On the use of contexts for representing knowledge in defeasible argumentation

Carlos Iván Chesñevar[1]

Instituto de Ciencias e Ingeniería de Computación (ICIC)
Grupo de Investigación en Inteligencia Artificial (GIIA)
Departamento de Ciencias de la Computación
Universidad Nacional del Sur
Av.Alem 1253 – (8000) Bahía Blanca – REPÚBLICA ARGENTINA
FAX: (54) (91) 883933 – PHONE: (54) (91) 20776 (ext.208) – Email: `ccchesne@criba.edu.ar`

KEYWORDS: artificial intelligence, defeasible reasoning, argumentative systems

Abstract

The notion of *context* and its importance in knowledge representation and non-monotonic reasoning was first discussed in Artificial Intelligence by John McCarthy. Ever since, contexts have found many applications in developing knowledge-based reasoning systems.

Defeasible argumentation has gained wide acceptance within the AI community in the last years. Different argument-based frameworks have been proposed. In this respect, *MTDR* (Simari & Loui, 1992) has come to be one of the most successful. However, even though the formalism is theoretically sound, there exist some dialectical considerations involving argument construction and the inference mechanism, which impose a rather procedural approach, tightly interlocked with the system's logic.

This paper discusses different uses of contexts for modelling the process of defeasible argumentation. We present an alternative view of *MTDR* using contexts. Our approach will allow us to discuss novel issues in *MTDR*, such as defining a set of moves and introducting an *arbiter* for regulating inference. As a result, protocols for argument generation as well as some technical considerations for speeding up inference will be kept apart from the logical machinery underlying *MTDR*.

[1]Supported by a fellowship of the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), República Argentina.

On the use of contexts for representing knowledge in defeasible argumentation

1 Introduction and motivations

An argumentative system [11, 13] is a formalization of the process of defeasible reasoning. Arguments are tentative pieces of reasoning an intelligent agent would be inclined to accept, all things considered, as an explanation for a given hypothesis. Given an argument A for a hypothesis h, the presence of *counterarguments* for A causes it to become weakened, and A may no longer be regarded as acceptable. However A may be *reinstated* if further counter-counterarguments appear. The final acceptance of the argument A as a *justified* reason for explaining h will result from a recursive procedure, in which arguments, counterarguments, counter-counterarguments, and so on, must be taken into account.

Argument-based systems have drawn the attention of the AI community during the last years. In that respect, many alternative formalizations have been proposed. In *A Mathematical Treatment of Defeasible Reasoning* [11], or *MTDR*, a clear and theoretically sound structure for an argument-based reasoning system was introduced. On this basis, some interesing results have been obtained, which include the development of argument-based expert systems and an extension of the original framework for incorporating TMS-facilities through an *arguments base* [3]. A conceptually improved version of the original framework based on dialectical considerations has been presented in [10].

The resulting framework for defeasible reasoning involves two aspects: on the one hand, defining *what* is supposed to be a *proof* (namely, the notion of justification). On the other hand, stating the way in which proofs are obtained through a dialectical procedure, which comes to be interlocked with the logical machinery underlying *MTDR*.

The notion of *context* and its application in knowledge representation and defeasible reasoning was first discussed by McCarthy in informal drafts. In R.V.Guha's Ph.D.Thesis [5] (under McCarthy's direction), this issue was studied in depth, and relevant contributions for actual AI applications were presented. Guha's main motivation was indeed the development of the Cyc system, a task which seemed to be intractable without contexts as a formalization tool. Logical properties of contexts have been recently established, and they have turned into a powerful tool for improving existing formalisms for nonmonotonic reasoning. McCarthy has discussed some issues about contexts from a philosophical point of view [8]. The notion of context itself seems to be rather ellusive. In his own words "...it seems to me unlikely that this study will result in a unique conclusion about *what a context is*. Instead, as usual in AI, various notions will be found useful."

This paper discusses some uses of contexts as a formalization tool in defeasible argumentation. We will follow McCarthy's suggestion, using contexts from a particular viewpoint which comes out to be useful for our purposes. We will restate some old definitions from *MTDR*, and introduce new ones. Some of the basic elements for context-based argumentation were originally introduced in [12]. In this paper, some of those definitions have been refined, and some new concepts are incorporated.

We are not pursuing a context-based system equivalent in expressiveness to *MTDR*. Instead, we want to integrate both declarative and procedural issues into a unifying framework, so that dialectical procedures and *MTDR*'s underlying logic can be kept apart. In order to do so, we will use contexts.

2 Contexts as formalization tools in AI

The concept of context was first discussed in AI by McCarthy, and in several informal drafts. The first in-depth study of contexts was carried out by Guha [5]. Ever since, the AI community has also accepted the need to consider contexts when transferring information among intelligent agents. Logical properties of contexts, such as soundness and completeness results, have been also established. Next we will discuss some basic notions about contexts (for a complete treatment of this subject, see [5]). 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.

In order to incorporate contexts into our framework, we will extend a classic first-order language L to include wffs of the form ist(C,F), where F is a wff of L, 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 L is also extended for interpreting the sentence ist(C,F) as a wff of the language. Logical connectives such as $\wedge$, $\vee$, and $\neg$ 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. In that respect, Guha [5] says that "the context object can be thought of as the reification of the context dependencies of the sentences associated with the context". Some of the most relevant features of contexts are:

- A context-based system is assumed to be always *in* some context. It is possible to change the context the system is in, by *entering* a new context. The inverse action is *exiting*.
- A context-based framework also includes *lifting rules*, which relate different contexts. Lifting rules can be thought of as having the form $holds(C_i, s) \longrightarrow holds(C_j, s)$, meaning "if sentence s is known to hold in context C_i, it will also hold in context C_j."
- There is always an *outermost* context associated with a system. This outermost context may be *trascended*, and a new outermost context could be introduced. This might be seen as equivalent to introducing one meta-level higher than was previously available. We will further consider this notion in section 4.

Next, we will discuss how to embed the *MTDR* framework for defeasible argumentation into a context-based system.

3 Context-based defeasible argumentation

3.1 *MTDR*: a brief overview

We will rephrase the main ideas behind our argumentative framework (the reader is referred to [11] for further details, and to the appendix for basic definitions).

Firstly, we say that an argument A will be *accepted* as a defeasible reason for a conclusion h if A is a *justification* for h. The acceptance of the original argument A as a justification for h will result from a recursive procedure, in which arguments, counterarguments, counter-counterarguments, and so on, should be taken into account. This leads to a tree structure, called *dialectical tree.*[2] Paths along that tree will be called *argumentation lines*, which can be thought of alternate sequences of *supporting* and *interfering* arguments in a debate. However,it may be the case that an argumentation line contains contradictory supporting (interfering) arguments, or that it does not convey any progress in the process of finding a justification. This results in cycles along an argumentation line, which result in *fallacious argumentation* [3], and therefore should be avoided. This leads us to the notion of *acceptable argumentation line.* As expected, only dialectical trees built from acceptable argumentation lines will be accepted as valid. An argument $\langle A, h\rangle$ will be considered to be a *justification* for h iff there exists an acceptable dialectical tree for $\langle A, h\rangle$.

3.2 *MTDR* within contexts[3]

According to the definitions presented in [12], the construction of a dialectical tree can be seen as a debate [4]between two parties: a *proponent* (*prop*) and an *opponent* (*opp*). The proponent must defend a given claim, either by advancing supporting arguments or by defeating previous interfering arguments. The opponent must outweigh the evidence presented by the proponent, advancing arguments which *interfere* the supporting arguments advanced so far. *Party* will denote either of the two parties, and $\overline{Party}$ will denote the adversary of *Party*.

We will associate the debate process with two kind of contexts. On the one hand, there will be a context for both proponent and opponent (denoted C_{prop} and C_{opp}, respectively). The context C_{Party} will be related to assertions of the form ist(C_{Party}, argument($\langle A_1, h_1\rangle$)), standing for "*Party* supports $\langle A_1, h_1\rangle$". On the other hand, we will capture the sucessive stages in the process of debate through so-called *debate contexts*, which are basically 'snapshots' of the arguments in conflict advanced so far. New debate contexts will be entered in whenever some party introduces an argument in a debate that defeats another. Debate contexts will be numbered C_0, C_1, ... , C_n. Every debate context will be related to an assertion ist(C_i,defeats(A,B)) (standing for "Argument A defeats argument B in context C_i") and assertions of the form ist(C_i,state(Arg,$State$)), which stands for "Argument Arg is in condition $State$ in context C_i". $State$ may be "!" (stands for *alive* or undefeated) or "†" (stands for *dead* or defeated), respectively.

[2]See [10] for a formal definition.
[3]See [12] for a more complete discussion of the basic definitions presented here.
[4]See [9, 10] for details.

3.2.1 Moves

The outermost debate context can be seen as the current stage in the process of debate, so that it should be associated with all relevant information for further argumentation. The *moves* performed by both parties in order to carry out the debate will be based on the information in the current debate context, as well as on arguments in C_{prop} and C_{opp}. There are several kinds of moves that could be defined. In [12], a basic set of two moves (assertion and attack) was introduced. We will now extend that set to four moves. The motivations for doing so are discussed below.

- **Assertion**: *Party* can just *assert* an argument $\langle A, h\rangle$, incorporating it into its set C_{Party} of arguments.
- **Attack**: *Party* can *attack* $\overline{Party}$ by advancing an argument from C_{Party} which *defeats* one of the $\overline{Party}$'s arguments present in the current debate context $C_{current}$.
- **Retraction**: *Party* can retract an argument advanced in a previous assert or attack move.
- **Concession**: *Party* may concede $\overline{Party}$ the acceptance of an argument advanced in a previous assert or attack move.

The first two moves allow parties to carry on a debate; however, this seems insufficient. In order to get a more natural model of dialectical interaction, parties should be allowed to retract arguments which seem deemed to be defeated or considered irrelevant for the debate. We must keep in mind that parties are not allowed to introduce neither contradictory nor circular arguments as the debate proceeds.[5] Performing a wrong move m should not force a party to support m for the rest of the debate. Retracting m seems therefore to be a reasonable solution.

On the other hand, it might be the case that some party finds out that one of the arguments advanced by the other party cannot be defeated for some reason (because of its strength, lack of computational resources, etc.). In that case, one party should be able to concede the other the acceptance of the argument at issue. That would simplify further moves in the debate, since parties would no longer disagree about the conclusion supported by that argument. This has some interesting implications, which may considerably affect the size of the search space (see section 4 for further details).

3.2.2 Some context-based inference rules

Next we will briefly describe a set of *inference rules* for modeling the process of performing defeasible argumentation within *MTDR* using the moves mentioned above.[6]

[5] See [10] for the formal definition of contradictory and circular argumentation.

[6] See [12] for details.

Rule 1: $$\frac{[\mathcal{K} \cup A \mathrel{|\!\!\sim} h] \quad [\mathcal{K} \cup A \not\mathrel{|\!\!\sim} \perp] \quad [\nexists A' \subset A,\ \mathcal{K} \cup A' \mathrel{|\!\!\sim} h]}{\langle A, h\rangle}$$

Rule 2: $$\frac{\langle A, h\rangle}{\mathsf{ist}(C_{Party},\ \mathsf{argument}(\langle A,h\rangle)) \wedge \mathsf{ist}(C_0,\ \mathsf{state}(\langle A,h\rangle,!))}$$

Rule 3: $$\frac{\mathsf{ist}(C_{Party},\ \mathsf{argument}(\langle A_1,h_1\rangle)) \quad \mathsf{ist}(C_{\overline{Party}},\langle A_2,h_2\rangle) \quad [\langle A_1,h_1\rangle \gg_{def} \langle A_2,h_2\rangle] \quad \mathsf{ist}(C_i,\ \mathsf{state}(\langle A_2,h_2\rangle,!))}{\mathsf{ist}(C_{succ(i)},\ \mathsf{defeats}(\langle A_1,h_1\rangle,\langle A_2,h_2\rangle))}$$

Rule 4: $$\frac{\mathsf{ist}(C_{Party},\ \mathsf{argument}(\langle A_1,h_1\rangle))}{\mathsf{ist}(C_{\overline{Party}},\ \mathsf{argument}(\langle A_1,h_1\rangle))}$$

Rule 5: $$\frac{\mathsf{ist}(C_i,\ \mathsf{defeats}(\langle A_1,h_1\rangle,\langle A_2,h_2\rangle))}{\mathsf{ist}(C_i,\ \mathsf{state}(\langle A_1,h_1\rangle,!)) \wedge \mathsf{ist}(C_i,\ \mathsf{state}(\langle A_2,h_2\rangle,\dagger))}$$

Rule 6: $$\frac{\mathsf{ist}(C_i,\ \mathsf{state}(\langle A_1,h_1\rangle,!)) \quad \mathsf{notist}(C_i,\ \mathsf{defeats}(\langle A_2,h_2\rangle,\langle A_1,h_1\rangle))}{\mathsf{ist}(C_{succ(i)},\ \mathsf{state}(\langle A_1,h_1\rangle,!))}$$

Through rule 1 we state when it is valid to construct a given argument.[7] Rules 2 and 3 define valid moves, namely assertions (rule 2) and attacks (rule 3). In order to assert an argument, a party must just be able to construct it from $(\mathcal{K}, \Delta)$. An attack can be performed only on those arguments currently alive asserted by the other party, and it involves the introduction of a new debate context. Rule 4 stands for the concession move; either of the two parties can incorporate some of the other party's argument into its associated context. Rule 5 establishes the consequence of a defeat relation between two arguments (the defeater is alive, and the defeated argument is dead). Rule 6 states how to *lift* arguments alive in previous debate contexts into the current debate context. The sentence **notist(C,s)** stands for "**ist(C,s)** does not hold".

Retracting an argument $\langle A_1, h_1\rangle$ basically involves eliminating the argument from C_{Party}, and introducing a new debate context, which properly reflects the current situation of the arguments involved in the debate. That new debate context could be the result of applying some kind of belief revision operator on the current debate context.[8]

As a debate is carried out, some arguments turn out to be reinstated, and can be lifted into the current debate context. These arguments can be identified through the *link* relation. (This process has been described in detail in [12]). Links are transitive, and linked arguments preserve their "aliveness" within a given argumentation line. Linked arguments can be thus reinstated into the current debate context. There were two situations to be avoided because of dialectical considerations, namely circular reasoning and contradictory argumentation [10]. Links allow us to specify rules that behave as constraints for the inference process, leading to inconsistency whenever the moves are not performed within these constraints.

[7]Premises enclosed in brackets mean that they can only be derived from the basic definitions of argument, counterargument and defeat.

[8]The definition of that operator is currently being studied. The retract move is thus presented rather informally, as a natural extension for an argument-based framework, as stated in [4, 1]. It must be remarked that retracting arguments is not present in *MTDR*.

3.2.3 Exhaustive debates

A *debate* is basically a finite sequence $\{m_1, m_2, m_3, \ldots, m_n\}$, where each m_i is a pair with the form $(Party, Move)$. If the first move in a debate d is asserting an argument $\langle A, h\rangle$, we will say that d is a debate *about* h.

A debate can be understood as sucessive applications of the inference rules given above. Let $(\mathcal{K}, \Delta)$ be a knowledge base. Let $d = \{m_1, m_2, ..., m_k\}$ be a debate about a claim h, based on $(\mathcal{K}, \Delta)$. We will say that a sentence s is *outcome* of the debate d iff s can be consistently derived from $(\mathcal{K}, \Delta)$ by application of the inference rules associated with $m_1, m_2, ..., m_k$. Since we are mainly interested in knowing whether the claim h is supported or not, we will restrict ourselves to the current debate context.

DEFINITION **3.1** Let C_i be the current debate context when performing a debate about a given claim h. Then the set of all sentences associated with C_i will be called the *current outcome* of the debate.

We are specially interested in those debates in which no further moves can be performed which affect the current outcome. This kind of debates will be called *exhaustive debates.* The notion of justification will be then characterized in terms of exhaustive debates.

DEFINITION **3.2** Let $d = \{m_1, m_2, ..., m_k\}$ be a debate about a claim h. We will say that d is *exhaustive* iff it is no longer possible to perform applications of inference rules changing the current outcome.

It must be remarked that the final labeling of the root node in a dialectical tree does not depend from *the way* the tree was obtained,[9] since the number of arguments involved is finite, and in the long run the "better" (most specific) arguments will prevail. This prompts an alternative definition of justification set in this framework:

DEFINITION **3.3** An argument $\langle A, h\rangle$ is a *justification* in for h if there exists an exhaustive debate d about the claim h in which $\mathsf{ist}(C_{current},\mathsf{state}(\langle A, h\rangle,!))$ is one of its outcomes.

4 Metalevel facilities provided by contexts

Metalevel reasoning seems to be a natural issue in defeasible argumentation, particularly in conection with frameworks for legal argumentation. A debate between proponent and opponent may be arbitered by a third party (the *arbiter*),[10] which analizes the confrontation among arguments 'from the outside', and solves different metalevel problems. Among those tasks to be performed by the arbiter, we can mention the following:

- Establishing whether the debate protocol is being correctly followed;
- Deciding current course of action on the basis of previous debates;

[9] This can be inferred from the inductive definition of justification given in [11].
[10] Called *determiner* or *judge* in other approaches.

- Determining whether the debate should be ended because of some particular reason (resources exhausted, time-out settings, etc.).

In this respect, it is interesting to recall an important feature of contexts, namely the notion of *trascendence* [8]:

> Human intelligence involves an ability that no-one has yet undertaken to put in computer programs, namely the ability to *trascend* the context of one's beliefs. [...] In fact, this ability is required for something less than full intelligence. We need it to be able to comprehend someone else's discovery even if we can't make the discovery ourselves.

McCarthy wants to exploit the notion of transcendence based on the fact that there always exists an implicit *outer* context. From the fact that a sentence p is true, for example, our intelligent agent should be able to infer $\mathsf{ist}(c_0,p)$, where c0 stands for the agent's outer context. Actually, this operation could be performed several times, resulting in new outer contexts c_{-1}, c_{-2}, and so on. Finally, McCarthy points out that the usefulness of trascendence will depend on having a suitable collection of rules for *lifting* sentences to the higher level contexts.

4.1 A third party in a debate: the arbiter

In order to define the role of the arbiter in a debate, we can also use context-based inference rules, involving sentences such as $\mathsf{ist}(C_{arbiter}, \mathsf{ist}(C,s))$, where C stands for a context (such as C_{Party}), and s stands for a particular sentence. Consider for example formalizing the fact that "the arbiter should know every argument presented by the proponent".[11] This can be easily done through the rule

$$\frac{\mathsf{ist}(C_{prop},\ \mathsf{argument}(\langle A_1, h_1\rangle))}{\mathsf{ist}(C_{arbiter},\ \mathsf{argument}(\langle A_1, h_1\rangle))}$$

The arbiter may even possess knowledge which trascends the knowledge of the parties involved. In this respect, it is interesting to consider the notion of *commitment store.* We know both proponent and opponent share a common knowledge base, *i.e.*, they are commited to accept certain beliefs (e.g. those derived from $\mathcal{K}$). However, dialectical considerations also impose a set of "shared" knowledge as the debate is carried out. That set will be called *commitment store.*

Consider the following situation: the proponent introduces a given argument $\langle A_1, h_1\rangle$ in a debate, the opponent counterargues by introducing a new argument $\langle A_2, h_2\rangle$. However, both arguments may have some subarguments in common (*i.e.*, the opponents *concedes* some inner literals in $\langle A_1, h_1\rangle$, but still rejects its conclusion). Up to that point, subarguments in common between $\langle A_1, h_1\rangle$ and $\langle A_2, h_2\rangle$ are no longer argueable within that argumentation line. We can capture this concept by introducing a new context, called C_{commit}, for representing information both parties are commited to as the debate proceeds. The information in C_{commit} would result from the application of the following inference rule:

[11] Of course, the same rule should apply also for opponent.

$$\frac{\text{ist}(C_{current}, \text{state}(\langle A_1, h_1\rangle, !)) \quad \text{ist}(C_{current}, \text{state}(\langle A_2, h_2\rangle, !)) \quad C = A_1 \cap A_2 \quad \text{ist}(C_{Party}, \text{argument}(\langle A_1, h_1\rangle)) \quad \text{ist}(C_{\overline{Party}}, \text{argument}(\langle A_2, h_2\rangle))}{\text{ist}(C_{commit}, \text{argument}(\langle C, h\rangle))}$$

The information in C is basically a set of grounded defeasible rules, whose conclusions does not necessarily belong to $\mathcal{K}$. However, we know that those literals should not be further counterargumentation points, for the reasons mentioned above. The arbiter should then consider the inner literals supported by those "arguments in common" under the same epistemic status as the one associated with ground facts derivable from $\mathcal{K}$. Thus, the arbiter could hold his own set of beliefs within his associated context according to the following rules:

$$\frac{\text{ist}(C_{commit}, \text{argument}(\langle A, h\rangle))}{\text{ist}(C_{arbiter}, \text{argument}(\langle A, h\rangle))} \qquad \frac{[\mathcal{K} \vdash s]}{\text{ist}(C_{arbiter}, s)}$$

The commitment store may be best enlarged by conceding arguments. Whenever some party concedes an argument $\langle A_1, h_1\rangle$ to the other party, all ground defeasible rules in $\langle A_1, h_1\rangle$ should be incorporated into the commitment store, since both parties agree on every inner literal in $\langle A_1, h_1\rangle$.

It must be observed that the previous analysis leads to important computational savings when considering implementation issues. Computing arguments is particularly expensive in *MTDR*, since it involves performing exhaustive search on the knowledge base. The structural resemblance of argument and counterargument would help simplify the specificity checking, by eliminating the analysis on literals supported by both argument and counterargument. Exploiting inference rule as those stated above could result in a sorted dialectical tree, in which the shortest argumentation lines are built as soon as possible (see [2]).

4.2 Defining debate protocols

Defining a proper *debate protocol* is a particularly important issue in defeasible argumentation. A protocol establishes how the parties engaged in the debate are going to interact. Some aspects of protocols involve defining when an argument defeats another, when termination is appropiate, etc. As R. Loui observes [6], protocols have been underestimated in nonmonotonic reasoning because resource-bounds "are not taken seriously", and considered to be associated with search strategies and implementation features. Loui contends that formal study of protocol is needed for nonmonotonic reasoning. He states that "policies without processes have no meaning"and "for policies, protocol is semantics".

A context-based framework as the one presented in this paper provides interesting features for defining protocols on the basis of the set of inference rules. We can easily specify particular constraints on the process of debate, by changing or modifying the set of inference rules. The most interesting feature is perhaps the possibility of defining protocols by demanding closure with respect to some particular inference rule. Thus, for example, every party might be enforced to introduce all possible arguments for supporting a given claim before attacking some of the arguments introduced by the other party. We must recall that a debate is a tree-like structure. By demanding closure with respect to attacks, we would be working on a depth-first basis; by demanding closure with respect to constructing all possible arguments, we would be working on a breadth-first basis.

Where is protocol to be defined? We consider this to be an open issue. It seems natural that the arbiter be the one who enforces the application of certain rules. Thus, protocols could be defined within the arbiter's context. However, the role of the arbiter itself might be part of the protocol, so that a new context trascending the arbiter's context, could be introduced. The context-based framework presented in this paper is properly suited for being further extended and considering metalevel facilities as the protocol definition, arbiter's role, etc.

5 Conclusions and future work

We contend that the formalization presented in this paper offers an alternative approach to defeasible argumentation using *MTDR*. We have presented a unifying framework for defining the different elements used in performing defeasible argumentation. On the one hand, we introduced several context names for capturing the information associated with different situations. Then, we presented a set of context-based inference rules for defining the process of inference in our framework. These rules basically relate the information associated with different contexts.

The use of contexts provided an useful formalization tool for distinguishing declarative from procedural features within *MTDR*. As a result, different protocols for defeasible argumentation can be defined, depending on the set of inference rules defined, as well as the preference criteria for deciding the order under which the rules are to be evaluated.

The set of moves presented in this paper is related to other argument-based approaches, such as Gordon's [4] and Brewka's [1], and extends considerably the *MTDR* framework. The work of Brewka is particularly significant, since it constitutes the first 'direct step forward in nearly two decades' [7] from the non-monotonic reasoning community into dialectics. Brewka considers disputation as the process of constructing a default theory the proponent and opponent agree upon. We think that interesting connections can be drawn between performing a debate within *MTDR* and building default theories as described above. Brewka himself finds useful to explore the relationships among game theoretic models of argumentation and nonmonotonic reasoning, when aiming at applications involving multiple agents.

There are still many issues to be solved in defeasible argumentation, and a lot of questions remain open. Having an adequate formalism for defining protocols is one of them [6]. Even though we know that there is still much work ahead, we hope that the results discussed in this paper be a useful contribution in that direction.

A The MTDR framework

In this appendix, we will briefly describe the main concepts of the *MTDR* framework. For a complete description, see [11].

A.1 Knowledge representation

The knowledge of an intelligent agent $\mathcal{A}$ will be represented using a first-order language $\mathcal{L}$, plus a binary meta-linguistic relation "$\succ\!\!-$", defined on $\mathcal{L}$, between a set of nonground literals (antecedent) and a nonground literal (consequent) which share variables. The members of this meta-linguistic relation will be called *defeasible rules*. The relation "$\alpha \succ\!\!- \beta$" is understood as expressing that "reasons to believe in the antecedent α provide reasons to believe in the consequent β". We will restrict the first-order language $\mathcal{L}$ to a subset involving only Horn clauses.

The set $\mathcal{K}$ will be a consistent subset of $\mathcal{L}$ representing the non-defeasible part of $\mathcal{A}$'s knowledge. Δ is a finite set of nonground defeasible rules representing information that $\mathcal{A}$ is prepared to take at less than face value. If $A \subseteq \Delta$, we will denote as $A^{\downarrow}$ the set of all ground instances of members of A.

The set $\mathcal{K}$ can be partitioned into two subsets: $\mathcal{K}_G$ (*general* knowledge) and $\mathcal{K}_P$ (*particular* or *contingent* knowledge). Sentences in $\mathcal{K}_P$ will be ground literals (*E.g.*: $flies(\text{tweety})$, $penguin(\text{opus})$) which do not appear as consequents of rules in $\mathcal{K}_G$ or Δ, since they represent basic input information sensored by $\mathcal{A}$, from which new information can be inferred. Sentences in $\mathcal{K}_G$ will be material implications having the form $a_1, a_2, \ldots, a_k \rightarrow b$, *e.g.* $penguin(X) \rightarrow bird(X)$. Defeasible rules have the form $a_1, a_2, \ldots, a_k \succ\!\!- b$, *e.g.* $bird(X) \succ\!\!- flies(X)$.

A.2 Inference

In order to make this paper self-contained, we present next definitions A.1 through A.8, which summarize the notion of inference in *MTDR* (for a complete definition of this framework, see [11]).

DEFINITION **A.1** Let Γ be a subset of $\mathcal{K} \cup \Delta^{\downarrow}$. A ground literal h is a *defeasible consequence* of Γ, abbreviated $\Gamma \mid\!\sim h$, 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 $\mathcal{K} \cup A \mid\!\sim h$ distinguishing the set A of defeasible rules used in the derivation from the set $\mathcal{K}$.

DEFINITION **A.2** Given a set $\mathcal{K}$, a set Δ of defeasible rules, and a ground literal h in the language $\mathcal{L}$, we say that a subset A of $\Delta^{\downarrow}$ 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) $\mathcal{K} \cup A \mid\!\sim h$, 2) $\mathcal{K} \cup A \not\mid\!\sim \perp$ and 3) $\not\exists A' \subset A$, $\mathcal{K} \cup A' \mid\!\sim h$. A *subargument* of $\langle A, h\rangle$ is an argument $\langle S, j\rangle$ such that $S \subseteq A$.

DEFINITION **A.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 $\langle A_1, h_1\rangle \overset{h}{\otimes\!\!\rightarrow} \langle A_2, h_2\rangle$ iff there exists a subargument $\langle A, h\rangle$ of $\langle A_2, h_2\rangle$ such that $\mathcal{K} \cup \{h_1, h_2\} \vdash \perp$. The literal h will be called a *counterargumentation literal.*

DEFINITION **A.4** Let $\mathcal{D} = \{a \in \mathcal{L} : a \text{ is a ground literal and } \mathcal{K} \cup \Delta^{\downarrow} \mid\!\sim a\}$, and let $\langle A_1, h_1\rangle$ and $\langle A_2, h_2\rangle$ be two argument structures. We say that A_1 for h_1 is *strictly more specific than* A_2 for h_2, denoted $\langle A_1, h_1\rangle \succ_{\text{spec}} \langle A_2, h_2\rangle$, if and only if

i) $\forall S \subseteq \mathcal{D}$ if $\mathcal{K}_G \cup S \cup A_1 \mid\!\sim h_1$ and $\mathcal{K}_G \cup S \not\mid\!\sim h_1$, then $\mathcal{K}_G \cup S \cup A_2 \mid\!\sim h_2$.

ii) $\exists S \subseteq \mathcal{D}$ such that $\mathcal{K}_G \cup S \cup A_2 \mid\!\sim h_2$, $\mathcal{K}_G \cup S \not\mid\!\sim h_2$ and $\mathcal{K}_G \cup S \cup A_1 \not\mid\!\sim h_1$.

DEFINITION **A.5** 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 $\langle A_1, h_1\rangle \gg_{\text{def}} \langle A_2, h_2\rangle$, 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*[12] than $\langle A, h\rangle$, or 2. $\langle A_1, h_1\rangle$ is unrelated by specificity to $\langle A, h\rangle$.

If $\langle A_1, h_1\rangle \gg_{\text{def}} \langle A_2, h_2\rangle$, we will also say that $\langle A_1, h_1\rangle$ is a *defeater* for $\langle A_2, h_2\rangle$. In case (1) $\langle A_1, h_1\rangle$ will be called a *proper* defeater, and in case (2) a *blocking* defeater.

[12]We use specificity as comparison criterion, but any other partial order among arguments might be possible.

DEFINITION **A.6** A *dialectical tree* $\mathcal{T}_{\langle A, h\rangle}$ 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 $\langle A_1, h_1\rangle$, $\langle A_2, h_2\rangle$, ..., $\langle A_n, h_n\rangle$. We construct the dialectical tree $\mathcal{T}_{\langle A, h\rangle}$, by putting $\langle A, h\rangle$ as the root node of $\mathcal{T}_{\langle A, h\rangle}$ and by making this node the parent node of the roots of the dialectical trees for $\langle A_1, h_1\rangle$, $\langle A_2, h_2\rangle$, ..., $\langle A_n, h_n\rangle$.

DEFINITION **A.7** Let $\mathcal{T}_{\langle A, h\rangle}$ be a dialectical tree for an argument structure $\langle A, h\rangle$. The nodes of $\mathcal{T}_{\langle A, h\rangle}$ can be recursively labeled as *undefeated nodes* (U-nodes) and *defeated nodes* (D-nodes) as follows:

1. Leaves of $\mathcal{T}_{\langle A, h\rangle}$ are *U-nodes.*
2. Let $\langle B, q\rangle$ be an inner node of $\mathcal{T}_{\langle A, h\rangle}$. 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 **A.8** Let $\langle A, h\rangle$ be an argument structure, and let $\mathcal{T}_{\langle A, h\rangle}$ be a dialectical tree. [13] We will say that A is a *justification* for h (or simply $\langle A, h\rangle$ is a *justification*) iff the root node of $\mathcal{T}_{\langle A, h\rangle}$ is an U-node.

References

[1] G. Brewka. A Reconstruction of Rescher's Theory of Formal Disputation Based on Default Logic. In *Workshop on Computational Dialectics*, Seattle, Washington, 1994.

[2] C. I. Chesñevar. *Stratifying defeasible rules in argumentative reasoning: a consistency-based approach.* Technical Report, Grupo de Investigación en Inteligencia Artificial, Univ. Nac. del Sur, Apr. 1994.

[3] A. J. García, C. I. Chesñevar, and G. R. Simari. Making Argument Systems Computationally Attractive. In *Anales de la XIII Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación*, Universidad de La Serena, La Serena (Chile), 1993.

[4] T. F. Gordon. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice.* PhD thesis, Fachbereich Informatik, Technische Hochschule Darmstadt (Alemania), June 1993.

[5] R. V. Guha. *Contexts: A Formalization and some applications.* PhD thesis, Stanford University, Departament of Computer Science, 1991.

[6] R. P. Loui. Process and Policy: Resource-Bounded Non-Demostrative Reasoning. Technical Report WUCS-92-43, Washington University, October 1992.

[7] R. P. Loui. Report on The Workshop on Computational Dialectics. Seattle, Washington, 1994.

[8] J. McCarthy. Notes on Formalizing Contexts. In *Proceedings of the 13th. International Joint Conference in Artificial Intelligence (IJCAI), Chambéry, Francia*, 1993.

[9] N. Rescher. *Dialectics, a Controversy-Oriented Approach to the Theory of Knowledge.* State University of New York Press, Albany, USA, 1977.

[10] G. R. Simari, C. I. Chesñevar, and A. J. García. The role of dialectics in defeasible argumentation. In *Anales de la XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación*, Universidad de Concepción, Concepción (Chile), Nov. 1994.

[11] G. R. Simari and R. P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.

[12] G. R. Simari and C. I. Chesñevar. Arguments and contexts. En *XXI Conferencia Latinoamericana de Informática*, PANEL 95, Canela (Brasil), agosto 1995.

[13] G. A. Vreeswijk. *Studies in Defeasible Argumentation.* PhD thesis, Vrije University, Amsterdam (Holanda), 1993.

[13] Actually, dialectical trees should satisfy certain additional requirements for being considered *acceptable* dialectical trees (see [10]). These requirements involve avoiding cycles and inconsistency within dialectical trees. That issue, however, deserves a detailed analysis which exceeds the scope of this paper.