Abstract

The concept of context was first discussed in AI by McCarthy, and ever since it has found many applications in knowledge representation and defeasible reasoning.

Argument-based systems have evolved into powerful formalisms for performing defeasible reasoning. In this respect, the MTDR framework (Simari & Loui, 1992) has come to be most successful. However, dialectical considerations imposed a rather procedural approach, which causes the protocol for making inferences to be tightly interlocked with the system’s underlying logic. Protocols are an important issue in argument-based frameworks, since they determine the performance and behavior of the system.

This paper presents an alternative formalization for MTDR, in which protocol issues may be kept apart from the logic for defeasible argumentation itself. The ability to perform inference will be captured using debate contexts.
1. Introduction

An Argumentative System [8, 6, 9] is a formalization of the process of defeasible reasoning. An argument $A$ for a hypothesis $h$ is a tentative piece of reasoning an agent would be inclined to accept, all things considered, as an explanation for $h$. In the presence of new incoming information, the argument $A$ may lose support or become weakened, and therefore the hypothesis $h$ may no longer be regarded as valid. In that manner nonmonotonicity arises.

Many aspects of argumentative systems have been recently the focus of research, producing some interesting results. In *A Mathematical Treatment of Defeasible Reasoning* [8], or *MTDR*, a clear and theoretically sound structure for an argument-based reasoning system was introduced. In [6], the original framework was conceptually improved, and a refined, dialectics-based approach was introduced. However, the formalization of this evolved, refined system has a strong procedural flavor, since dialectics particularly involves the way the process of debate is performed.

As a result, the protocol used for making inferences comes to be tightly interlocked with the underlying logic for defeasible argumentation. The importance of protocols in argumentative reasoners, as well as the need to formalize the features of those protocols using an expressive notation, has been stressed by Loui in [3]. As he points out, considerations about protocols and search seem to have been underestimated in AI, by considering them mostly just “a detail of implementation”. Nevertheless, the recent development of argumentative systems and its successful applicability in defeasible reasoning have drawn the attention of the AI community to this issue.

This paper presents an alternative formalization for defining an Argument System, in which protocol issues may be kept apart from the logic for defeasible argumentation itself. The ability to perform inference through argumentation will be captured using contexts, a formalization tool which has been recently studied, proving to be useful for knowledge representation and defeasible reasoning [2].

2. Contexts

The concept of context was first discussed in AI by McCarthy [4], and in several informal drafts. Contexts have found a large number of uses in AI. Guha’s Ph.D. Thesis [2] was the first in-depth study of contexts. The AI community has also accepted the need to consider contexts when transferring information from an agent to another. Logical properties of contexts, such as soundness and completeness results, have been also established [1]. Next we will present some basic notions about contexts (for a complete treatment of this subject, see [2]).
Contexts are “rich objects” in the domain of a theory. Contextual effects on an expression may be so rich that they cannot be captured in the scope of a logic. This leads us to include contexts as objects in our ontology. We will extend a classic first-order language to include wffs of the form ist(C,F), where F is a wff of a first-order language, and C is the name of the context, and ist(C,F) stands for ‘F is true in context C’. The context symbol C is supposed to capture any assumption that is not in F, but is required to make F a meaningful statement. The semantics of the FOL is also extended for interpreting the sentence ist(C,F) as a wff of the language. Logical connectives such as ∧, ∨, and ¬ preserve their traditional meaning. The idea is that F might depend on some contextual aspects that have not (yet) been specified, and these aspects are to be captured by the context argument. It might not be possible ever to list completely all of these context dependencies. At any time, we might have only a partial description of the context and this is why contexts are assumed to be rich objects. As Guha [2] says, “the context object can be thought of as the reification of the context dependencies of the sentences associated with the context”.

In section 4, we will see how to embed the MTDR framework for defeasible argumentation into a context-based system.

3. Dialectics and protocols

In this section, we will first briefly summarize the main features of the MTDR framework. We will also introduce the motivations and guidelines for an alternative formalization using contexts. A complete definition of the MTDR framework can be found elsewhere [8].

3.1. Dialectics in defeasible argumentation. According to the definitions presented in the appendix, an argument A is accepted as a defeasible reason for a conclusion h if A is a justification for h. To decide the acceptability of an argument A for a literal h, its associated counterarguments B₁, B₂, ⋯ Bₖ will be obtained (def. 6.3) each Bᵢ being a (defeasible) reason for rejecting A. If some Bᵢ is supported on “better” (or unrelated) evidence than A, then Bᵢ will be a candidate for defeating A (def. 6.4). Since counterarguments are also arguments, the former analysis should be in turn carried out on them. Thus, Bᵢ will defeat A unless there exists an argument Cᵢ (which corresponds to one of the counterarguments C₁,C₂,⋯Cᵦ associated to Bᵢ) that defeats Bᵢ. In that case, we will be forced to reject Bᵢ, and hence our original argument A would be reinstated.

From the recursive procedure mentioned above a tree structure, called dialectical tree (def. 6.5), can be generated. Paths in a dialectical tree are called argumentation lines (def. 6.6), which can be thought of alternate sequences of supporting and interfering arguments in a debate (def. 6.7). Argumentation lines can be shown to be finite [6].
However, the basic definition of argumentation line allows cycles as the dialectical tree is being built, and this leads to fallacious argumentation (see [6]). This imposes some constraints on the construction of argumentation lines, taking us to the notion of acceptable argumentation line (def. 6.8) and acceptable dialectical tree (def. 6.9), which are free of fallacies.

Finally, the definition of undefeated nodes and defeated nodes (def. 6.10) in a dialectical tree suggests a bottom-up labeling procedure, through which we are able to determine if the root turns out to be labeled as defeated or undefeated. If the root \( \langle A, h \rangle \) comes to be undefeated, \( A \) will be a justification for \( h \) (def. 6.11).

3.2. The need of defining protocols. The procedure just described closely resembles a debate between two parties, i.e., it is a dialectical process. According to the concepts presented in [6], a dialectical tree appears to be built in a depth-first fashion: supporting and interfering arguments are advanced alternatively, and argumentation lines expand as a new counterargument for defeating a leaf node is introduced. This was indeed the way justifications were originally obtained in the MTDR framework [8]. However, we must be aware that a dialectical tree does not necessarily grow up from its leaves, since the labeling process is performed dynamically, and some defeated arguments may be reinstated, becoming potential candidates for counterargumentation. Consider the following example:

**Example 3.1** Given the argumentation line \( \lambda = [\langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \langle A_3, h_3 \rangle] \) as a dialectical tree is being built, there are two points for counterarguing: \( \langle A_3, h_3 \rangle \) (leaf) and \( \langle A_1, h_1 \rangle \) (reinstated supporting argument).

Thus, we should have the choice of either expanding \( \lambda \) by introducing a counterargument for \( \langle A_3, h_3 \rangle \), or creating a new argumentation line by introducing another counterargument for \( \langle A_1, h_1 \rangle \). The interaction between two parties in a dialectical process as the one which takes place in the MTDR framework must be performed according to some protocol. As Loui points out in [3], protocols constitute an important issue within the process of debate. They allow us to formalize and study the behavior of different argumentative systems, comparing them with other alternative formalisms.

The MTDR formalism provides a definition of what an acceptable dialectical tree should be. We want to be able to jump to a higher abstraction level, allowing the definition of protocols which specify how that dialectical tree can be constructed. Contexts will help us to clump together related assertions concerning different stages in the process of obtaining a justification. They will allow us to focus the process of debate, and to define local consistency constraints which do not affect the whole knowledge base (for example, the need of preserving consistency for a given set of arguments, or the avoidance of circular argumentation lines).
4. Defeasible argumentation using contexts

We will present an alternative formalization for the MTDR framework, in which the process of building a dialectical tree will be implicit. First, we will introduce a number of elements using a suitable, dialectics-based notation. These elements are based on the original concepts presented in [5], which were further extended in [3].

The construction of a dialectical tree resembles a debate, in which two parties interact: a proponent (prop) and an opponent (opp) (see [5, 3] for more details). Let Party denote either of the two parties, and let Party denote the adversary of Party. The proponent’s task is defending a given claim, either by advancing supporting arguments or by defeating previous arguments presented by the opponent. The opponent’s task is to outweigh the evidence presented by the proponent, advancing arguments which interfere the supporting arguments advanced so far.

Arguments in a debate may be either dead or alive, abbreviated as “∀” and “∃”, respectively. As new arguments are introduced in a debate, they are per se alive. If an argument \( \langle A_2, h_2 \rangle \) defeats another argument \( \langle A_1, h_1 \rangle \), this causes \( \langle A_1, h_1 \rangle \) to be dead. An argument \( \langle A_3, h_3 \rangle \) defeating \( \langle A_2, h_2 \rangle \) causes \( \langle A_1, h_1 \rangle \) to be reinstated, turning it alive again.

Supporting and interfering arguments constitute two different entities from a pragmatic point of view, each of them deserving a similar logical treatment, but ‘opposed’ in meaning as the debate is carried out. Similarities will be captured using a context structure for storing the arguments presented by each party.

**Definition 4.1 (Supporting (Interfering) Context)** We will denote as \( C_{\text{prop}} \) (\( C_{\text{opp}} \)) the context associated with the arguments presented by the proponent (opponent) in a debate. These two contexts represent every argument involved in the process of debate. The context \( C_{\text{Party}} \) will be characterized by assertions of the form \( \text{ist}(C_{\text{Party}}, \langle A_1, h_1 \rangle) \), which stands for “Party claims \( \langle A_1, h_1 \rangle \)”.

A debate constitutes a confrontation of arguments and counterarguments. This involves different debate stages, which will be captured through debate contexts. Every time a new counterargument is introduced (by any of the two parties), a new debate context will be entered in. Debate contexts will be denoted as \( C_0, C_1, C_2, \ldots, C_k \). The debate context \( C_0 \) will be the first debate context.

**Definition 4.2 (Debate Context)** We will denote as \( C_i, \) (\( i = 0,1,2,\ldots,n \)) the context associated with a given stage reached in the process of debate. This context will be called

---

3 This notation resembles the one introduced by Rescher in [5], and substitutes the terms defeated and undefeated used in [6]. That terminology was confusing at times, since some arguments are called “defeaters”, and expressions as “undefeated defeaters” could arise.
the debate context. If \( C_{\text{current}} \) denotes the current debate context, then a new debate context \( C_{\text{succ}(\text{current})} \) will be entered in if some party introduces a new argument into the debate which defeats some argument in \( C_{\text{current}} \).

Every debate context \( C_i \) will be characterized by an assertion \( \text{ist}(C_i, \text{defeats}(A, B)) \), standing for “Argument \( A \) defeats argument \( B \) in context \( C_i \)”, and assertions of the form \( \text{ist}(C_i, \text{state}(\text{Arg}, \text{State})) \), which stand for “Argument \( \text{Arg} \) is in condition \( \text{State} \) in context \( C_i \)”. \( \text{State} \) may be “!” or “!”.

The outermost debate context represents the current stage in the process of debate. It should include all relevant information for further argumentation. Parties will perform moves from the information stored in the current debate context. As expected, a debate will be formally defined as a sequence of moves involving two parties.

**Definition 4.3 (Moves)** Parties (proponent and opponent) will have the possibility of performing two kinds of moves for advancing arguments.

- **Assertion**: Party can just assert an argument \( \langle A, h \rangle \), incorporating it into its set \( C_{\text{Party}} \) of arguments.
- **Attack**: Party can attack \( \overline{\text{Party}} \) by advancing an argument from \( C_{\text{Party}} \) which defeats one of the \( \overline{\text{Party}} \)’s arguments present in the current debate context \( C_{\text{current}} \).

**Definition 4.4 (Debate)** A debate is a finite sequence \( \{m_1, m_2, m_3, \ldots, m_n\} \), where each \( m_i \) is a pair with the form \( (\text{Party}, \text{Move}) \).

As a debate is being carried out, new arguments defeat old ones. Every time a new defeater (attack) is introduced, a new debate context will be entered in. However, information from previous debate contexts can be lifted into this new context, in order to keep the outcome of the debate updated. This will be performed using lifting rules.

Next we will state inference rules for defining the process of performing defeasible argumentation using contexts. Some of their premises can only be derived from the basic definitions of argument, counterargument and defeat. Such premises are enclosed in brackets ([ ]).

1) Argument construction: \[
\frac{[K \cup A \not\vdash h]}{\langle A, h \rangle}
\]

2) Assertion: \[
\frac{\text{ist}(C_{\text{Party}}, \langle A, h \rangle) \land \text{ist}(C_0, \text{state}(A, !))}{\text{ist}(C_{\text{Party}}, \langle A, h \rangle)}
\]
3) Attack: \[ \text{ist}(C_{\text{Party}}(A_1, h_1)) \quad \text{ist}(C_{\text{Party}}(A_2, h_2)) \]
\[ [(A_1, h_1) \gtrless \text{def}(A_2, h_2)] \quad \text{ist}(C_i, \text{state}(A_2, !)) \]
\[ \text{ist}(C_{\text{succ}(i)}, \text{defeats}(A_1, A_2)) \]

4) Defeat: \[ \text{ist}(C_i, \text{defeats}(A_1, A_2)) \]
\[ \text{ist}(C_i, \text{state}(A_1, !)) \land \text{ist}(C_i, \text{state}(A_2, !)) \]

5) Link: \[ \text{ist}(C_i, \text{defeats}(A_2, A_1)) \quad \text{ist}(C_j, \text{defeats}(A_3, A_2)) \]
\[ i < j \]
\[ \text{ist}(C_j, \text{links}(A_3, A_1)) \]

6) Transitivity among links: \[ \text{ist}(C_i, \text{links}(A_2, A_3)) \quad \text{ist}(C_j, \text{links}(A_2, A_1)) \]
\[ i < j \]
\[ \text{ist}(C_j, \text{links}(A_3, A_1)) \]

7) Reinstatement (argum. line): \[ \text{ist}(C_i, \text{defeats}(A_1, A_2)) \quad \text{ist}(C_i, \text{links}(A_1, B)) \]
\[ \text{ist}(C_i, \text{state}(B, !)) \]

8) Reinstatement: \[ \text{ist}(C_i, \text{state}(A_1, !)) \quad \text{notist}(C_i, \text{defeats}(A_2, A_1)) \]
\[ \text{ist}(C_{\text{succ}(i)}, \text{state}(A_1, !)) \]

9) Non-circularity \[ \text{ist}(C_i, \text{links}(A, A)) \]
\[ \perp \]

10) Consistency \[ \text{ist}(C_i, \text{links}(A, B)) \quad \text{ist}(C_{\text{Party}}, A) \quad \text{ist}(C_{\text{Party}^\prime}, B) \]
\[ \perp \]

Rule 1 states how to construct an argument, according to the def. 6.2. As stated in def. 4.3, parties can make two kinds of moves: assertions (rule 2) and attacks (rule 3). In order to assert an argument, a party must just be able to construct it from \((K, \Delta)\). An attack can be performed only on arguments asserted by the other party being currently alive, and it involves the introduction of a new debate context. Rule 4 establishes the consequence of a defeat relation between two arguments (the defeater is alive, and the defeated argument is dead).

Next we will analyze how to lift information into the current debate context. We need to introduce a new relation between arguments, which will allow us to infer which arguments within the current argumentation line can be reinstated.

**Definition 4.5** Let \(A_1, h_1\), \(A_2, h_2\) and \(A_3, h_3\) be arguments, such that \(A_1, h_1\) defeats \(A_2, h_2\), and \(A_2, h_2\) defeats \(A_3, h_3\). Then we say that there is a link between \(A_1, h_1\) and \(A_3, h_3\).

The definition of link is expressed in rule 5. Links among arguments are transitive (rule 6). It can be shown that linked arguments within the current argumentation line preserve
their associated condition (i.e., the last argument introduced into the debate is alive, and so are those arguments linked to it). This has been stated in rule 7. An argument belonging to any other argumentation line may be transferred from debate context $C_i$ into debate context $C_{\text{succ}(i)}$ as long as no ‘defeats’ assertion affects it (rule 8). These last two rules may be used for reinstating arguments which belong to previous debate contexts. Those arguments lifted into the current context will be potential candidates for further counterargumentation.

The ability to conceptualize reinstatements within the current argumentation line in terms of links turns out to be most helpful, since it allows us to impose the necessary constraints on the process of building an acceptable dialectical tree. Circular reasoning can be avoided by demanding the link relation not to be reflexive, i.e., arguments should not be linked to themselves (rule 9). Contradictory argumentation, in turn, can be avoided by not allowing links among arguments which belong to both parties (rule 10). These two rules should be applied as constraints, and may be thought of as premises for the rule 3 (attack). If some move results in deriving $\bot$, the possibility of performing that move is denied.

Performing moves in a debate can be understood as the application of the inference rules given above. Given a debate $d = \{m_1, m_2, \ldots, m_k\}$, we will say that a sentence $s$ is an outcome of the debate $d$ if $s$ can be consistently derived from $(\mathcal{K}, \Delta)$ by application of the inference rules. Each assertion (Party asserts $(A, h_i)$) constitutes an application of rule 2, and each attack (Party asserts $(A_1, h_1)$ defeating $(A_2, h_2)$) constitutes an application of rule 3. Any other inference rule whose premises are satisfied may be also applied.

We are specially interested in those debates in which no further moves are possible. We will call them exhaustive debates.

**Definition 4.6** A debate $d = \{m_1, m_2, \ldots, m_k\}$ is exhaustive if no more moves are possible after move $m_k$ is performed.

Figure 1 shows an example of an exhaustive debate. No more moves are possible (possible arguments were exhausted), and the original argument $(A_1, h_1)$ turns out to be alive. This prompts an alternative definition of justification set in this framework:

**Definition 4.7** An argument $(A, h)$ is a justification for $h$ iff there exists an exhaustive debate $d$ in which $\text{ist}(C_{\text{current}}, \text{state}(A, !))$ is one of its outcomes.

The final labeling of the root node in a dialectical tree does not depend from the way the tree was obtained, since the number of arguments involved is finite, and in the long run the “better” (most specific) arguments will prevail. This can be expressed as follows:

**Proposition 4.1** Let $(A, h)$ be an argument based on a given $(\mathcal{K}, \Delta)$. If $A$ is a justification, then every exhaustive debate will have $\text{ist}(C_{\text{current}}, \text{state}(A, !))$ as one of its outcomes. \(\square\)

---

4In rule 8, $\text{notist}(C, s)$ means ‘$\text{ist}(C, s)$ does not hold’.

5This can be inferred from the inductive definition of justification given in [8].
Given a knowledge base \((K, \Delta)\) from which the arguments \(\langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \langle A_3, h_3 \rangle, \langle A_4, h_4 \rangle, \) and \(\langle A_5, h_5 \rangle\) may be constructed, such that \(\langle A_2, h_2 \rangle \defeq \langle A_1, h_1 \rangle, \langle A_3, h_3 \rangle \defeq \langle A_1, h_1 \rangle, \langle A_4, h_4 \rangle \defeq \langle A_2, h_2 \rangle, \langle A_5, h_5 \rangle \defeq \langle A_3, h_3 \rangle, \langle A_1, h_1 \rangle \defeq \langle A_4, h_4 \rangle.\) The following exhaustive debate may be carried out:

<table>
<thead>
<tr>
<th>Move</th>
<th>Results in</th>
<th>By rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. \textit{Prop} claims (\langle A_1, h_1 \rangle)</td>
<td>((C_0, \text{state}(A_1, !)), \text{ist}(C_{prop}, \langle A_1, h_1 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td>2. \textit{Opp} claims (\langle A_2, h_2 \rangle), defeating (\langle A_1, h_1 \rangle).</td>
<td>((C_1, \text{state}(A_2, !)), \text{ist}(C_{opp}, \langle A_2, h_2 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>((C_1, \text{defeats}(A_2, A_1)))</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>((C_1, \text{state}(A_1, !)), \text{ist}(C_{state}, \langle A_1, h_1 \rangle))</td>
<td>4</td>
</tr>
<tr>
<td>3. \textit{Prop} claims (\langle A_4, h_4 \rangle), defeating (\langle A_2, h_2 \rangle).</td>
<td>((C_2, \text{state}(A_4, !)), \text{ist}(C_{prop}, \langle A_4, h_4 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>((C_2, \text{defeats}(A_4, A_2)))</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>((C_2, \text{state}(A_4, !)), \text{ist}(C_{state}, \langle A_4, h_4 \rangle))</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{links}(A_4, A_1)))</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>((C_2, \text{state}(A_1, !)))</td>
<td>7</td>
</tr>
<tr>
<td>4. \textit{Opp} fails to claim (\langle A_1, h_1 \rangle).</td>
<td>((C_0, \text{state}(A_1, !)), \text{ist}(C_{opp}, \langle A_1, h_1 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{defeats}(A_1, A_4)))</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{links}(A_1, A_2)))</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>(\bot)</td>
<td>10</td>
</tr>
<tr>
<td>5. \textit{Opp} claims (\langle A_3, h_3 \rangle), defeating (\langle A_1, h_1 \rangle).</td>
<td>((C_0, \text{state}(A_3, !)), \text{ist}(C_{opp}, \langle A_3, h_3 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{defeats}(A_3, A_1)))</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{state}(A_3, !)), \text{ist}(C_{state}, \langle A_3, h_3 \rangle))</td>
<td>4</td>
</tr>
<tr>
<td>6. \textit{Prop} claims (\langle A_5, h_5 \rangle), defeating (\langle A_3, h_3 \rangle).</td>
<td>((C_0, \text{state}(A_5, !)), \text{ist}(C_{prop}, \langle A_5, h_5 \rangle))</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>((C_4, \text{defeats}(A_5, A_3)))</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>((C_3, \text{links}(A_5, A_1)))</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>((C_4, \text{state}(A_1, !)))</td>
<td>7</td>
</tr>
</tbody>
</table>

Since no more moves are possible, and \(\text{state}(A_1, !)\) belongs to the current debate context, \(\langle A_1, h_1 \rangle\) is a justification.

Figure 1: An example of an exhaustive debate

5. Conclusions

The \textit{MTDR} framework provided us with a powerful formalism for performing defeasible argumentation. Dialectical considerations, though useful for detecting and correcting flawed reasoning patterns, imposed a rather procedural approach to the original proposal presented in [8]. We contend that the formalization presented in this paper offers an alternative, non-procedural approach to defeasible argumentation within the \textit{MTDR} framework. Its most relevant features are:
• The ability to strengthen or weaken constraints on the process of debate, by changing or modifying the set of inference rules. Rule 2, for example, could be rewritten so that arguments in $C_{Party}$ should be consistent among themselves.

• Different protocols may be defined using the same inference rules, by demanding closing with respect to some particular rule as the debate proceeds. If we want leaves in the dialectical tree to have preeminence over inner nodes when looking for our next counterargumentation point, we will have a sort of depth-first debate. If, on the contrary, inner nodes prevail over leaves, a breadth-first debate will result. These two situations can be captured by preferring rule 7 (reinstatement in the current argumentation line) to rule 8 (generic reinstatement).

We think that a context-based framework as the one presented in this paper provides an useful formalization tool for stating definitions of different protocols for defeasible argumentation, allowing us to analyze their behavior and performance. Having a formalism for defining and studying protocols constitutes an important step in the quest for an optimal protocol which allow us to determine which is the shortest exhaustive debate for a given argument $\langle A, h \rangle$. This issue, however, deserves further treatment, and is being worked on at the time.

6. Appendix: the MTDR framework

We will briefly introduce the main concepts and definitions of the MTDR framework (see [8, 6] for details). The knowledge of an intelligent agent $A$ will be represented using a first-order language $\mathcal{L}$, plus a binary meta-linguistic relation “$\triangleright$” between sets of non-ground literals of $\mathcal{L}$ which share variables. The members of this meta-linguistic relation will be called defeasible rules, and they have the form “$\alpha \triangleright \beta$”. The relation “$\triangleright$” is understood as expressing that “reasons to believe in the antecedent $\alpha$ provide reasons to believe in the consequent $\beta$.”

Let $K$ be a consistent subset of sentences of $\mathcal{L}$. This set can be partitioned in two subsets $K_G$, of general (necessary) knowledge, and $K_P$, of particular (contingent) knowledge. The beliefs of $A$ are represented by a pair $(K, \Delta)$. $K$ represents the non-defeasible part of $A$’s knowledge. $\Delta$ is a finite set of defeasible rules, and represents information that $A$ is prepared to take at less than face value. $\Delta^r$ denotes the set of all ground instances of members of $\Delta$.

**Definition 6.1** Let $\Gamma$ be a subset of $K \cup \Delta^r$. A ground literal $h$ is a defeasible consequence of $\Gamma$, abbreviated $\Gamma \triangleright h$, if and only if there exists a finite sequence $B_1, \ldots, B_n$ such that $B_n = h$ and for $1 \leq i < n$, either $B_i \in \Gamma$, or $B_i$ is a direct consequence of the preceding elements in the sequence by virtue of the application of any inference rule of the first-order theory associated with the language $\mathcal{L}$. Ground instances of the defeasible rules are regarded as material implications for the application of inference rules. We will write $K \cup A \triangleright h$ distinguishing the set $A$ of defeasible rules used in the derivation from the set $K$. 
Definition 6.2 Given a set $\mathcal{K}$, a set $\Delta$ of defeasible rules, and a ground literal $h$ in the language $\mathcal{L}$, we say that a subset $A$ of $\Delta$ is an argument structure (or just argument) for $h$ in the context $\mathcal{K}$ (denoted by $\langle A, h \rangle_{\mathcal{K}}$, or just $\langle A, h \rangle$) if and only if: 1) $\mathcal{K} \cup A \not\models h$, 2) $\mathcal{K} \cup A \not\models \bot$ and 3) $\exists A' \subset A, \mathcal{K} \cup A' \models h$. A subargument of $\langle A, h \rangle$ is an argument $\langle S, j \rangle$ such that $S \subseteq A$.

Definition 6.3 Given two arguments $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$, we say that $\langle A_1, h_1 \rangle$ counterargues $\langle A_2, h_2 \rangle$, denoted $\langle A_1, h_1 \rangle \overset{h}{\Rightarrow} \langle A_2, h_2 \rangle$ iff
1. There exists a subargument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ such that $\mathcal{K} \cup \{h_1, h_2\} \not\models \bot$.
2. For every proper subargument $\langle S, j \rangle$ of $\langle A_1, h_1 \rangle$, it is not the case that $\langle A_2, h_2 \rangle \rightarrow \langle S, j \rangle$.

Definition 6.4 Given two argument structures $\langle A_1, h_1 \rangle$ and $\langle A_2, h_2 \rangle$, we say that $\langle A_1, h_1 \rangle$ defeats $\langle A_2, h_2 \rangle$ at literal $h$, denoted $\langle A_1, h_1 \rangle \overset{\mathrm{def}}{\Rightarrow} \langle A_2, h_2 \rangle$, if and only if there exists a subargument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ such that: $\langle A_1, h_1 \rangle$ counterargues $\langle A_2, h_2 \rangle$ at the literal $h$ and
1. $\langle A_1, h_1 \rangle$ is strictly more specific\(^6\) than $\langle A, h \rangle$, or 2. $\langle A_1, h_1 \rangle$ is unrelated by specificity to $\langle A, h \rangle$.
If $\langle A_1, h_1 \rangle \overset{\mathrm{def}}{\Rightarrow} \langle A_2, h_2 \rangle$, we will also say that $\langle A_1, h_1 \rangle$ is a defeater for $\langle A_2, h_2 \rangle$.

Definition 6.5 A dialectical tree $T_{\langle A, h \rangle}$ for an argument $\langle A, h \rangle$ is recursively defined as follows:

1. A single node containing an argument structure $\langle A, h \rangle$ with no defeaters is by itself a dialectical tree for $\langle A, h \rangle$. This node is also the root of the tree.
2. Suppose that $\langle A, h \rangle$ is an argument structure with defeaters $\langle A_1, h_1 \rangle$, $\langle A_2, h_2 \rangle$, $\ldots$, $\langle A_n, h_n \rangle$.
   We construct the dialectical tree $T_{\langle A, h \rangle}$, by putting $\langle A, h \rangle$ as the root node of $T_{\langle A, h \rangle}$ and by making this node the parent node of the roots of the dialectical trees for $\langle A_1, h_1 \rangle$, $\langle A_2, h_2 \rangle$, $\ldots$, $\langle A_n, h_n \rangle$.

Definition 6.6 Let $\langle A_0, h_0 \rangle$ be an argument structure. Then every path $\lambda$ in $T_{\langle A_0, h_0 \rangle}$ from the root $\langle A_0, h_0 \rangle$ to a leaf $\langle A_n, h_n \rangle$, denoted $\lambda = [\langle A_0, h_0 \rangle, \langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \ldots, \langle A_n, h_n \rangle]$, constitutes an argumentation line for $\langle A_0, h_0 \rangle$.

Definition 6.7 Let $T_{\langle A_0, h_0 \rangle}$ be a dialectical tree, and let $\lambda = [\langle A_0, h_0 \rangle, \langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \ldots, \langle A_n, h_n \rangle]$ be an argumentation line for $\langle A_0, h_0 \rangle$. Then every $\langle A_i, h_i \rangle$ in $\lambda$ can be labeled as a supporting or interfering argument as follows
1. $\langle A_0, h_0 \rangle$ is a supporting argument in $\lambda$, and
2. If $\langle A_i, h_i \rangle$ is a supporting (interfering) argument in $\lambda$, then $\langle A_{i+1}, h_{i+1} \rangle$ is an interfering (supporting) argument in $\lambda$.
We will denote as $S_{\lambda}$ and $I_{\lambda}$ the set of all supporting and interfering arguments in $\lambda$, respectively.

Definition 6.8 Let $\lambda = [\langle A_0, h_0 \rangle, \langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \ldots, \langle A_n, h_n \rangle]$ be an argumentation line. Then, $\lambda$ will be called an acceptable argumentation line iff

\(^6\) We use specificity as comparison criterion, but any other partial order among arguments would be valid.
1. Supporting (interfering) arguments in $\lambda$ are concordant pairwise, i.e., $\kappa \cup A_i \cup A_j \neq \bot$, for every $\langle A_i, h_i \rangle, \langle A_j, h_j \rangle \in S^\lambda(I^\lambda)$.

2. Let $\langle A_i, h_i \rangle$ be an argument structure in $S^\lambda(I^\lambda)$. There is no argument $\langle A_j, h_j \rangle$ in $I^\lambda(S^\lambda)$, such that $i < j$ and $\langle A_i, h_i \rangle$ defeats $\langle A_j, h_j \rangle$.

**Definition 6.9** Let $\langle A, h \rangle$ be an argument structure. An acceptable dialectical tree for $\langle A, h \rangle$, denoted $T_{\langle A, h \rangle}$, is recursively defined as follows:
1. A single node containing an argument structure $\langle A, h \rangle$ with no defeaters is by itself an acceptable dialectical tree for $\langle A, h \rangle$. This node is also the root of the tree.
2. Suppose that $\langle A, h \rangle$ is an argument structure with defeaters $\langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \ldots, \langle A_n, h_n \rangle$.
   We construct the dialectical tree for $\langle A, h \rangle$, $T_{\langle A, h \rangle}$, by putting $\langle A, h \rangle$ in the root node of it and by making this node the parent node of the roots of the acceptable dialectical trees of $\langle A_1, h_1 \rangle, \langle A_2, h_2 \rangle, \ldots, \langle A_n, h_n \rangle$, i.e., $T_{\langle A_1, h_1 \rangle}, T_{\langle A_2, h_2 \rangle}, \ldots, T_{\langle A_n, h_n \rangle}$.

**Definition 6.10** Let $T_{\langle A, h \rangle}$ be a dialectical tree for an argument structure $\langle A, h \rangle$. The nodes of $T_{\langle A, h \rangle}$ can be recursively labeled as undefeated nodes (U-nodes) and defeated nodes (D-nodes) as follows:
1. Leaves of $T_{\langle A, h \rangle}$ are U-nodes.
2. Let $\langle B, q \rangle$ be an inner node of $T_{\langle A, h \rangle}$. Then $\langle B, q \rangle$ will be an U-node iff every child of $\langle B, q \rangle$ is a D-node. $\langle B, q \rangle$ will be a D-node iff it has at least an U-node as a child.

**Definition 6.11** Let $\langle A, h \rangle$ be an argument structure, and let $T_{\langle A, h \rangle}$ be an acceptable dialectical tree. We will say that $A$ is a justification for $h$ iff the root node of $T_{\langle A, h \rangle}$ is an U-node.

7. References


