

Towards an Extensible Argumentation System

Adam Zachary Wyner and Trevor Bench-Capon

Department of Computer Science
Ashton Building
University of Liverpool
Liverpool, L69 3BX, United Kingdom

Abstract

Many types of inter-agent dialogue, including information seeking, negotiation and deliberation can be fruitfully seen as varieties of argumentation. Argumentation frameworks, originally introduced by Dung, provide a powerful tool for evaluating the sets of conflicting arguments which emerge from such dialogues. Originally argumentation frameworks considered arguments as completely abstract entities related by a single attack relation, which always succeeded. Use of the frameworks in practical application such as law, e-democracy and medicine, however, motivated a distinction between successful and unsuccessful attacks, determined by properties of the conflicting arguments. This remains insufficient, however, to capture a range of phenomena which can arise from procedural and contextual considerations, and which require that the success of an attack depend not only on the properties of the conflicting arguments but also on the nature of the attack or the context in which it is made. In this paper we present, in functional decomposition style, an analysis of arguments, their properties and relations which can accommodate a wide range of such phenomena. Our analysis is extensible and is presented in a series of stages which capture first the abstract notions of original argumentation frameworks, is then extended to embrace properties of arguments, and then further extended to include properties of relations between arguments. We illustrate each stage of this progression by representing characteristic systems of each type, and discuss the particular features of argumentation which they can address.

Introduction

In recent theoretical research on argumentation frameworks ((Dung 1995), (Bench-Capon 2003), (Atkinson 2005), among others), arguments are first class objects. Arguments are in an attack relation, where one argument attacks another argument; the attack relation succeeds unless the attacking argument is itself defeated. Arguments can be understood as *nodes* and the attack relation as *arcs*. In some proposals, the success of an attack is regulated by ascribing *properties* to arguments which *filter* the attack relation relative to the property of the arguments; for instance, preferences or values can

be ascribed to the arguments ((Amgoud & Cayrol 1998) and (Bench-Capon 2003)). However, these approaches are not adequate to account for a range of uses of argumentation. We consider several examples. For instance, distinct authorities may determine the success of an attack. Alternatively, the legal force of a precedent may vary with respect to level of the legal context in which the precedent was decided and the level of the court in which the new case is under consideration. Furthermore, arguments may be attacked in a range of ways, for instance, attacking the rule or asserting that an exception holds. In these uses, arguments must be differentiated with respect to properties they bear as well as different sorts of attacks. To account for these other uses, we present a functional decomposition analysis of arguments wherein argument objects bear a range of additional properties, and we can differentiate sorts of attacks. The important novelty of the paper is that it extends the formal approach based on (Dung 1995) with additional properties of arguments and labelled arcs and that it brings formal argumentation to bear on a broader range of phenomena.

In the following, we present our analysis as a series of *extensions* to the abstract Argumentation Framework of ((Dung 1995) and the value-based Argumentation Framework of (Bench-Capon 2003)). In other words, we maintain arguments as nodes and attacks as arcs as well as related notions of *admissible* sets of arguments, but then add properties of arguments as well as *labels* on the arcs. Notions such as admissibility are modified accordingly.

The structure of the paper is as follows. We recast (Dung 1995) and (Bench-Capon 2003) in a notationally equivalent form that facilitates further modifications. Then, we extend the formalism to account for how different authorities may distinguish the outcome of attacks. We show how we can integrate the *Carneades* Argumentation Framework (Gordon & Walton 2006). We sketch an analysis of precedent in court hierarchies (Wyner & Bench-Capon 2007). In the section on future work, we suggest extensions to account for *Burden of Proof* (Prakken & Sartor 2006) as well as *Critical Questions* (Atkinson 2005), and Case-based Reasoning.

An Extensible Argumentation System

An *Extensible Argumentation System* (ExArs) is a language comprised of argument objects, relations, and definitions of auxiliary concepts. We provide only those components of

the system as needed to make our point, then extending it to account for some other phenomena. We use subscripting to differentiate the systems. Our assumption is that (Dung 1995) is the most abstract system, which we extend. However, for clarity, we subscript all the elements.

We assume `boolean` and `object` types. The `boolean` type has two subsorts – `true` and `false`. Initially, we have but one sort of *object* type, namely, arguments. Expressions of the form `object` \rightarrow `boolean` are to be understood as functions from objects to truth-values; that is, in this instance, it expresses the *characteristic function* such that an expression of that type denotes a *set of objects*. By the same token, `(object` \times `object)` \rightarrow `boolean` denotes a set of ordered pairs of objects, which is a relation. The most basic system only has a set of arguments, which are the nodes, and relations between arguments, which are the arcs.

Definition – ExArS₀ is a tuple $\langle \text{Arg}_0, \text{Rel}_0 \rangle$, where:

- Arg_0 is a set of *argument names*, a_1, \dots, a_n , of type `argument` which denote arguments.
- Rel_0 is an *attack* relation between arguments, where we read $\text{Rel}_0(a_1, a_2)$ as argument a_1 attacks argument a_2 . Rel_0 is of type `(argument` \times `argument)` \rightarrow `boolean`.

We give a sample of the main auxiliary definitions of (Dung 1995), which are notational variants of the original definitions. Suppose R, S are subsets of Arg_0 .

Definition – Acceptability, Admissibility, and Extensions

- $x \in \text{Arg}_0$ is *acceptable with respect to* S if for $\forall y \in \text{Arg}_0$ where $\text{Rel}_0(y, x)$, $\exists z \in S$ where $\text{Rel}_0(z, y)$.
- S is *conflict-free* if $\neg \exists y \exists x \in S, x \neq y$ and $\text{Rel}_0(x, y)$.
- A conflict-free set S is *admissible* if $\forall x \in S, x$ is acceptable with respect to S .
- S is a *preferred extension* if it is a *maximal* (with respect to \subseteq) admissible set.

Definitions for *stable extension*, *coherence*, *credulously accepted*, and *skeptically accepted* follow suit.

First Extension – Value-based Argumentation

The Value-based argumentation framework of (Bench-Capon 2003) builds on (Dung 1995). The principle intuition is that an argument attack may succeed or fail *relative to a value that is ascribed to an argument*; that is, intuitively, if there is an argument that I should go eat pastry, and it is attacked by an argument that I should diet, I might still accept that I should go eat pastry because I value eating pastry more than dieting. Here we provide it as our first example of an extension to **ExArS₀**; it is a notational variant of (Bench-Capon 2003). We only provide some of the key clauses.

Definition – ExArS₁

- Arg_1 , a set of arguments.
- Rel_1 , an attack relation between arguments.

- Val is set of *value names*, v_1, \dots, v_n , of type `value`, which denote values. We assume Val_1 is defined for **ExArS₁**.
- *RankingScheme* is a total ordering of Val ; it is understood as an *audience*. For x_i and $y_j \in \text{Val}$, if $\langle x_i, y_j \rangle \in \text{RankingScheme}$, we say that x_i is preferred to y_j in the audience *RankingScheme*. A *RankingScheme* is of type `(value` \times `value)` \rightarrow `boolean`. We assume a *RankingScheme₁* is define for **ExArS₁**.

In addition to these components, arguments are assigned a value.

Definition – Argument-Value Ascription

- $\forall x \in \text{Arg}_1 \exists y \in \text{Val}_1$ $\text{assignArgVal}(x) = y$, where assignArgVal is a function from arguments to values. We assume assignArgVal_1 is defined for **ExArS₁**.

Definitions of argument defeat, acceptability, conflict-free, and admissibility are relativized to audiences and the values of the arguments. Notice, in particular, that the success or failure of an argument attack is *determined with respect to a property associated with an argument*. Other extensions relativize the analysis in other ways. As we define other extensions, we *subscript* defined terms to the particular extension. We assume that sets S and R are subsets of Arg_1 .

Definition – Defeat, Acceptability, and Admissibility

- For arguments $x, y \in \text{Arg}_1$, x *defeats₁* y with respect *RankingScheme₁* if: $\langle x, y \rangle \in \text{Rel}_1 \wedge \neg [\langle \text{assignArgVal}(y), \text{assignArgVal}(x) \rangle \in \text{RankingScheme}_1]$
- An argument x is *acceptable₁* to the subset S : $\forall y \in \text{Arg}_1$ that $\text{defeat}_1 x, \exists z \in S$ that $\text{defeats}_1 y$.
- A subset R of Arg_1 is *conflict-free₁*: $\forall x \forall y [\langle x, y \rangle \in R \times R, \text{either } \langle x, y \rangle \notin \text{Rel}_1 \text{ or } \neg [\langle \text{assignArgVal}(y), \text{assignArgVal}(x) \rangle \in \text{RankingScheme}_1]]$
- A subset R of Arg_1 is *admissible₁* if: R is *conflict-free₁* and $\forall x \in R$ is *acceptable₁* to R .

As with the Dungian framework, we can define notational variants of the notions of *preferred extensions* and *subjectively acceptable credulously*.

To this point, we have but recast familiar argumentation frameworks into our language. The advantage, as shown in the subsequent section, is that we can then *extend* the basic components of these frameworks to address a range of additional issues and problems in the argumentation literature while keeping the basics of the framework intact.

Further Extensions

In this section, we extend the argumentation frameworks to account for issues in argumentation which have not previously been accounted for in a Dungian style analysis. First, we consider a case where we have *multiple audiences*, which may themselves be ordered; in this case, we *differentiate* attack relations, which is to label the arcs. In the second extension, we incorporate the *substructure* of arguments, making reference to premises, conclusions, and rules; again, attacks may be labelled with respect to what *substructure* they attack and a ranking may be placed on them. In the third extension, we show how precedence and procedural contexts

of legal hierarchies can be expressed. In the fourth extension, we present key notions of an analysis of the Burden of Proof in an extended system.

Multiple Audiences – ExArS₂

In ExArS₁, we had but one ranking of values (i.e. one audience), and attacks succeeded or failed with respect to rankings and values on arguments. However, in an argument, there may be two or more audiences, which means there are two (or more) different rankings of the values. In this case, we must consider *Multi-agent Systems*, where we understand the audiences to be the agents. For instance, we can label one audience for a government official *GovtOfficial* and another for a religious minister *RelMinister*: in $\text{RankingScheme}_{GovtOfficial}$, v_i is preferred to v_j while in $\text{RankingScheme}_{RelMinister}$, v_j is preferred to v_i . Clearly, these conflict. Any two (or more) distinct audiences could be so represented. Moreover, the different audiences may have different capacities to argue about the outcome of an attack. For example, suppose that the values are $v_i = \text{economic well-being}$ and $v_j = \text{spiritual well-being}$, where in certain debates, the government official's valuation of the arguments is paramount, while in others it is the religious ministers, leaving aside exactly how such determinations are made.

To abstractly account for such cases, we subsort the ranking schemes and the attack relations. For each sort, we have subsorts to represent the *GovtOfficial* and the other represents the *RelMinister*. For simplicity, we refer to the subsorts as *agents*. The outcome of the attack relation is relative to the label on the node: if the arc is labelled with *GovtOfficial*, then we use the ranking scheme for the *GovtOfficial* in order to determine the outcome of the attack; if the arc is labelled with *RelMinister*, then we instead use the ranking scheme for the *RelMinister*.

For clarity, we provide the extension along with an example of two ranking schemes. We turn to the attack relations in a moment.

Definition – ExArS₂

- Arg_2 is $\{a_1, a_2, a_3, a_4\}$.
- Val_2 is $\{v_1, v_2\}$ and RankingScheme_2 is defined with respect to Val_2 .
- assignArgVal_2 is a function from elements of Arg_2 to elements of Val_2 .
- Rel_2 is the attack relation defined on Arg_2 .
- $\text{RankingScheme}_{RelMinister} \subset \text{RankingScheme}_2$ is $\{\langle v_2, v_1 \rangle\}$
- $\text{RankingScheme}_{GovtOfficial} \subset \text{RankingScheme}_2$ is $\{\langle v_1, v_2 \rangle\}$.

For example, suppose the following values of arguments.

Definition – Argument-Value Ascription

- $\text{assignArgVal}_2(a_1) = v_1$
- $\text{assignArgVal}_2(a_2) = v_2$
- $\text{assignArgVal}_2(a_3) = v_1$
- $\text{assignArgVal}_2(a_4) = v_2$

The key novelty is to introduce *labels* for the arcs, where subsorts of the attack relation.

Definition – Subsorts of Argument Attacks

- $\text{argAttacks}_{RelMinister} \subset \text{Rel}_2$ is $\{\langle a_1, a_2 \rangle, \langle a_2, a_1 \rangle\}$
- $\text{argAttacks}_{GovOfficial} \subset \text{Rel}_2$ is $\{\langle a_2, a_3 \rangle, \langle a_3, a_4 \rangle, \langle a_4, a_3 \rangle\}$

The subsorts of attacks represent who has the *control* of the attack. Depending on which agent has control of the attack, we relativize the ranking scheme to that agent's values. Thus, we express *which agent's values determine the success or failure of the attack*. We assume a function from arcs to ranking schemes:

Definition – Function from Arcs to Ranking Schemes

- arcRankFun is a function from arcs to ranking schemes: $\langle \text{argument}, \text{argument} \rangle \rightarrow \text{RankingScheme}$, where $\langle \text{argument}, \text{argument} \rangle \in \text{RankingScheme}$. We assume a arcRankFun_2 is defined relative to ExArS₂.

We have given an example of how attack relations are relativized, but note that additional definitions are required. For instance, we could *partition* the attack relations so as to avoid conflicts between the agents; alternatively, two attacks have the same arc, we could order the attack relations, giving priority to one over the other. Both could be defined with a *procedural context*. However, a full specification is left to future work.

With this, our definitions for notions such as argument defeat and admissibility are relativized to the ordered values of the audience, assuming S and R are subsets of Arg_2 .

Definition – Attack, Acceptability, and Admissibility

- For arguments $x, y \in \text{Arg}_2$, x defeats₂ y with respect to the values of the agent which controls that arc if: $\langle x, y \rangle \in \text{Rel}_2 \wedge \neg[\langle \text{assignArgVal}_2(y), \text{assignArgVal}_2(x) \rangle \in \text{arcRankFun}_2(x, y)]$.
- An argument x is acceptable₂ to the subset S if: $\forall y \in \text{Arg}_2$ that defeats₂ x , $\exists z \in S$ that defeats₂ y .
- A subset R of Arg_2 is conflict-free₂ if: $\forall x \forall y [\langle x, y \rangle \in R \times R$, either $\langle x, y \rangle \notin \text{Rel}_2$ or $\neg[\langle \text{assignArgVal}(y), \text{assignArgVal}(x) \rangle \in \text{arcRankFun}_2(\langle x, y \rangle)]]$.
- A subset R of Arg_2 is admissible₂ if: R is conflict-free₂ and $\forall x \in R$ is acceptable₂ to R.

With these definitions, the set $\{a_2, a_3\}$ is admissible₂ in ExArS₂. Given but one value ranking as in ExArS₁, we could get a different result. However, we would not be able to represent distinct controls over attacks.

Carneades (Gordon & Walton 2006)

In (Gordon & Walton 2006), an argumentation theory is presented where argument objects are related to statements so as to provide *mereological structure*, in which arguments have assumptions, a conclusion, and a reasoning relation. Assumptions are further subdivided into *premises*, *presuppositions*, and *exceptions*; however, we shall only have space to consider assumptions in general. With such structure, we can represent fine-grained argument attacks such as attacks on assumptions, conclusions, and reasoning relations

as found in common-sense argumentation. This is relevant not only to incorporate well-known approaches to argumentation into a Dungian style framework (e.g. Toulmin Structures), but more importantly to provide an analysis of *procedural contexts*, wherein different sorts of arguments and argument attacks are allowable in a given context (cf. (Wyner & Bench-Capon 2007)). While (Gordon & Walton 2006) use *argument* objects, their approach cannot be characterized as an extension of ExArS_0 as they do not provide definitions for attack or admissible sets of arguments. In this paper, we do not have space to fully justify or present (Wyner & Bench-Capon 2007), as our main focus is to show how the Dungian framework can be systematically extended. Therefore, we focus on just a few key definitions.

We want an argumentation framework which account for the structure of the following commonsense argument, where given assumptions (Assum) and a reasoning rule (implicit here), we can draw a conclusion.

A Commonsense Argument:

- Assum: John was seen to enter the house at 2pm.
- Assum: John was seen to leave the house at 3pm.
- Assum: John could not leave the house unobserved.
- Assum: There is no secret passage out of the house.
- Conclusion: Therefore, John was in the house at 2:30pm.

More to the point, we want an argumentation framework which allows one argument to *attack* the assumptions, conclusion, or reasoning rule of the other argument, as is found in argumentation. For instance, one might attack the *eyewitness testimony* that *John was seen to enter the house at 2pm*, and thus call the conclusion into question.

We do not need values or multiple agents as in ExArS_2 . Mereologically, an argument has assumptions, a conclusion, and a reasoning relation which systematically maps assumptions to a conclusion. We introduce statements and reasoning relations as first-class objects, which are in relation to an argument.

Definition – ExArS_3 Objects

- Arg_3 is a set of arguments $a_1, \dots, a_2, a_3, a_4$.
- Stat is a set of atomic statement names s_1, \dots, s_n , which denote atomic propositions. Statement names are of type *statement*. If s is a statement, then $\neg s$ is a statement. In no model can s and $\neg s$ both hold in any context; s and $\neg s$ are called *contraries*. $\neg s$ is the only complex statement.
- ReasRel is a set of reasoning relation names r_1, \dots, r_n , which are of type *reasoning relation*.

The assumptions and reasoning relations are related to the argument.

Definition – ExArS_3 Relations

- If s is a statement and a is an argument, then $\text{assum}(a,s)$ is a well-formed relation on arguments. It is read as *the statement s is an assumption of argument a* . The assumption relation is of type $(\text{argument} \rightarrow \text{statement}) \rightarrow \text{boolean}$; it is a many-to-many relation.

- If a is an argument and r is a reasoning relation name, then $\text{reasRelFunc}(a) = r$ is a function from arguments to reasoning relations. It is read as *the reasoning relation of argument a is r* . The function is of type $\text{argument} \rightarrow \text{reasoning relation}$.

A conclusion of an argument is a statement which holds with respect to the set of assumptions and some reasoning relation. The following defines the set of assumptions.

Definition – Set of Assumptions

- $\lambda s \text{ assum}(a,s)$ is the set of statements which are assumptions of a given argument a . It is a set of type $(\text{statement} \rightarrow \text{boolean}) \rightarrow \text{argument}$.

A conclusion is a statement which is functionally related to the argument, assumptions, and reasoning relation. For our purposes here, we assume an argument only has *one* conclusion just as it has only *one* reasoning relation.

Definition – Conclusion Relation

- If a is an argument name, A is a set of assumptions, r a reasoning relation name, s a statement, then $\text{conclusion}(a, \lambda s \text{ assum}(a,s), r) = s$ is a function from an argument, assumptions, and a reasoning relation. It identifies the statement s which follows from assumptions $\lambda s \text{ assum}(a,s)$ and reasoning relation r of argument a .

We assume that two arguments a_1 and a_2 are identical when they have the same assumptions, conclusions, and reasoning relations. Furthermore, one argument a_1 is a subargument of another argument a_2 if the conclusions and reasoning relations of a_1 are the same as a_2 , but the assumptions of a_1 is a proper subset of a_2 . In addition, given two arguments with the same assumptions and reasoning relation, the same conclusion must follow. For brevity, these assumptions are not given here, but formally specified in (Wyner & Bench-Capon 2007). With them, we have the following:

Definition – Argument Distinction

- $\forall x, y \in \text{Arg}_3 \text{ distinctArg}(x,y)$ if: x and y are not identical arguments and neither is a subargument of the other.

In these terms, we can *analytically* define the notion of attack: the arguments are not only distinct, but their conclusions are contraries.

Definition – General Argument Attack

- Where $\text{argAtt} \subseteq (\text{Arg}_3 \times \text{Arg}_3)$, $\forall x, y \text{ argAtt}(x,y)$ if: $\text{distinctArg}(x,y) \wedge \text{conclusion}(x, \lambda s(\text{assum}(x,s)), r_x) = \neg \text{conclusion}(y, \lambda s(\text{assum}(y,s)), r_y)$. We read this as *x attacks y* .

This definition correlates to the more familiar *rebuttal* attack; it claims that *any* attack of one argument on another is at least an *attempt* to rebut. In contrast, it is unclear in virtue of what one argument attacks another in (Dung 1995), and by the same token, in virtue of what arguments hold together in an admissible set.

We have defined the most general sort of attack. However, we can have *subsorts of attacks keyed to the mereological structure of the arguments*, which is an analytic basis

of labelling the attack arcs. For our purposes here, we can have attacks on assumptions or attacks on rules, though one could define other sorts of attacks given other subproperties of arguments. In effect, the subset of attack expresses *why the conclusion is denied*.

An *attack on the reasoning relation* means that one argument attacks another and the arguments *differ in terms of the reasoning relation*. This correlates to the more familiar *undercutting* attack.

Definition – Reason Relation Attack

- Where $\text{reasonRelAttack} \subset \text{argAtt}$, $\text{reasonRelAttack}(a_1, a_2)$ if: $\text{reasonRelFun}(a_1) \neq \text{reasonRelFun}(a_2)$.

We can say here that the difference in the conclusion is *attributed to the differences in the reasoning relations that are applied*. In other words, the *reason* why we do not accept the conclusion is because we do not accept the reasoning relation which led to the conclusion.

An *attack on an assumption* means that the assumption of one argument is the contrary of the assumption of another argument. This correlates to the more familiar *premise defeat*.

Definition – Assumption Attack

- Where $\text{assumAttack} \subset \text{argAttack}$, $\forall y, z$, $\text{assumAttack}(y, z)$ if: $\exists x [x \in \lambda s(\text{assum}(y, s)) \wedge \neg x \in \lambda s(\text{assum}(z, s))]$

To this point, we have define subsorts of attacks in terms of reasoning relations and assumptions. Given additional properties ascribed to arguments, we could define further subsorts of attacks such as attacks on *presuppositions* or *exceptions* as in (Gordon & Walton 2006).

Given subsorts of attacks, we can relatively define *subsorts for defeat, acceptability, conflict-free, and admissibility*. Such subsorts may be associated with different *procedural contexts*. For instance, in the appeals process in a legal hierarchy, disputes about matters of fact may be carried out at the court of first instance; however, upon appeal to a higher court, the facts as determined by the lower court are taken as givens and not open to further dispute. In our terms, at the lower court, there may be attacks on assumptions, but not at the higher court. Such a shift in procedural context need not imply that *other sorts of attacks are ruled out*; for instance, while attacks on assumptions might not be legitimate in the higher court, attacks on the reasoning relation might still be legitimate. In virtue of such an attack, a previous conclusion could be overturned (cf. (Wyner & Bench-Capon 2007)).

The definition of *defeat* and *admissibility* most similar to (Dung 1995) would use the *argAtt*, as follows.

Definition – Defeat, Acceptability, and Admissibility for Contraries

- For arguments $x, y \in \text{Arg}_3$, x defeats₃ y with respect to contrary conclusions if: $\langle x, y \rangle \in \text{argAtt} \wedge \neg \exists z \in \text{Arg}_3 \langle z, x \rangle \in \text{argAtt}$.
- An argument x is acceptable to the subset S if: $\forall y \in \text{Arg}_3$ that defeats₃ x , $\exists z \in S$ that defeats₃ y .

- A subset R of Arg_3 is conflict-free₃ if: $\forall x \forall y [\langle x, y \rangle \in R \times R$, either $\langle x, y \rangle \notin \text{Rel}_3$ or $\neg [\langle \text{assignArgVal}(y), \text{assignArgVal}(x) \rangle \in \text{arcRankFun}_3(\langle x, y \rangle)]]$.
- A subset R of Arg_3 is admissible₃ if: R is conflict-free₃ and $\forall x \in R$ is acceptable₃ to R .

Alternatively, instead of defining these notions based on *argAtt*, we could define a notion of defeat with respect to assumptions.

Definition – Defeat with respect to Assumptions

- For arguments $x, y \in \text{Arg}_3$, x defeats₄ y with respect to assumptions: $\langle x, y \rangle \in \text{assumAttack} \wedge \neg \exists z \in \text{Arg}_3 \langle z, x \rangle \in \text{assumAttack}$.

Another approach would be, as in *value-based approaches*, to impose an ordering on the attack relations, so making the defeat of an argument depend on an additional ordering parameter. Clearly, other notions can be defined given the different ways that arguments can attack and defeat one another. However, we leave further refinement and application for future research (cf. (Wyner & Bench-Capon 2007) for one extension of such notions to account for precedent and procedure in legal hierarchies of appeals).

Future Work

In this section, we briefly sketch two other extensions to account for additional aspects of argumentation.

Critical Questions

(Walton 1996) proposes a range of *argumentation* schemes such as *argument from expert opinion*, *argument from eye-witness testimony*, and *argument from exceptional case*, among others. For each argument scheme, critical questions can be posed, the answers to which support or undermine the argument; for example, *Is the expert really qualified to offer an opinion on the case at hand?* (Atkinson 2005) extends the range of critical questions. However, though (Atkinson 2005) broadly uses the Dungian Framework, the critical questions are outside the framework.

We suggest that an extension to the Dungian Framework along the lines of that for Carneades of (Gordon & Walton 2006). The idea is that we additional properties of statements in an argument such that the arguments can be attacked in specific ways. An attack is associated with a question; the way in which we specify defeat with respect to the attack defines the answer to the question. For example, suppose the critical question is *Does the expert witness have the credentials to provide testimony?* This is question that is answered either *Yes* or *No*. The question suggests that the answer is not obvious, so it constitutes an attack on the argument. The criteria of defeat for such a question constitutes *how* the question is answered, that is, the criteria by which the argument under attack stands, where the answer is yes, or falls, where the answer is no. In this way, a critical question is another sort of attack by one argument on one argument, which can defeat it.

Burden of Proof

The *Burden of Proof* refers to the obligation on a party to defend a position against attack. There are three types – legal burden, evidentiary burden, and tactical burden – which vary in terms of which party in the dispute bears the obligation and on what issues. We consider what we take to be the core issue, namely, that a party at a time bears the burden of proof, and the burden of proof implies that an argument which is not yet *admissible* must be found to be admissible or otherwise the argument is defeated.

According to (Prakken & Sartor 2006), the Burden of Proof cannot be accounted for in a Dungian Framework. We suggest an extension to account for the core issue. We introduce participants and the notion of satisfying the Burden of Proof in a framework where admissible sets of arguments can be extended with additional arguments. We assume that we need not define the set of admissible arguments with respect to the whole domain of arguments, but can do so with respect to a partial domain. Thus, there can be a set of admissible arguments and a set of arguments which could be added to it to yield an admissible argument.

For example, consider a case of property, where the defendant claims to own a piece of property against the claim of the plaintiff. In this case, the defendant's claim is under attack by the plaintiff's claim. Let us suppose that the defendant has the burden of proof. This means that the defendant must provide arguments to the court that support his claim. Failing to do so, the defendant's claim fails, and the plaintiff's claim succeeded. But in providing arguments to support his claim, the defendant is *attacking all arguments which attack his claim*. Having succeeded in this, the defendant can *add* his claim to the set of arguments which were previously admissible.

Comparisons

The paper is based on a range of sources ((Dung 1995), (Bench-Capon 2003), (Gordon & Walton 2006), (Wyner & Bench-Capon 2007), (Atkinson 2005), and (Prakken & Sartor 2006)). Our aim has been not to replace these approaches, but to adopt and adapt them into a cohesive and coherent formal argumentation framework, while retaining the key observations and analyses. The key novel contributions of the paper are in two areas. First, we have provided a general format to extend the Dungian Framework in a number of fruitful directions to account for an additional spectrum of problems in argumentation. Second, we have introduced and applied *labelled arcs* in two extensions, showing how these can be used to represent and reason about complex issues in argumentation. We have also suggested that the extensible argumentation system can be extended further to cover a critical questions, case-based reasoning, and burden of proof.

Conclusion

In this paper, we have shown how the Dungian Argumentation Framework can be extended in a variety of ways to address additional aspects of argumentation which had not previously been provided for. The manner of the extensions

allows these additional aspects to be presented in a uniform and consistent way. Key among the extensions is the creation of *labelled arcs* which represent attack relations. With such arcs, we can distinguish sorts of attacks, which leads to a range of different ways to define admissible sets of arguments.

References

- Amgoud, L., and Cayrol, C. 1998. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 1–7. San Francisco, CA: Morgan Kaufmann.
- Atkinson, K. 2005. *What Should We Do?: Computational Representation of Persuasive Argument in Practical Reasoning*. Ph.D. Dissertation, Department of Computer Science, University of Liverpool, Liverpool, United Kingdom.
- Bench-Capon, T. J. M. 2003. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.* 13(3):429–448.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–358.
- Gordon, T., and Walton, D. 2006. The carneades argumentation framework: Using presumptions and exceptions to model critical questions. In Dunne, P. E., and Bench-Capon, T., eds., *Computational Models of Argument: Proceedings of COMMA 2006*, 195–207. Amsterdam: IOS Press.
- Prakken, H., and Sartor, G. 2006. Presumptions and burdens of proof. In van Engers, T., ed., *Legal Knowledge and Information Systems. JURIX 2006: The Nineteenth Annual Conference*, 21–30. Amsterdam: IOS Press.
- Walton, D. 1996. *Argumentation Schemes for Presumptive Reasoning*. Mahwah, N.J.: Erlbaum.
- Wyner, A., and Bench-Capon, T. 2007. Precedence and procedure: an argumentation-theoretic approach. In *Proceedings of the International Conference on Artificial Intelligence and the Law*. Under Review.