emptyset, d)$. An* explanation *for why the AA outcome of $N$ is $\bar{d}$ is any maximal dispute tree for $(\emptyset, d)$.*

We refer to a case with outcome that can occur in any maximal dispute tree for $(\emptyset, d)$ as a *critical* case with outcome [16]. In other words, $(\emptyset, d)$ is thus a critical case with outcome and any case with outcome that attacks a critical case with outcome is also a critical case with outcome.

From the working example, two maximal dispute trees can be extracted from the argumentation framework, as shown in Figure 2. The left tree in the figure is a non-admissible dispute tree which explains why the company has to pay a penalty. The dispute tree on the right in the figure is an admissible dispute tree which explains why the company does not have to pay a penalty, but it is overridden by the dispute tree on the left in the figure since it is non-admissible.

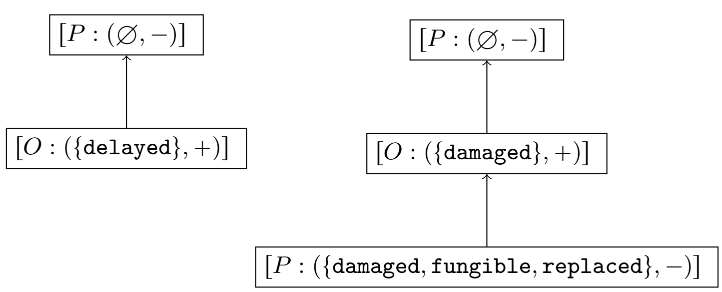


**Fig. 2.** Two Maximal Dispute Trees for $(\emptyset, -)$ with respect to the argumentation framework corresponding to the example

## 3 Case Models

Case models [19] aim to formally evaluate arguments from cases. A case in case models is a logical formula, usually a conjunction of literals. A case model consists of a set of cases $C$, and their preference ordering $\geqslant$. The cases in a case model must be logically consistent, mutually incompatible and different. The preference ordering $\geqslant$ in case models must be total and transitive (hence is what is called a total preorder, corresponding to a numerically representable ordering).

**Definition 5 (Case Model [19]).** *Let $\mathcal{L}$ be a classical logical language generated from a set of propositional constants in a standard way. We write $\neg$ for negation, $\wedge$ for conjunction, $\vee$ for disjunction, $\leftrightarrow$ for equivalence, $\top$ for a tautology, and $\bot$ for a contradiction. The associated classical, deductive, monotonic consequence relation is denoted $\models$. A case model is a pair $(C, \geqslant)$ with finite $C \subseteq \mathcal{L}$, such that the following hold, for all $\varphi$ , $\psi$ and $\chi \in C$:*

1. $\models \neg\varphi$ *(logically consistent);*
2. *If* $\not\models \varphi \leftrightarrow \psi$, *then* $\models \neg(\varphi \wedge \psi)$ *(mutually incompatible);*
3. *If* $\models \varphi \leftrightarrow \psi$, *then* $\varphi = \psi$ *(different);*
4. $\varphi \geqslant \psi$ *or* $\psi \geqslant \varphi$ *(total);*
5. *If* $\varphi \geqslant \psi$ *and* $\psi \geqslant \chi$, *then* $\varphi \geqslant \chi$ *(transitive).*

The strict weak order $>$ standardly associated with a total preorder $\geqslant$ is defined as $\varphi > \psi$ if and only if it is not the case that $\psi \geqslant \varphi$ (for $\varphi$ and $\psi \in C$). When $\varphi > \psi$ , we say that $\varphi$ is (strictly) preferred to $\psi$. The associated equivalence relation $\sim$ is defined as $\varphi \sim \psi$ if and only if $\varphi \geqslant \psi$ and $\psi \geqslant \varphi$. Case models evaluate arguments from cases as follows.

**Definition 6 (Argument Evaluation in Case Models [19]).** *An argument is a pair $(\varphi, \psi)$ with $\varphi$ and $\psi \in \mathcal{L}$ where $\varphi$ expresses the argument's premise and $\psi$ expresses the argument's conclusion. We say an argument $(\varphi, \psi)$ has grounding in case $\omega$ if and only if $\omega \models \varphi \wedge \psi$ and an argument $(\varphi, \psi)$ is relevant to case $\omega$ if $\omega \models \varphi$. Let $(C, \geqslant)$ be a case model. Then we define, for all $\varphi$ and $\psi \in \mathcal{L}$:*

- $(\varphi, \psi)$ *is* coherent *with respect to $(C, \geqslant)$ if and only if $(\varphi, \psi)$ has grounding in some cases, i.e.* $\exists \omega \in C : \omega \models \varphi \wedge \psi$.
- $(\varphi, \psi)$ *is* presumptively valid *with respect to $(C, \geqslant)$ if and only if $(\varphi, \psi)$ has grounding in a case that is maximal within the relevant cases, i.e.* $\exists \omega \in C : \omega \models \varphi \wedge \psi$; *and* $\forall \omega' \in C$ : *if* $\omega' \models \varphi$, *then* $\omega \geqslant \omega'$.
- $(\varphi, \psi)$ *is* conclusive *with respect to $(C, \geqslant)$ if and only if $(\varphi, \psi)$ has grounding in every relevant case, i.e.* $\exists \omega \in C : \omega \models \varphi \wedge \psi$; *and* $\forall \omega \in C$ : *if* $\omega \models \varphi$, *then* $\omega \models \varphi \wedge \psi$.

Attacks between arguments are defined in case models as follows [19].

**Definition 7 (Attack in Case Models).** *Let $\mathcal{L}$ be a classical logical language, $(C, \geqslant)$ be a case model, and $(\varphi, \psi)$ be a presumptively valid argument. Then $\chi \in \mathcal{L}$ is* defeating *or* successfully attacking *the argument when $(\varphi \wedge \chi, \psi)$ is not presumptively valid. A case $\omega \in C$ provides grounding for the attack if $\omega \models \varphi \wedge \chi$. Furthermore, if $(\varphi \wedge \chi, \neg\psi)$ is presumptively valid, $\chi$ is rebutting; otherwise, $\chi$ is undercutting. If $\varphi = \top$ then, $\chi$ is undermining.*

## 4 Constructing Case Models

In this section, we present our contribution of formalizing a method for constructing a case model from an AA-CBR case base. We found that the construction is related to a concept of nearest case with outcome in AA-CBR, which is defined as follows [6].

**Definition 8 (Nearest Case with Outcome).** *Let $N$ be a fact situation, and $CB$ be a case-base. $(X, o_x) \in CB$ is nearest to $N$ (not always unique) if and only if $X \subseteq N$, and $\nexists (Y, o_y) \in CB$ with $Y \subseteq N$ and $X \subsetneq Y$. In other words, $X$ is $\subseteq$-maximal in the case base.*

If there is a unique nearest case with outcome $(X, o)$ to a fact situation $N$, then the AA outcome of $N$ is $o$ [6], as illustrated in Example 2.

*Example 2.* Continuing from Example 1, let the fact-domain be $\mathcal{F} = \{$`delayed`, `damaged`, `fungible`, `replaced`$\}$, fact situations can be classified with their unique nearest case with outcome as follows.

- $co_0 = (\emptyset, -)$ is unique nearest to:
  $\emptyset$, {`fungible`},{`replaced`},{`fungible`, `replaced`}
- $co_1 = (\{$`delayed`$\}, +)$ is unique nearest to:
  {`delayed`}, {`delayed`, `fungible`}, {`delayed`, `replaced`},
  {`delayed`, `fungible`, `replaced`}
- $co_2 = (\{$`damaged`$\}, +)$ is unique nearest to:
  {`damaged`}, {`damaged`, `fungible`}, {`damaged`, `replaced`}
- $co_3 = (\{$`damaged`, , `fungible`, `replaced`$\}, -)$ is unique nearest to:
  {`damaged`, , `fungible`, `replaced`}
- No unique nearest case with outcome:
  {`delayed`, `damaged`}, {`delayed`, `damaged`, `fungible`},
  {`delayed`, `damaged`, `replaced`}, {`delayed`, `damaged`, `fungible`, `replaced`}

The unique nearest case with outcome is defeasible. A fact situation can be monotonically grown without changing its unique nearest case with outcome until it reaches *exceptional* conditions. Extending from the concept of nearest case with outcome, we define the boundary of case with outcome as follows.

**Definition 9 (Boundary).** *Let $\mathcal{F}$ be the* fact-domain, *$CB$ be a case-base with $(X, o_x)$, and $(CB, attacks)$ be an argumentation framework formed by AA-CBR. Let $CB_{\rightarrow(X,o_x)}$ be a set of all cases with outcome in $CB$ that attack $(X, o_x)$, i.e. $CB_{\rightarrow(X,o_x)} = \{(Y, o_y) \in CB | (Y, o_y)$ attacks $(X, o_x)\}$.*

- *A* boundary *of $(X, o_x)$, denoted by $F_{\rightarrow(X,o_x)}$, is a set of all propositions occurring in $CB_{\rightarrow(X,o_x)}$, i.e. $F_{\rightarrow(X,o_x)} = \mathcal{F}_{CB_{\rightarrow(X,o_x)}}$.*
- *An* internal sub-boundary *of $(X, o_x)$ is a subset of boundary that cover $X$ but does not cover a fact situation of a case with outcome that attacks $(X, o_x)$. We denote a set of all internal sub-boundary by $I_{\rightarrow(X,o_x)}$, i.e.*
  $I_{\rightarrow(X,o_x)} = \{B \subseteq F_{\rightarrow(X,o_x)} | X \subseteq B \wedge \nexists (Y, o_y) \in CB_{\rightarrow(X,o_x)} Y \subseteq B\}$.

In following definitions, we express a ternary conditional operator as $(a\ ?\ b : c)$, interpreted as if $a$ then $b$ otherwise $c$, and we define a naming function *name* that maps from every case with outcome to a set of name propositions $\mathcal{N}$ distinct from $\mathcal{F}$, i.e. $n : \mathcal{F} \times \{+, -\} \mapsto \mathcal{N}$. For ease of exposition, we use the same symbol for referring to the case with outcome and its name proposition.

Firstly, we define a function *case* based on an informal construction described in [19] for constructing a logical sentence from a critical case with outcome and one of its internal sub-boundary. The literals used for constructing the sentence are from five sources: (1) the outcome (2) the name propositions (3) the considering sub-boundary (4) propositions inside the boundary but outside the considering internal sub-boundary (5) propositions inside the casebase but outside the boundary of the default case. The function is formally defined as follows.

**Definition 10 (Case construction).** *Let $CB$ be a case base with a default case $(\varnothing, d)$ and a critical case with outcome $(X, o_x)$, $B_x \in I_{\rightarrow(X,o_x)}$, and $\delta$ be a proposition. $case(X, o_x, B_x, \delta)$ is a function defined as*

$$
\begin{aligned}
case(X, o_x, B_x, \delta) = \quad & (o_x = d\ ?\ \delta : \neg\delta) \ \wedge \bigwedge_{n \in \mathcal{N}} (n = name(X, o_x)\ ?\ n : \neg n) \ \wedge \\
& \bigwedge_{p_i \in B_x} p_i \ \wedge (I_{\rightarrow(X,o_x)} = \{X\}\ ?\ \top : \bigwedge_{p_k \in F_{\rightarrow(X,o_x)} \setminus B_x} \neg p_k) \wedge \\
& (X = \varnothing\ ? \bigwedge_{p_l \in \mathcal{F}_C B \setminus F_{\rightarrow(\varnothing,d)}} p_l : \top)
\end{aligned}
$$

Since an internal sub-boundary of $(X, o_x)$ has a unique nearest case, that is $(X, o_x)$, *case* is a one-to-one function, namely given the sentence constructed from *case*, we can trace back which case with outcome and which internal sub-boundary that the sentence is constructed from.

Secondly, we define a function *depth*, which is a mapping function from any case with outcome to an integer, expressing the depth of attacks from the default case to the considering case with outcome. This function is then used to determine the preference between cases. The function is defined as follows.

**Definition 11 (Attack depth).** *Let $CB$ be a case base with a default case $(\varnothing, d)$ and a critical case with outcome $(X, o_x)$, and $(CB, attacks)$ be an argumentation framework formed by AA-CBR. $depth(X, o_x)$ is a function defined as*

$$
depth(X, o_x) = \begin{cases} 0 & \text{if } X = \varnothing \\ 1 + \max\limits_{(X,o_x)\ attacks\ (Y,o_y)} depth(Y, o_y) & \text{otherwise} \end{cases}
$$

Using these two function, we present the following formal method for constructing case models as follows.

**Definition 12.** *Let $CB$ be a case base with a default case $(\emptyset, d)$ and $\delta$ be a proposition. We say a case model $(C, \geqslant)$ is constructed from $CB$ with respect to $\delta$ if and only if the following conditions hold.*

1. *for every critical $(X, o_x) \in CB$ and $B_x \in I_{\rightarrow(X,o_x)}$, there exists $case(X, o_x, B_x, \delta) \in C$; and*
2. *for every critical $(X, o_x), (Y, o_y) \in CB$, $B_x \in I_{\rightarrow(X,o_x)}$, and $B_y \in I_{\rightarrow(Y,o_y)}$ such that $c_1 = case(X, o_x, B_x, \delta), c_2 = case(Y, o_y, B_y, \delta) \in C$, $c_1 \geqslant c_2$ if and only if $depth(X, o_x) \leqslant depth(Y, o_y)$; and*
3. *there are no other cases in $C$ except those given by 1.*

Since *case* is a one-to-one function, the result of *case* are different from each other. With the layout of negations in the construction, cases in constructed case models are mutually incompatible. The preference ordering is total and transitive since it is derived from numeric comparisons.

**Table 1.** Constructing cases in case model from the working example

| Cases with outcome and boundaries | Internal sub-boundary | Cases in case model |
|---|---|---|
| $co_0 = (\emptyset, -)$ Boundary = {`delayed`, `damaged`} | $\emptyset$ | $c_0 : \delta \wedge co_0 \wedge \neg co_1 \wedge \neg co_2 \wedge \neg co_3$ $\wedge$ `fungible` $\wedge$ `replaced` |
| $co_1 = (${`delayed`}$, +)$ Boundary = {`delayed`} | {`delayed`} | $c_{1a} : \neg\delta \wedge \neg co_0 \wedge co_1 \wedge \neg co_2 \wedge \neg co_3$ $\wedge$ `delayed` |
| $co_2 =$ $(${`damaged`}$, +)$ Boundary = {`damaged`, `fungible`, `replaced`} | {`damaged`} {`damaged`, `fungible`} {`damaged`, `replaced`} | $c_{1b} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge \neg$`fungible` $\wedge \neg$`replaced` $c_{1c} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge$ `fungible` $\wedge \neg$`replaced` $c_{1d} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge \neg$`fungible` $\wedge$ `replaced` |
| $co_3 = (${`damaged`, `fungible`, `replaced`}$, -)$ Boundary ={`damaged`, `fungible`, `replaced`} | {`damaged`, `fungible`, `replaced`} | $c_2 : \delta \wedge \neg co_0 \wedge \neg co_1 \wedge \neg co_2 \wedge co_3$ $\wedge$ `damaged` $\wedge$ `fungible` $\wedge$ `replaced` |

The preference ordering: $c_0 > c_{1a} \sim c_{1b} \sim c_{1c} \sim c_{1d} > c_2$

From Example 1, a case model $(C, \geqslant)$ is constructed as in Table 1. Case $c_0$, which is a most preferred case in $C$, is constructed from the default case $co_0$. `fungible` and `replaced` are attached to the case since they are not in the boundary of the default case. $c_{1a}$ is constructed from $co_1$ since it has only one internal sub-boundary. In contrast, $c_{1b}, c_{1c}, c_{1d}$ are constructed from the same case with outcome $co_2$ since it has three internal sub-boundaries. $c_{1a}, c_{1b}, c_{1c}, c_{1d}$ are immediately less preferred than $c_0$ because they are constructed from the cases with outcome that directly attack the default one. Meanwhile, $c_2$ is constructed from $co_3$ and $c_2$ is the least preferred in $C$.

## 5 Explaining Case Models

In this section, we present another contribution of developing dispute trees for explaining case models. To develop the explanation, we first look into the concept of analogy, which is defined as follows [19].

**Definition 13 (Analogy).** *Let $\mathcal{L}$ be a classical logical language, $(C, \geqslant)$ be a case model, and $\sigma \in \mathcal{L}$ be a situation. We say $\alpha \in \mathcal{L}$ expresses an* analogy *of a case $\omega \in C$ and $\sigma$ if $\omega \models \alpha$ and $\sigma \models \alpha$.*

For any case $\omega$ and any situation $\sigma$, we have that $\top$ is the most general analogy of $\omega$ and $\sigma$, and $\omega \vee \sigma$ is the most specific analogy of $\omega$ and $\sigma$ [23]. By extending the concept of specificity from AA-CBR, we introduce a *literal analogy* as an analogy in the form of $\top$ or a conjunction of literals. This makes $\top$ still the most general literal analogy of $\omega$ and $\sigma$, but $\omega \vee \sigma$ is not always the most specific literal analogy due to the logical or. The exception is that sometimes there is a conjunction of literals that is equivalent to $\omega \vee \sigma$, in that case, such a conjunction is the most specific literal analogy.

**Definition 14 (Literal Analogy).** *We say an analogy $\alpha$ is a* literal analogy *of $\omega$ and $\sigma$ if and only if $\alpha$ is $\top$ or a conjunction of literals. and we say a literal analogy $\alpha$ is the* most specific literal analogy *of $\omega$ and $\sigma$ if and only if for every literal analogy $\alpha'$ of $\omega$ and $\sigma$, $\alpha \models \alpha'$.*

By the concept of literal analogy, we introduce a new type of rebuttals called *specificity rebuttal*, based on the attack relations in AA-CBR, also inspired by [13, 17]. It intuitively means the rebuttal consists in finding a more specific literal analogy from a most preferred case with the opposite outcome.

**Definition 15 (Specificity Rebuttal).** *Let $\mathcal{L}$ be a classical logical language, $(C, \geqslant)$ be a case model, $(\varphi, \psi)$ be a presumptively valid argument, and $\sigma \in \mathcal{L}$ be a situation. We say a non-tautologous $\chi \in \mathcal{L}$ (i.e. $\chi \neq \top$) is* specificity rebutting *the argument with respect to $\sigma$ if and only if*

- *$\exists \omega \in C : \omega \models \varphi \wedge \neg\psi$; $\forall \omega' \in C$: if $\omega' \models \varphi \wedge \neg\psi$, then $\omega \geqslant \omega'$ ($\omega$ is a most preferred case in the set of such $\omega'$ with respect to $\geqslant$); and*
- *$\varphi \wedge \chi$ is a most specific literal analogy of $\omega$ and $\sigma$.*

Now, we present dispute trees in case models based on those in AA-CBR as follows.

**Definition 16 (Dispute Tree in Case Models).** *Let $\sigma \in \mathcal{L}$ be a situation, $(C, \geqslant)$ be a case model, and $\psi_0$ be a logic formula such that $(\top, \psi_0)$ is presumptively valid with respect to $(C, \geqslant)$. A* dispute tree *for $\psi_0$ with respect to $(C, \geqslant)$ and $\sigma$ is a tree $\mathcal{T}$ such that:*

1. *every node of $\mathcal{T}$ is of the form $[L : (\varphi, \psi)]$ where $L \in \{P, O\}$ and $\varphi, \psi \in \mathcal{L}$.*
2. *the root of $\mathcal{T}$ is $[P : (\top, \psi_0)]$*

3. *for every* $[P : (\varphi, \psi)]$ *and for every* $\chi \in \mathcal{L}$ *that is specificity rebutting* $(\varphi, \psi)$ *with respect to* $\sigma$*, there exists* $[O : (\varphi \wedge \chi, \neg\psi)]$ *as a child of* $[P : (\varphi, \psi)]$*;*
4. *for every* $[O : (\varphi, \psi)]$*, there exists at most one child* $[O : (\varphi \wedge \chi, \neg\psi)]$ *such that* $\chi$ *is specificity rebutting* $(\varphi, \psi)$ *with respect to* $\sigma$*;*
5. *there are no other nodes in* $\mathcal{T}$ *except those given by 1-4.*

*A dispute tree* $\mathcal{T}$ *is a* maximal *dispute tree if and only if for every* $[O : (\varphi, \psi)]$ *which is a leave in* $\mathcal{T}$*, no* $\chi \in \mathcal{L}$ *that is specificity rebutting* $(\varphi, \psi)$ *with respect to* $\sigma$*.*

We prove a theorem that a maximal dispute tree in the constructed case models is homomorphic to some maximal dispute tree in AA-CBR, i.e. there is a mapping (not always bijective) from nodes in a maximal dispute tree in the constructed case models to nodes in the corresponding maximal dispute tree in AA-CBR such that the parent-child adjacencies are still preserved. Roughly speaking, a maximal dispute tree in the constructed case models can be reduced into a maximal dispute tree in AA-CBR.

**Theorem 1.** *Given a fact situation* $N$*, a case base* $CB$ *with a default case* $(\varnothing, d)$*; the corresponding AA framework* $(AR, attacks)$*; and the case model* $(C, \geqslant)$ *constructed from* $CB$ *with respect to a proposition* $\delta$*. A maximal dispute tree* $\mathcal{T}$ *for* $\delta$ *with respect to* $(C, \geqslant)$ *and* $\bigwedge_{p_i \in N} p_i$ *is homomorphic to some maximal dispute tree* $\mathcal{T}'$ *for* $(\varnothing, d)$ *with respect to* $(CB, attacks)$*, with a homomorphic mapping from a node* $[L : (\varphi, \psi)]$ *in* $\mathcal{T}$ *to a node* $[L : (X, o)]$ *in* $\mathcal{T}'$ *such that a most preferred case in* $\{\omega | \omega \models \varphi \wedge \psi\}$ *is constructed from* $(X, o)$*.*

*Proof.* We prove by induction that $\mathcal{T}$ is homomorphic to some maximal dispute tree $\mathcal{T}'$ for $(\varnothing, d)$ with respect to $(AR, attacks)$ .

- **base case:** The root $[P : (\top, \delta)]$ of $\mathcal{T}$ corresponds to the root $[P : (\varnothing, d)]$ of $\mathcal{T}'$ $(\top, \delta)$ has grounding in a most preferred case in $\{\omega | \omega \models \delta\}$ with respect to $\geqslant$, which is always constructed from $(\varnothing, d)$.
- **inductive step:** If $[O : (\varphi', \neg\psi)]$ is a child of $[P : (\varphi, \psi)]$ in $\mathcal{T}$ and $[P : (\varphi, \psi)]$ corresponds to $[P : (Y, o_y)]$ in $\mathcal{T}'$, then there exists a most preferred case $\omega_x$ in the set $\{\omega | \omega \models \varphi \wedge \neg\psi\}$ with respect to $\geqslant$. Since $\omega_x$ is constructed from some $(X, o_x) \in CB$, we have that $(X, o_x)$ attacks $(Y, o_y)$ because $o_x \neq o_y$ (as $\omega_x \models \neg\psi$); $Y \subsetneq X$ (as there exists a non-tautologous $\chi$ such that $\omega_x \models \varphi \wedge \chi$); and $\nexists (Z, o_x) \in AR$ with $Y \subsetneq Z \subsetneq X$ (as $\omega_x$ is a most preferred case in the set, hence $(X, o_x)$ is far from $(Y, o_y)$ by a distance of *attacks* 1). Therefore, $[O : (X, o_x)]$ is a child of $[P : (Y, o_y)]$ (This can be applied analogously for a case that $[P : (\varphi', \neg\psi)]$ is a child of $[O : (\varphi, \psi)]$).
- If $\mathcal{T}$ is maximal, then for all opponent node $[O : (\varphi, \psi)]$ which are leaves in $\mathcal{T}$, no $\chi$ is specificity rebutting $(\varphi, \psi)$ with respect to $\bigwedge_{p_i \in N} p_i$. Hence, there is no $(X, o_x) \in AR$ that attacks $(Y, o_y)$ if $[O : (Y, o_y)]$ corresponds to $[O : (\varphi, \psi)]$, otherwise there is $\chi = \bigwedge_{p_j \in X \setminus Y} p_j$ that is specificity rebutting $(\varphi, \psi)$ with respect to $\bigwedge_{p_i \in N} p_i$, which leads to the contradiction. Hence, $\mathcal{T}'$ is maximal.

Figure 3 shows examples of maximal dispute trees for $\delta$ with respect to the case model in Table 1 and the situation $\texttt{delayed} \wedge \texttt{damaged} \wedge \texttt{fungible} \wedge \texttt{replaced}$. We have that the dispute trees on the left and the right of the Figure 3 are homomorphic to the dispute trees on the left and the right of Figure 2 respectively.

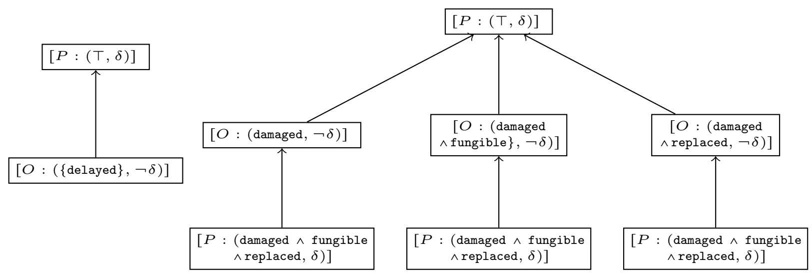


**Fig. 3.** Examples of maximal dispute trees for $\delta$ with respect to the case model constructed from the example

## 6 Discussion

In this paper, we present a method for constructing case models and dispute trees for explaining case models based on AA-CBR. However, unlike dispute trees in AA-CBR [5, 6] that start with a default case, dispute trees in case models can start with any arbitrary formula $\psi$ such that $(\top, \psi)$ is presumptively valid. Although the formula is originally derived from a proposition representing a default outcome, it is not necessary to be such a proposition. Since previous studies [1, 16] show that AA-CBR case bases can be translated into stratified logic programs, it follows immediately from this paper that case models constructed from AA-CBR case bases can also be translated into stratified logic programs. Unfortunately, not every case models can be translated into stratified logic programs because case models can express inconsistencies, which stratified logic programs cannot express. Future research could investigate whether there is a programming paradigm or a logical framework that every case model can be translated into. Interesting candidates are answer set programming or defeasible logic since they can express inconsistencies.

Besides AA-CBR, it is interesting to investigate constructing and explaining case models from other perspectives, such as from precedential constraint. Some differences between dispute trees in case models and dialogue games in precedential constraint [14, 22] are, for example, dispute trees in case models play on hypothetical arguments, i.e. arguments that might not have grounding in real precedent cases, while dialogue games in precedential constraint play on real precedent cases. Another difference is that the dispute trees in case models studied here consider only specificity rebuttals. They do not consider the idea in precedential constraint that a precedent case can defend a decision for a new case with stronger support without using specificity. Therefore, new types of explanations and attack relations in case models might be found if we construct and explain case models from precedential constraint or other perspectives.

## 7 Conclusion

This paper presents a method of constructing case models based on abstract argumentation for case-based reasoning (AA-CBR). The constructed case models consists of cases, each of which is constructed from each internal sub-boundary of each critical case with outcome in the case base, and preferences over cases, which are determined by the distance of attacks between the default case with outcome and the considering case with outcome in the corresponding argumentation framework in AA-CBR. By connecting AA-CBR to case models, we can derive dispute trees with respect to a case model constructed and a situation. It has been shown that the maximal dispute trees in case models can be reduced into maximal dispute trees in AA-CBR. In future work, it would be interesting to study constructing and explaining case models from other perspectives and to study relations between case models and other programming paradigms or logical frameworks.

**Acknowledgements.** This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

## References

1. Athakravi, D., Satoh, K., Law, M., Broda, K., Russo, A.: Automated inference of rules with exception from past legal cases using ASP. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 83–96. Springer International Publishing, Cham (2015)
2. Atkinson, K., Bench-Capon, T., Bollegala, D.: Explanation in AI and law: Past, present and future. Artificial Intelligence **289**, 103387 (2020)
3. Bench-Capon, T.J.: Hypo's legacy: introduction to the virtual special issue. Artificial Intelligence and Law **25**(2), 205–250 (2017)
4. Cocarascu, O., Stylianou, A., Čyras, K., Toni, F.: Data-empowered argumentation for dialectically explainable predictions. In: ECAI 2020, pp. 2449–2456. IOS Press, Amsterdam, The Netherlands (2020)
5. Cyras, K., Satoh, K., Toni, F.: Abstract argumentation for case-based reasoning. In: Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning. pp. 243–254. AAAI Press, CA, USA (2016)
6. Cyras, K., Satoh, K., Toni, F.: Explanation for case-based reasoning via abstract argumentation. In: Computational Models of Argument. pp. 243–254. IOS Press, Amsterdam, The Netherlands (2016)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)

8. Dung, P.M., Kowalski, R.A., Toni, F.: Dialectic proof procedures for assumption-based, admissible argumentation. Artificial Intelligence **170**(2), 114–159 (2006)
9. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence **171**(10-15), 642–674 (2007)
10. Henderson, J., Bench-Capon, T.: Describing the development of case law. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 32–41. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019)
11. van Leeuwen, L., Verheij, B.: A comparison of two hybrid methods for analyzing evidential reasoning. In: Legal Knowledge and Information Systems. pp. 53–62. IOS Press, Amsterdam, The Netherlands (2019)
12. Loui, R.P., Norman, J.: Rationales and argument moves. Artificial Intelligence and Law **3**(3), 159–189 (1995)
13. Prakken, H.: A tool in modelling disagreement in law: preferring the most specific argument. In: Proceedings of the 3rd international conference on Artificial intelligence and law. pp. 165–174. Association for Computing Machinery, New York, NY, USA (1991)
14. Prakken, H., Ratsma, R.: A top-level model of case-based argumentation for explanation: formalisation and experiments. Argument & Computation **13**(2), 159–194 (2022)
15. Prakken, H., Wyner, A., Bench-Capon, T., Atkinson, K.: A formalization of argumentation schemes for legal case-based reasoning in aspic+. Journal of Logic and Computation **25**(5), 1141–1166 (2015)
16. Satoh, K., Kubota, M., Nishigai, Y., Takano, C.: Translating the Japanese Presupposed Ultimate Fact Theory into logic programming. In: Proceedings of the 2009 Conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference. pp. 162–171. IOS Press, Amsterdam, The Netherlands (2009)
17. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial intelligence **53**(2-3), 125–157 (1992)
18. Verheij, B.: Formalizing value-guided argumentation for ethical systems design. Artificial Intelligence and Law **24**(4), 387–407 (2016)
19. Verheij, B.: Formalizing arguments, rules and cases. In: Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law. p. 199–208. ICAIL '17, Association for Computing Machinery, New York, NY, USA (2017)
20. Verheij, B.: Proof with and without probabilities: Correct evidential reasoning with presumptive arguments, coherent hypotheses and degrees of uncertainty. Artificial Intelligence and Law **25**, 127–154 (2017)
21. Walton, D.N.: Argumentation schemes for presumptive reasoning. Routledge, New York (1996)
22. van Woerkom, W., Grossi, D., Prakken, H., Verheij, B., Čyras, K., Kampik, T., Cocarascu, O., Rago, A., et al.: Justification in case-based reasoning. In: Proceedings of the First International Workshop on Argumentation for eXplainable AI. pp. 1–13. CEUR Workshop Proceedings, Utrecht University, The Netherlands (2022)
23. Zheng, H., Grossi, D., Verheij, B.: Logical comparison of cases. In: AI Approaches to the Complexity of Legal Systems XI-XII, pp. 125–140. Springer, Cham (2020)