Guido Boella
Dov M. Gabbay
Leendert van der Torre
Serena Villata

Meta-Argumentation Modelling I: Methodology and Techniques

Abstract. In this paper, we introduce the methodology and techniques of meta-argumentation to model argumentation. The methodology of meta-argumentation instantiates Dung's abstract argumentation theory with an extended argumentation theory, and is thus based on a combination of the methodology of instantiating abstract arguments, and the methodology of extending Dung's basic argumentation frameworks with other relations among abstract arguments. The technique of meta-argumentation applies Dung's theory of abstract argumentation to itself, by instantiating Dung's abstract arguments with meta-arguments using a technique called flattening. We characterize the domain of instantiation using a representation technique based on soundness and completeness. Finally, we distinguish among various instantiations using the technique of specification languages.

Keywords: Abstract Argumentation, Modelling, Reasoning, Artificial Intelligence.

1. Introduction

Consider the dialogue between the two lawyers in Figure 1. They are arguing about the argumentation of the suspect Jack The Killer, who is accused of being the assassin of Sir John Ashley. Lawyer 1 observes that "argument *a common clerk cannot enter the house of Sir John* attacks *the argument Jack The Killer killed Sir John*" but lawyer 2 argues that "argument *Jack was the administrator of Sir John's fortune* attacks the attack between the argument *a common clerk cannot enter the house of Sir John* and the argument *Jack The Killer killed Sir John*".

Or consider two politicians arguing about social welfare, using arguments like "employment will go up" or "productivity will go down". Two commentators observing the debate may argue about it, using arguments like "the argument "employment will go up" is accepted by the politicians" or "the politicians accept that the argument "employment will go up" supports the argument that "productivity will go down"." This phenomena of people arguing about other people's arguments is common: lawyers argue

Presented by **Name of Editor**; *Received* December 1, 2005

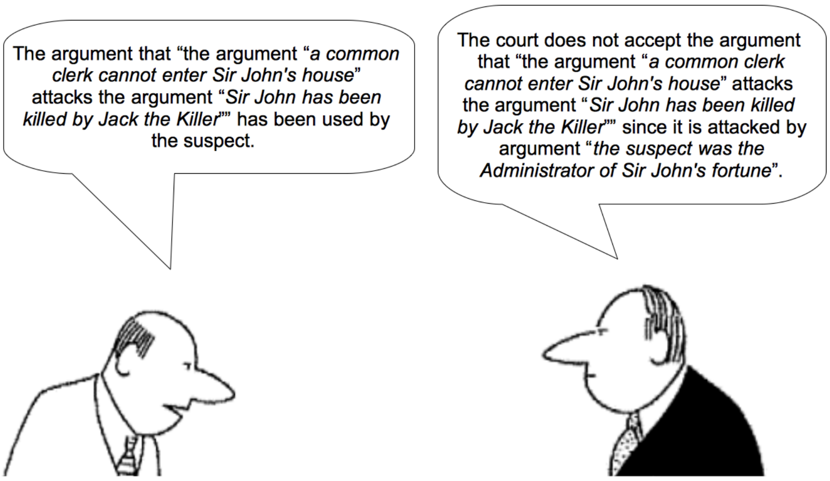

The argument that “the argument “*a common clerk cannot enter Sir John's house*” attacks the argument “*Sir John has been killed by Jack the Killer*”” has been used by the suspect.
The court does not accept the argument that “the argument “*a common clerk cannot enter Sir John's house*” attacks the argument “*Sir John has been killed by Jack the Killer*”” since it is attacked by argument “*the suspect was the Administrator of Sir John's fortune*”.

Figure 1. A dialogue between two lawyers about suspect’s arguments.

about the argumentation of suspects in a courtroom, citizens argue about the argumentation of politicians when making their voting decisions during elections, teachers may argue about the argumentation of their students when evaluating their exams, and parents may argue about their children’s argumentation when arguing how to raise their children. We call this arguing about argumentation *meta-argumentation.*

Meta-argumentation has received little attention thus far. On the one hand, Jakobovits and Vermeir[34] present how to use labelings to define what arguments should be accepted or not. All of the labelings and restricted labelings of the argumentation framework, together with their attacks, are represented in the meta-argumentation framework. On the other hand, Cayrol and Lagasquie-Schiex [25] presents a meta-argumentation framework in which are represented two kinds of binary relations between the arguments, the attack relation and the support relation. A recent approach to meta-argumentation has been presented by Modgil and Bench-Capon [43] where an extension of Dung’s argumentation framework enabling the integration of meta-level reasoning about preferences is presented. For a further discussion on these uses of meta-argumentation in the literature, see Section 4.

In this paper we propose meta-argumentation as a general methodology and technique to model argumentation. It is inspired by the examples of the lawyers, commentators, citizens, teachers and parents, but it is also going beyond such examples when the arguers and the meta-arguers are the *same* reasoners. For example, a lawyer may not only argue whether an argument of a suspect attacks another argument, but he may also argue in a similar way about his or her own arguments. As another example, people may be arguing, but then question the rules of the dialogue game, and argue about them, as shown by Figure 2. The child is arguing that "argument *I was ill* attacks argument *I have to do my homework*" but then he finds that "argument *I have a nice tan* attacks argument *I was ill*".

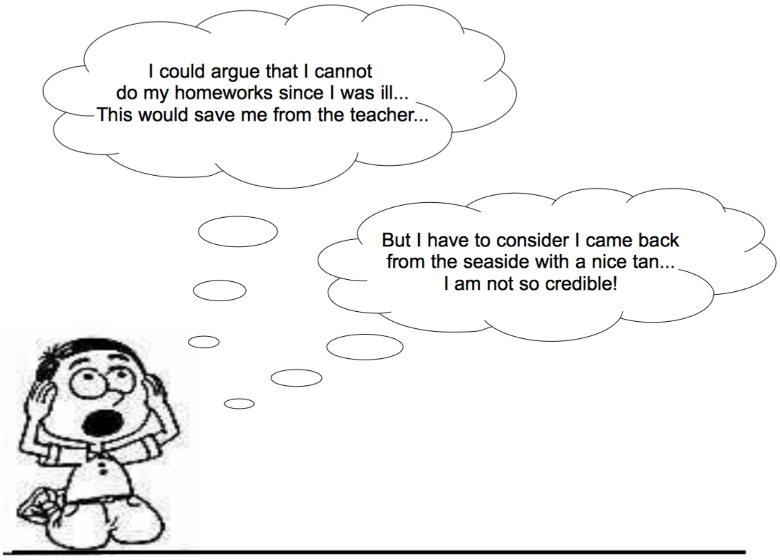

I could argue that I cannot do my homeworks since I was ill... This would save me from the teacher...
But I have to consider I came back from the seaside with a nice tan... I am not so credible!

Figure 2. A child arguing about his own arguments.

The motivation of our meta-argumentation methodology comes from the well known and generally accepted observation that Dung's theory of abstract argumentation cannot be used directly when modeling argumentation in many realistic examples, such as multiagent argumentation and dialogues [11], decision making [38], coalition formation [1], combining Toulmin's micro arguments [50], normative reasoning [5], or meta-argumentation. When Dung's theory of abstract argumentation cannot be applied directly, there are two methodologies to model argumentation using the theory, which leads to the dilemma of choosing among these two alternatives.

Instantiating abstract arguments. Starting from a knowledge base, a set of arguments is generated from this base, and the attack relation among the arguments is derived from the structure of the arguments [46].

Extending Dung's framework. Alternatively, the description of argumentation frameworks is extended, for example with preferences among abstract arguments [3, 35], abstract value arguments [9], second- and higher-order attack relations [41, 8, 42], support relations among abstract arguments [25], or priorities among abstract arguments [47].

We argue in this paper that the dilemma can be resolved using our meta-argumentation methodology, because it is a merger between the methodology of instantiating abstract arguments on the one hand, and extending argumentation frameworks on the other hand. As we recently observed [32], we can instantiate Dungs theory with meta-arguments, *such that we use Dung's theory to reason about itself.* E.g., one may argue whether "don't throw rubbish on the floor!" counts as an argument or not, whether it counts as an attack on "be free!", or whether it supports "respect other people!", or which argumentation semantics should be used. It combines the best of both worlds by instantiating Dung's abstract argumentation theory with an extended argumentation theory. In contrast to the apparent choice between the two commonly used methodologies, our motto is that the instantiation *is* the extension. In other words, an instantiation in the above sense may be seen as a special kind of extension, namely an extension which cannot be further extended. This perspective has several useful consequences. For example, an extension may be seen as an intermediate step between Dung's theory and its instantiation, and extensions can be combined. In this paper, we address the following question:

- How to use meta-argumentation as a general methodology for modeling various kinds of argumentation?

The general research question breaks down in the following sub-questions:

1. What is the methodology of meta-argumentation, and how does it build on established ideas in formal argumentation? We focus here on ideas in abstract argumentation, since the existing notion of abstraction is a good starting point to define meta-argumentation.
2. What are the techniques of meta-argumentation, and how do they build on existing new ideas in argumentation? We focus here on flattening algorithms for fibring argumentation frameworks [31, 30], representation techniques for extended argumentation [36, 37], and specification formalisms and logics of argumentation [15, 29, 13, 53].

Figure 3 provides an abstract example of argument instantiation. Argument $a \rightarrow b$ is instantiated by arguments a and b attacking each other and by a preference relation in which a is preferred over b. This preference relation may also be represented by means of a third argument c attacking the attack $b \rightarrow a$ in such a way to establish the preference of a.

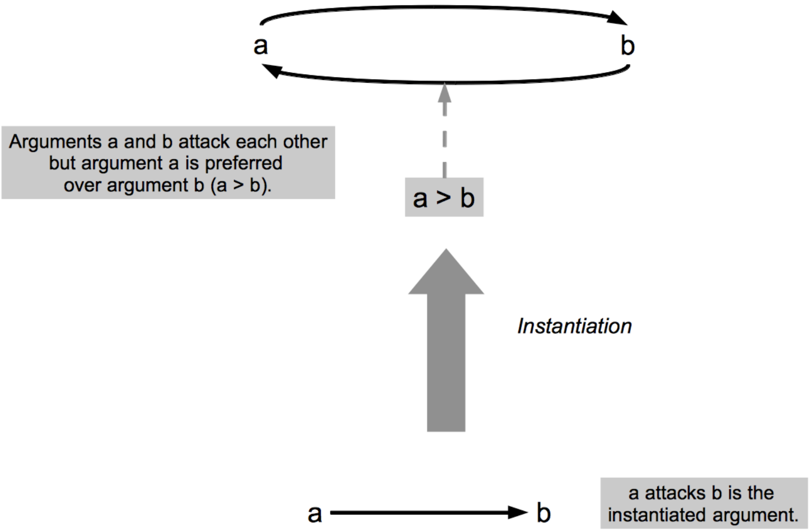

a
b
Arguments a and b attack each other but argument a is preferred over argument b (a > b).
a > b
Instantiation
a
b
a attacks b is the instantiated argument.

Figure 3. Instantiation of an abstract argument.

We consider three techniques used in meta-argumentation: flattening, representation and specification languages. For higher-order attacks, in Boella *et al.* [32] we use the Jakobovits-Vermeir [34] and Caminada [23] labeling to introduce meta-arguments like 'argument A is accepted' or 'argument A is undecided'. Following several similar proposals in the recent literature [42, 31, 30], we use X and Y meta-arguments to model second- and higher-order attacks. Here we use for higher-order attacks a flattening technique introduced by Gabbay [31, 30], which may be seen as a generalization of our earlier work, as well as a growing body of other earlier work [6, 43, 41, 19, 18]. It is based on the introduction of attack meta-arguments $X_{a,b}$ and $Y_{a,b}$, where $Y_{a,b}$ represents that the attack of argument a to argument b is in force, such that if a is accepted, b cannot be accepted, and $X_{a,b}$ represents the negation of $Y_{a,b}$.

Our initial approach in [32] as well as other comparable approaches focusses on the use of meta-argumentation to represent preferences and higher order attacks, by introducing meta-arguments for the attacks. In this paper we explain the methodology and techniques using these two examples. In Villata [52], we illustrate the methodology and techniques of meta-argumentation on three other challenges in formal argumentation: the merging of argumentation frameworks in multi-agent argumentation, the representation of a subsumption relation among arguments in argument ontologies, and the representation of the Toulmin scheme when representing and combining micro arguments.

The paper follows the research questions and is organized as follows. Section 2 introduces the methodology of meta-argumentation, starting with a general introduction, introducing Dung's framework and abstraction, extended argumentation frameworks and reductions to Dung's basic theory, and finally Baroni and Giacomin's framework and acceptance functions. Section 3 introduces the techniques by first giving an informal introduction, then introducing flattening functions, representation techniques, and finally specification formalisms. Related work and conclusions end the paper.

2. Meta-argumentation methodology

In this section we explain the methodology of meta-argumentation to model argumentation and we explain how it builds on three well established ideas in argumentation theory: Dung's theory of abstract argumentation, extended argumentation frameworks, and Baroni and Giacomin's study of acceptance functions. The techniques of meta-argumentation are deferred to Section 3.

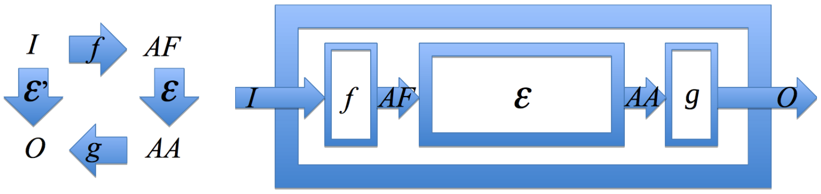
2.1. Meta-argumentation methodology: an informal introduction

We start with an informal introduction about meta-argumentation theory, highlighting the two well known methodologies of extending and instantiating argumentation.

2.1.1. Unifying instantiations and extended argumentation

Dung's argumentation theory formalizes the reasoning leading to accepted arguments, on the basis of attacks among arguments. In Dung's terminology, it is a theory of argumentation semantics, which relates attack relations among arguments to acceptable arguments. In our terminology, it is a theory of acceptance functions. To *use* Dung's theory, we have to describe the arguments and the attack relation, such that we can use one of the argumentation semantics or acceptance functions to obtain the acceptable arguments. The theory does not assume any structure on the arguments, which are therefore called *abstract* arguments, such that the description of the arguments and the attack relation in Dung's theory is unconstrained, and the theory can be used in many contexts. We call a set of arguments together with an attack relation a *basic* argumentation framework, to distinguish it from the extended argumentation frameworks discussed below. We call this use of the theory, based on an instantiation of abstract arguments, an *instantiation* of Dung's theory.

The instantiation of Dung's theory is visualized in Figure 4. Using elementary mathematics, Figure 4(a) describes the instantiation as four functions, where Dung's acceptance is a function $\mathcal{E}$ from argumentation frameworks AF to sets of extensions of acceptable arguments AA, f is a function from argumentation inputs I to argumentation frameworks AF, and g is a function from acceptable arguments to argumentation outputs O. From a system or cybernetic perspective, Figure 4(b) describes the instantiation as an argumentation system, with input I and output O. From a software engineering perspective, we can see it as a (reasoning) component, where f and g are packing and unpacking procedures. Numerous other interpretations are possible too. For example, analogous to Tarski's deductive systems, we can see argumentation as a logical relation between inputs and outputs. Such kinds of interpretations may be useful to obtain formal relations with other theories, but will not play a further role in this paper.



(a) Categories (b) Basic Argumentation System

Figure 4. Instantiating Dung's basic argumentation theory: a function f transforms an argumentation input I to an argumentation framework AF, whose extensions of accepted arguments $AA = \mathcal{E}(AF)$ are transformed back into the argumentation output O. The argumentation output is a function of the argumentation input $O = \mathcal{E}'(I)$, derived from the two transformations and the acceptance function. Summarizing $O = \mathcal{E}'(I) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(I)))$.

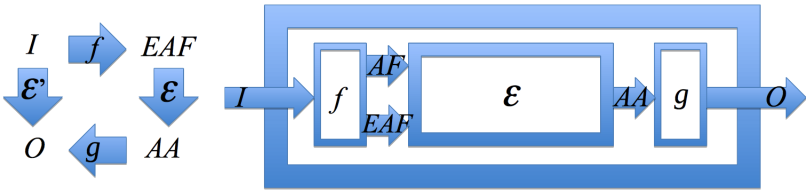
There are several ways in which we can use the diagram of Figure 4.

For example, when we have a formal theory relating some input I to some output O by a function $\mathcal{E}'$, then we can look for functions f and g to complete the diagram. This is what happens when Dung's theory is used as a general theory for reasoning in which conflict resolution plays an important role, where the generality of the theory comes from the fact that many kinds of other reasoning formalisms can use Dung's theory as a substantial part to resolve conflicts. In other words, many theories have been transformed to a binary attack relation among arguments, and the conclusions of the theories can be retrieved from the accepted arguments. Examples of input and outputs in Figure 4 are non-monotonic logic theories and their conclusions, logic programs and their extensions, Reiter default theories and their extensions, decision theories and their decisions, game theories and their solutions, knowledge bases and their conflict free mergers, legal theories, normative theories and their obligations and permissions, and much more. In Dung *et al.* [27], arguments essentially are sets of formulas called assumptions, from which conclusions can be drawn with strict inference rules. In fact, the extensions defined by the various semantics of Bondarenko *et al.* [21] are not sets of arguments but sets of assumptions and in [27] it is shown that an equivalent fully argument-based formulation, as introduced in [26], can be given. In some cases the functions f and g are relatively simple, and the relation between input and output is nearly fully characterized by the argumentation, and in other cases the functions are more complicated, since conflict resolution is only a small part of the reasoning.

Another way to use the diagram is for cases when we have an input I and an output O, but we do not have the relation between them, i.e. we do not have the function $\mathcal{E}'$. The function may be partially known, for example we want the relation between input and output to satisfy some principles, or we have some benchmark examples which we want the function $\mathcal{E}'$ to satisfy. In such a case, instead of defining the function $\mathcal{E}'$ from scratch, we may try to define the functions f and g, and derive $\mathcal{E}'$ from it. For example, in this way we can derive new semantics for logic programs using new argumentation semantics.

The basic picture of using Dung's framework in Figure 4 has been modified by extending Dung's argumentation framework with other relations among abstract arguments, such as preference-based relations [3], value-based relations [9], support relations in bipolar argumentation [25], second- and higher-order attack relations [41, 8, 42] and priorities relations among abstract arguments [47].

The use of an extended argumentation framework is visualized in Figure 5. Figure 5(a) describes the instantiation using again the four functions $\mathcal{E}$, $\mathcal{E}'$, f and g, where acceptance is now a function $\mathcal{E}$ from *extended* argumentation frameworks EAF to sets of extensions of acceptable arguments AA, and f is a function from argumentation inputs I to *extended* argumentation frameworks EAF. As before, g is a function from acceptable arguments to argumentation outputs O. Figure 5(b) describes the related instantiation as an extended argumentation system, which is analogous to the basic argumentation system. The challenge of the extended argumentation theory is to define the acceptance function $\mathcal{E}$ working on extended argumentation frameworks, and to relate this acceptance function for extended argumentation frameworks to Dung's acceptance functions for basic argumentation frameworks.

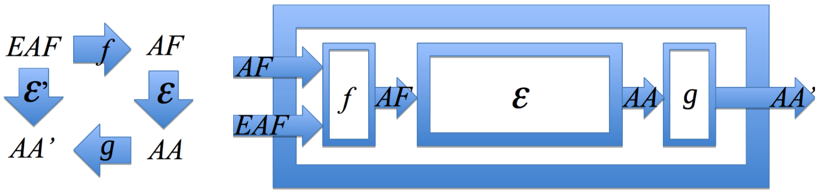


(a) Categories (b) Extended Argumentation System

Figure 5. Extending Dung's theory: a function f transforms an argumentation input I to an extended argumentation framework EAF, which contains besides attack relations among arguments represented in AF also other kind of relations among arguments. As in Figure 4, the argumentation output is a function of the argumentation input $O = \mathcal{E}'(I)$, derived from the two transformations and the acceptance function, $O = \mathcal{E}'(I) = g(AA) = g(\mathcal{E}(EAF)) = g(\mathcal{E}(f(I)))$.

The main idea of a unified methodology is to see extended argumentation framework as an instantiation. This may be seen as a way to answer the challenge to define acceptance functions $\mathcal{E}$ for extended argumentation frameworks, since it defines this acceptance function using Dung's acceptance functions for basic argumentation frameworks. For example, it may define the acceptance function for preference-based argumentation frameworks by defining an attack in the basic argumentation framework as an attack in the extended argumentation framework by an argument which is not less preferred than the attacked argument.

This perspective on extended argumentation frameworks as instantiations is visualized in Figure 6. Figure 6(a) describes the instantiation using again the four functions $\mathcal{E}$, $\mathcal{E}'$, f and g, where acceptance is now a function $\mathcal{E}'$ from *extended* argumentation frameworks EAF to sets of extensions of acceptable arguments AA', as well as a function $\mathcal{E}$ from basic argumentation frameworks to sets of extensions of acceptable arguments AA. Moreover, f is a function from extended argumentation frameworks EAF to basic argumentation frameworks AF, and g is a function from acceptable arguments to acceptable arguments. Figure 6(b) describes the related instantiation as an instantiated argumentation system.

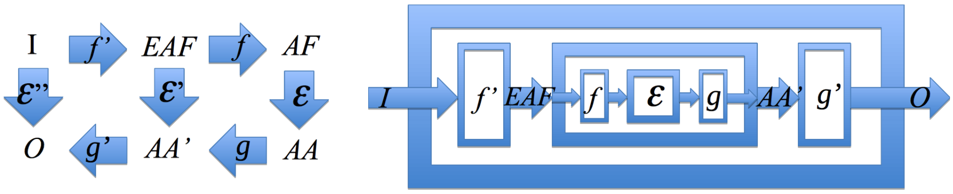


(a) Categories (b) Argumentation System

Figure 6. Extended argumentation framework as an instantiation: a function f transforms an extended argumentation framework AF to a basic argumentation framework AF. As in Figure 4, the accepted arguments of th extended framework are a function of the extended argumentation framework $AA = \mathcal{E}'(EAF)$, derived from the two transformations and the acceptance function of basic argumentation, $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$.

In this unified methodology, it becomes easier to combine instantiations and extended argumentation frameworks. For example, regularly an instantiation represents arguments by logical rules, it defines preferences among arguments, and it distinguishes between undercut and rebut attacks. In such a case, we can define an extended argumentation framework which models the preferences and the two kinds of attacks, but which leaves the arguments abstract. The extended argumentation framework may be seen as an intermediate step between Dung's theory and its instantiation. Moreover, in the same way, extended argumentation frameworks can be combined. For example, we may have an extension with preferences, and an extension which distinguishes among rebut and undercut attacks, and these two extensions can be combined.

This perspective on combining extended argumentation frameworks and instantiations is visualized in Figure 7. Figure 7(a) describes the instantiation using again the various functions by combining the functions from Figure 4(a) and Figure 6(a). Figure 7(b) describes combination as an instantiated argumentation system, which replaces the component E of Figure 4(b) by the whole argumentation system of Figure 6(b).



(a) Categories

(b) Argumentation System

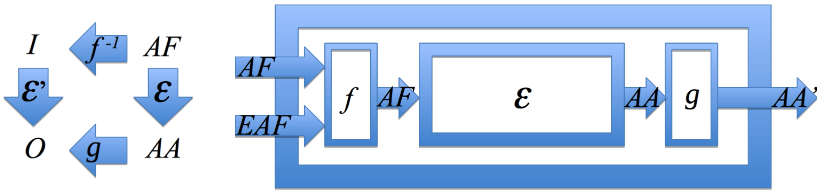
Figure 7. Combining instantiation and extended argumentation frameworks: a function f' transforms an argumentation input I to an extended argumentation framework EAF, and a function f translates this extended argumentation framework to a basic argumentation framework AF. As in Figure 4, the argumentation output is a function of the argumentation input $O = \mathcal{E}''(I)$, derived from the two transformations f' and g', and the acceptance function $\mathcal{E}'$. Moreover, as in Figure 6, the acceptable arguments of the extended argumentation framework are a function of the extended argumentation function $AA' = \mathcal{E}'(EAF)$, derived from the two transformations f and g, and the acceptance function $\mathcal{E}$. Summarizing $O = \mathcal{E}''(I) = g'(AA') = g'(\mathcal{E}'(EAF)) = g'(\mathcal{E}'(f'(I))) = g'(g(\mathcal{E}(f(f'(I)))$.

Summarizing, the functional compositions and the combination of argumentation systems in Figure 7 give two equivalent perspectives on our unification of the two methodologies of instantiating Dung's argumentation framework, and extending it with abstract relations. Sometimes the functional composition is more intuitive or useful, and sometimes the system composition is more useful.

2.1.2. Meta-argumentation methodology

The general methodological problem we consider in this paper is how to use Dung's theory. Using the terminology developed above, we now make this problem more precise. Dung's theory is the theory of acceptance functions $\mathcal{E}$ defined on basic argumentation frameworks and sets of accepted arguments. The use of such a theory is represented by a function $\mathcal{E}'$ from argumentation input to argumentation output. The methodological problem is thus how to develop a theory that transforms acceptance functions $\mathcal{E}$ into other functions $\mathcal{E}'$. This function transformation is the general representation of the use or instantiation of Dung's argumentation theory.

This instantiation problem is visualized in Figure 8. It is the same figure as the instantiation problem of Dung's theory in Figure 4, besides the replacement of function f from argumentation input to argumentation frameworks, by its inverse function f^{-1} from argumentation frameworks to argumentation inputs. We are more precise about this in section 2.4.2, here we discuss when the inverse is a partial function (some elements of the argumentation framework are not mapped to anything), or when it is a multi-valued function, when two argumentation inputs are mapped to the same argumentation framework. This emphasizes that we start with an acceptance function $\mathcal{E}$, and we are looking for functions $\mathcal{E}'$.



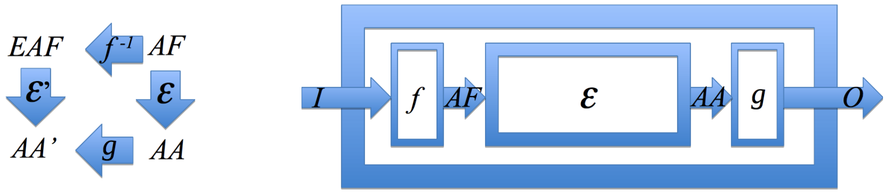
(a) Categories (b) Basic Argumentation System

Figure 8. The methodological problem: how to use Dung's acceptance functions $\mathcal{E}$ to find functions $\mathcal{E}'$ between argumentation input I and argumentation output O? This function transformation consists of two parts: a function f^{-1} transforms an argumentation framework AF to an argumentation input I, and a function g transforms the accepted arguments into argumentation output. Summarizing $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a,b) \in \mathcal{E}\}$.

Usually, the instantiation of a basic argumentation framework maps the arguments to structured arguments. For example, in propositional argumentation, an argument is mapped to a propositional formula, and in explanation-based argumentation, an abstract argument is mapped to a pair (K,p) where K is a set of propositional formulas and p is a propositional formula, where K is explaining the proposition p. If we have an argumentation framework with two argument a and b where argument a attacks argument b but not vice versa, then in the instantiated framework, the argument a may be described by a pair $\langle\{p, p \rightarrow q\}, q\rangle$ and argument b by the pair $\langle\{\neg q, \neg q \rightarrow r\}, r\rangle$. In that case, argument a attacks argument b, because q is inconsistent with the explanation of argument b, but there is no attack vice versa, since r does not occur in the explanation of argument a.

We are interested in the instantiation of basic argumentation frameworks by extended argumentation frameworks. Abstractly, we are interested in the case where an instantiation of Dung's argumentation theory is a function or algorithm from the set of basic argumentation frameworks to a set of extended argumentation frameworks. For example, consider the argumentation framework that contains two arguments "unemployment goes up" and "inflation goes down", and where the former attacks the latter. We can instantiate the argumentation framework by an extended framework where the two arguments attack each other, but the former is preferred to the latter. In the basic argumentation framework the abstract argument that inflation goes up attacks the argument that unemployment goes down but not vice versa, whereas in the instantiated extended argumentation framework the two arguments attack each other, but the argument that unemployment goes up is stronger than the argument that inflation goes down.

Our meta-argumentation approach is a particular way to define mappings from argumentation frameworks to extended argumentation frameworks: the arguments are interpreted as meta-arguments, of which some are mapped to "argument a is accepted," where a is an abstract argument from the extended argumentation framework. In other words, the function f assigns to each argument a in the extended argumentation framework, an argument "argument a is accepted" in the basic argumentation framework. This meta-argumentation methodology is visualized in Figure 9.



(a) (b) Argumentation System

Figure 9. The meta-argumentation methodology: we use Dung's acceptance functions $\mathcal{E}$ to find functions $\mathcal{E}'$ between extended argumentation frameworks EAF and acceptable arguments AA'. This function transformation consists of two parts: a function f^{-1} transforms an argumentation framework AF to an extended argumentation framework EAF, and a function g transforms the accepted arguments of the basic argumentation framework into acceptable arguments of the extended argumentation frameworks. Summarizing $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a, b) \in \mathcal{E}\}$.

2.1.3. Meta argumentation viewpoint

Wooldridge *et al.* [53] argue that one cannot think of argumentation without thinking of meta-argumentation too. They claim that

> Our key motivation is the following observation: *Argumentation and formal dialogue is necessarily a meta-logical process.* This seems incontrovertible: even the most superficial study of argumentation and

> formal dialogue indicates that, not only are arguments made about object-level statements, they are also made about arguments. In such cases, an argument is made which refers to another argument. Moreover, there are clearly also cases where the level of referral goes even deeper: where arguments refer to arguments that refer to arguments.

We call this the meta-argumentation viewpoint. In modeling, a viewpoint is associated with a stakeholder with her concerns and gives rise to views on systems. The methodology of meta-argumentation as a way to model argumentation is based on a conceptualization of argumentation using the relation between two theories of argumentation and meta-argumentation.

We assume a fundamental relation about the relation between these two levels: *meta-argumentation has to be able to mirror argumentation.* For example, when politicians argue, the commentators should be able to argue in the same way. For example, if the politicians use as primitives arguments a from a universe of arguments U, together with a mechanism to derive acceptable arguments from relations among the arguments, and the commentators have as primitives meta-arguments ma from a universe of meta-arguments MU together with a mechanism to derive acceptable meta-arguments from relations among the meta-arguments, then the set of arguments must be reflected in the set of meta-arguments, and there must be a relation between the ways acceptable arguments and acceptable meta-arguments are derived.

Our methodology follows from the fundamental relation between argumentation and meta-argumentation theory: *we can apply a theory of argumentation to itself.* We call this process of applying a theory of argumentation to itself *meta-argumentation.* For example, a teacher would argue that argument “I was ill” of his student does not attack her argument “every day, students have to do their homework” since it is attacked by argument “if you have a nice tan, then you were not ill!”

The meta-argumentation methodology is inspired by ideas in modeling. In modeling, the idea of abstraction and refinement is commonplace. For example, argument $a \rightarrow b$ can be instantiated by arguments a and b which attack each other and by argument c which represents the preference of a over b attacking $b \rightarrow a$. The notion of meta-argumentation modeling raises the question how this kind of modeling relates to other kinds of modeling, and whether insights from general theories of modeling can be used to define a theory of meta-argumentation. Meta-modeling in software engineering is the analysis, construction and development of rules, constraints, models and theories applicable and useful for modeling a predefined class of problems. As its name implies, this concept applies the notions of meta- and modeling. A model is an abstraction of phenomena in the real world while a metamodel is yet another abstraction, highlighting properties of the model itself. A model always conforms to a unique metamodel.

One of the currently most active branch of *Model Driven Engineering* is the approach named model-driven architecture proposed by *OMG*. This approach is based on the utilization of a language to write metamodels called the *Meta Object Facility* or *MOF*, designed as a four-layered architecture. It defines an M3-model, which conforms to itself. Every model element on every layer is strictly in correspondence with a model element of the layer above. *MOF* only provides a way to define the structure, or abstract syntax of a language. Typical metamodels proposed by *OMG* are UML, SysML, SPEM or CWM.

In the same way, the idea of meta-argumentation is to apply argumentation to itself. It is inspired by the unified modeling language (UML), which is used to define itself. Following this analogy, we may say that an argumentation theory is a model of reasoning, and that meta-argumentation theory is a model that of this model of reasoning. UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML includes a set of graphical notation techniques to create visual models of software systems, as we do for meta-argumentation.

An extended argumentation theory is a natural representation for meta-argumentation since it allows to represent every kind of additional relation between arguments, such as preferences, support, subsumption and so on. The extended argumentation framework is defined and this framework becomes a standard Dung's argumentation framework. In the remainder of this section we make these informal ideas more precise. We start introducing Dung's abstract argumentation framework in order to represent how to instantiate arguments, then we discuss meta-argumentation in relation with extended argumentation frameworks. Finally, we discuss Baroni and Giacomin's framework, introducing acceptance functions and principles, which are used in our meta-argumentation methodology and techniques.

2.2. Methodology 1: Instantiating abstract arguments

We first introduce Dung's theory of abstract argumentation, and then we explain how we use it in the meta-argumentation methodology.

2.2.1. Dominance as argumentation

Dominance theory is a theory which takes as input a set of elements and a binary dominance relation, which may have to satisfy some conditions, and produces as output solutions in the form of a subset of the elements [22]. It originates from game theory, where stable sets were introduced as a solution concept in the 1940s. The same structure was used in other areas, for example in decision making for reasoning about preferences: the binary relation now represents that an element is preferred to another one, and the solution is the set of most preferred elements [33]. Various conditions have been studied on the preference relation, for example transitivity.

When the binary relation does not contain cycles, it is straightforward to define the undominated elements, but when there are cycles in the graph, it becomes more problematic to have good intuitions about the expected solution, and it becomes harder to compute solutions given the proposed solution concepts. For example, without cycles it is straightforward to define stable sets, but with cycles it is more problematic.

Dung's theory of abstract argumentation [26] may be seen as a kind of dominance theory where the elements of the set are called arguments, the binary relation is called the attack relation, and the solution is characterized by the principle of reinstatement. The concept of defence has been introduced in order to reinstate some of the defeated arguments, namely those whose defeaters are in turn defeated.

Dung's theory is based on a binary *attack* relation among arguments, which are abstract entities whose role is determined only by its relation to other arguments. Its structure and its origin are not known. We restrict ourselves to *finite* argumentation frameworks, i.e., in which the set of arguments is *finite*.

Definition 1 (Argumentation framework). *An argumentation framework is a tuple $\langle A, \rightarrow \rangle$ where A is a finite set (of arguments) and $\rightarrow$ is a binary (attack) relation defined on $A \times A$.*

The various semantics of an argumentation framework are all based on the notion of defence.

Definition 2 (Defence). *Let $\langle A, \rightarrow \rangle$ be an argumentation framework. Let $\mathcal{S} \subseteq A$. $\mathcal{S}$ defends a if $\forall b \in A$ such that $b \rightarrow a$, $\exists c \in \mathcal{S}$ such that $c \rightarrow b$.*

A semantics of an argumentation theory consists of a conflict free set of arguments, i.e., a set of arguments that does not contain an argument attacking another argument in the set.

Definition 3 (Conflict-free). *Let $\langle A, \rightarrow \rangle$ be an argumentation framework. The set $\mathcal{S} \subseteq A$ is conflict-free if and only if there are no $a, b \in \mathcal{S}$ such that $a \rightarrow b$.*

The following definition summarizes the most widely used acceptability semantics of arguments given in the literature.

Definition 4 (Acceptability semantics). *Let $AF = \langle A, \rightarrow \rangle$ be an argumentation framework. Let $\mathcal{S} \subseteq A$.*

- *$\mathcal{S}$ is an* admissible *extension if and only if it is conflict-free and defends all its elements.*
- *$\mathcal{S}$ is a* complete extension *if and only if it is conflict-free and we have $\mathcal{S} = \{a \mid \mathcal{S}$ defends $a\}$.*
- *$\mathcal{S}$ is a* grounded extension *of AF if and only if $\mathcal{S}$ is the smallest (for set inclusion) complete extension of AF.*
- *$\mathcal{S}$ is a* preferred extension *of AF if and only if $\mathcal{S}$ is maximal (for set inclusion) among admissible extensions of AF.*
- *$\mathcal{S}$ is the* skeptical preferred extension *of AF if and only if $\mathcal{S}$ is the intersection of all preferred extensions of AF.*
- *$\mathcal{S}$ is a* stable extension *of AF if and only if $\mathcal{S}$ is conflict-free and attacks all arguments of $A \backslash \mathcal{S}$.*

Which semantics is most appropriate in which circumstances depends on the application domain of the argumentation theory.

A problem may be raised concerning this terminology, because these so-called semantics do not represent the complete meaning of an argumentation framework. For example, if two argumentation frameworks have the same extensions, are they equivalent? Following ideas in logic programming, we may say that this is the case in a weak sense, but sometimes two argumentation frameworks with the same extensions are not equivalent in the stronger sense that the extensions remain the same if we add arguments or attacks to the argumentation framework. An example of weak $\mathcal{E} - equivalence$ is given in Figure 10. We therefore prefer to refer to acceptance functions over argumentation semantics.

2.2.2. Abstraction in meta-argumentation

We now relate Dung's theory to our notion of meta-argumentation. The basic idea is that the common representation and the common reasoning of

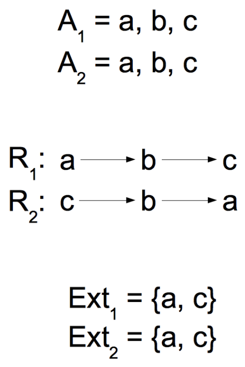


Figure 10. Weakly $\mathcal{E} - \mathit{equivalence}$ between two AF.

argumentation and meta-argumentation is characterized by Dung's theory. In other words, the common idea of both levels of argumentation is the attack among arguments, and a mechanism to select acceptable arguments. The relation between argumentation and meta-argumentation is in the notion of "abstract".

Dung's theory represents the complex way of reasoning about arguments by a relatively simple mathematical structure, directed graphs and a way to associate with directed graphs a subset of the nodes. Dung claims about the abstract nature of its theory in [26]:

> "In the first step, a formal, abstract but simple theory of argumentation is developed to capture the notion of acceptability of arguments. In the next step, we demonstrate the "correctness" (or "appropriateness") of our theory. It is clear that the "correctness" of our theory cannot be "proved" formally. The only way to accomplish this task is to provide relevant and convincing examples. [...] An argument is an abstract entity whose role is solely determined by its relations to other arguments. No special attention is paid to the internal structure of the arguments."

Other interpretations of Dung's argumentation framework abstract nature are given by Prakken and Vreeswijk [45] and Bench-Capon and Dunne [11]. However, in our use of Dung's theory in meta-argumentation, the utilization of abstract mathematics to represent human reasoning is only part of the explanation of the use of the word "abstract" in abstract argumentation. Many ways of reasoning are represented by relatively simple mathematical theories, for example reasoning about decisions is represented by a probability distribution and a utility function, together with a decision rule like maximize expected utility, reasoning about interaction among decision makers is represented by a simple matrix of pay-offs for strategies and a solution concept like the Nash equilibrium, and many other forms of reasoning are represented by logical formalisms with associated reasoning methods. In those cases we normally do not refer to abstract decision making, abstract game theory, or abstract logics. This suggests that there is something more to abstract argumentation.

Our interpretation is based on another understanding of "abstract". To understand the notion of "abstract", we have to consider the argumentation theories that existed before Dung introduced his abstract theory, see Prakken [46] for a discussion. Many of them were more detailed, detailing the structure of arguments, or distinguishing kinds of attacks. Therefore, one may see Dung's abstract argumentation theory as an alternative for these other more detailed theories, using the notion of abstract arguments. However, we believe that Dung's theory was not only an alternative for existing theories, but – and here comes the second meaning of the notion of "abstract" – it was also an *abstraction* of existing theories. At a conceptual level, this notion of abstraction means that Dung's theory generalizes the existing argumentation theories, in the sense that it captures the fundamental properties of the many existing argumentation formalisms around. Some of these fundamental properties are the fundamental concept of attack among arguments, or the idea that a set of arguments can defend an argument against attacks of other arguments, or the idea that the result of argumentation theory is a set of accepted arguments, or the idea that there can be various sets of arguments that can be accepted together. All these ideas can be found in more detailed argumentation theories, and Dung's abstract theory generalizes the existing theories into a general abstract theory.

Our interpretation of "abstract", as an abstraction of existing theory in a uniform abstract language, is a natural concept in modeling and reasoning. For example, when two agents have distinct concepts to describe the world, or reason about them, then a common language may be defined for them to talk to each other. The language may abstract away some concepts which are used only by one of the agents, for example because he is an abstract on the domain described by this concept. For example, in the semantic web, description logic is used as ontology language which requires the adoption of various forms of non-monotonic reasoning techniques, as well as non-standard inferences, in order to describe concepts.

It may be argued that our interpretation of "abstract" is far fetched, because Dung does not show, not even discuss, how his theory can be seen as an abstraction from existing argumentation theories. He applies his theory not to argumentation theory itself, but to logic programming, non-monotonic reasoning, and game theory. Thus he shows that his abstract theory can be used as a general reasoning framework capturing other kinds of reasoning rather than capturing the kind of reasoning about argumentation. However, in our opinion, this does not contradict the idea that Dung's argumentation theory is seen as an abstraction from other argumentation theories. On the one hand Dung's theory abstracts various kinds of argumentation reasoning, and on the other hand the abstract theory can be used to characterize kinds of reasoning in other areas.

2.2.3. Instantiating abstract arguments

Prakken [46] presents the ASPIC framework, a general abstract model of argumentation with structured arguments. The ASPIC framework allows for a general use of inference rules, by expressing the rules through schemes, in the logical sense, with metavariables ranging over the logical language $\mathcal{L}$. Thus, when it is used the framework becomes a general framework for argumentation with structured arguments. The ASPIC framework is extended and generalized in four respects: 1) a third way of argument attack, called premise attack as the result of a combination of “plausible” and “defeasible” argumentation, 2) the attacks' notions are generalized from the notion of contradiction between formulas ϕ and $\neg\phi$ to an abstract relation of contrariness between formulas which is not necessarily symmetric, 3) four kinds of premises are distinguished, 4) attack relations are solved in part with preference relations between arguments, defeasible rules and the knowledge base. Anyway, these kinds of approaches are not unproblematic. For example, as claimed by Caminada and Amgoud [24], even if these systems are suitable in domains like legal reasoning, unfortunately, they fail to meet the objectives of an inference system, leading thus to very unintuitive results. As instance, with these systems it may be the case that an agent believes that “if a then it is always the case that b”, and the system returns as output argument a but not argument b or if the agent also believes that “if c then it is always the case that b, the system may return arguments a and c, which means that the output of the system is indirectly inconsistent. For further details on these issues, see Amgoud and Besnard [2] and Caminada and Amgoud [24].

In general, an instantiation of Dung's theory is based on a set of arguments with internal structure, such that the attack relation among these instantiated arguments can be derived from their internal structure. The internal structure may come from the underlying mechanism of argument generation that produces the universe of instantiated arguments, as mentioned in Section 2.4.1. For example, the instantiated arguments can be constructed from a knowledge-base containing rules or logical formulas. In other words, if the internal structure of two arguments is known in all its details, then from these descriptions can be derived whether they attack each other, whether one attacks the other, or they do not attack each other. For example, if the arguments are described by propositional formulas, then the attack relation may be based on a notion of propositional inconsistency. If the arguments are described by Toulmin schemes, then there can be rebutting attacks when the claims conflict, and undercutting attacks when a claim conflicts with a warrant. An instantiation is thus defined by a set of descriptions of the internal structure of arguments, an attack relation defined for these descriptions, and an instantiation function that associated with each abstract argument an argument description. For example, consider an argumentation framework that contains two arguments, and where the former attacks the latter. We can instantiate the former argument by a rule that “if inflation goes up, then unemployment goes up”, together with the fact that “inflation goes up”, and the latter argument by the fact that “inflation goes down”. The first argument is instantiated by two arguments, one which is a support relation and the other which is an argument, while the second argument is instantiated simply by an argument. Since the arguments composing the first argument attack the argument composing the second one, the former instantiated argument attacks the latter.

2.3. Methodology 2: Extending Dung’s basic frameworks

We first discuss some examples of extended argumentation framework, and then we explain how they fit our theory of meta-argumentation. When representing examples in this theory, such as multiagent argumentation and dialogues [11], Toulmin schemes [50] or examples from normative reasoning [5], the language is typically extended, for example with preferences among arguments [3, 35], value arguments [9], second- and higher-order attack relations [41, 8, 42], support relations among arguments [25], or priorities among arguments [47]. However, that seems to be in conflict with the idea of an *abstract* theory: in principle, it should be instantiated or refined rather than extended [31, 30].

2.3.1. Some examples of extending Dung's basic framework

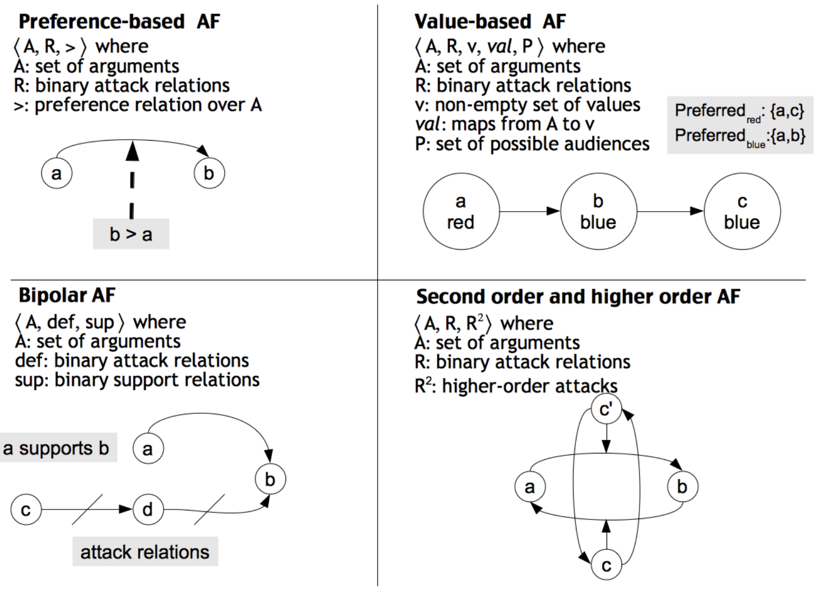


Figure 11. Examples of extended argumentation frameworks.

Four examples of extended argumentation frameworks are illustrated in Figure 11. Preference-based argumentation introduces a preference relation between the arguments. For example, as shown in Figure 11, Amgoud [3] defines a preference-based AF as a triplet $\langle A, R, \prec\rangle$ where A is a set of arguments (in this paper, they represent coalitions structures), R is a binary relation representing a defeat relationship between arguments and $\prec$ is a partial or complete pre-ordering on A. In particular, we have that the notion of defense is define in the following way: let a, b be two arguments such that aRb, then b defends itself against a iff $b \prec a$, as in Figure 11. See Kaci and van der Torre [35] for a further discussion.

Second- and higher-order argumentation frameworks introduce in Dung's standard argumentation framework a new kind of attack $\rightarrow^2$, which is a binary relation between arguments and attack relations. Roughly, these attacks are attacks raised from an argument against another attack relation. This introduces a new interpretation of the notion of attack in which both the arguments are accepted, only the attack relation is attacked. Modgil [41] observes that a preference of argument a over argument b can be seen as an attack on the attack from b to a, in the sense that if a is preferred to b, then b cannot attack a. The author introduces a three place attack relation, which we call here second-order attack, and it is defined as $\langle A, R, R^2 \rangle$ where R^2 is a binary higher-order attack relation such that if $(X, (Y, Z))$ and $(X', (Z, Y)) \in R^2$, then $(X, X'), (X', X) \in R$. These relation are represented in Figure 11 where arguments a and b attack each other and arguments c and c' express the preference of a over b and converse, respectively. Thus arguments c and c' attack each other too, since their preferences are incompatible. In Modgil and Bench-Capon [43], the authors show how hierarchical second-order argumentation can be represented in Dung's theory using attack arguments. Moreover, Barringer *et al.* [8] argue that the attack of b to $d \rightarrow c$ can itself be attacked.

Abstract argumentation networks were generalized by Bench-Capon [9], where a colouring, which represents the type of arguments, is added to the network and colours are linearly ordered by strengths. The main rationale behind the introduction of colours consists in modeling the intuition that arguments can be divided into kinds and that some kinds of arguments are more important than others. This kind of approaches extend Dung's standard argumentation framework presenting value-based argumentation frameworks which are defined, for instance, as $\langle A, R, v, val, P \rangle$ where A and R are as usual, v is a non empty set of values, val is a function which maps from elements of A to elements of v and P is the set of possible audiences. An example is provided by Figure 11 from Bench-Capon [9], where a and c would be skeptically acceptable. If, however, we consider the values for the two possible audiences, *red* and *blue*, the following two preferred extensions are obtained: for red, which prefers red to blue, we get $\{a, c\}$ while for blue, which prefers blue to red, we get $\{a, b\}$.

Bipolar argumentation has been introduced by Cayrol and Lagasquie-Schiex [25]. The authors aim in defining support and defeat independently one from the other. An abstract bipolar argumentation framework is an extension of the basic Dung's argumentation framework in which two kinds of interactions between arguments are used, having thus a bipolar representation of the interactions between arguments. At the meta level, they have arguments in favor of other arguments, i.e., the support relation, and also arguments against other arguments, i.e., the defeat relation. An example of bipolar argumentation network is provided in Figure 11.

Toulmin [50] gives in his scheme a representation of the process of defending a particular claim against a challenger. Several challenges arises from this scheme such as the representation of micro arguments and their relationships of defeat and support. Concerning the argument schema proposed by Toulmin [50], Bench-Capon [12] takes the onus of proof to be agreed at the outset, allowed for chaining arguments together so that some data can be the claims of other arguments, and that claims can serve as the data for succeeding arguments, and introduced the notion of presupposition, which is supposed to represent propositions assumed to be true in the context. With this schema, the author argues to have some flexibility in assigning particular roles to premises in an argument.

Another extension of Dung's abstract argumentation framework is introduced by Bochman [14]. This EAF provides a direct representation of global conflicts between sets of arguments. The extension is called collective argumentation and turns out to be suitable for representing semantics of disjunctive logic programs. Collective argumentation theories are shown to possess a four-valued semantics, and are closely related to multiple-conclusion consequence relations. Two special kinds of collective argumentation, positive and negative argumentation, are considered in which the opponents can share their arguments. Negative argumentation turns out to be especially appropriate for analyzing stable sets of arguments. Positive argumentation generalizes certain alternative semantics for logic programs.

One of the main problems with extended argumentation frameworks consists in the adaptation of Dung's semantics. Each of the extended argumentation frameworks presented above defines its own semantics and this increases the complexity of these frameworks and the combination of some them together. This leads to a lack of a universal argumentation theory and a proliferation of specific frameworks which are so specific which cannot be simply used in other contexts. Our meta argumentation methodology is a candidate for such a more general theory.

2.3.2. Applying Dung's theory of abstract argumentation to itself

In the context of Dung's theory of abstract argumentation, we define extended argumentation as an instance of abstract argumentation as follows:

Meta-argumentation is Dung's theory. Argumentation frameworks are not extended but only instantiated.

Meta-arguments "accept(a)" for all arguments a. The set of meta-arguments contains, among others, the meta-argument "argument "a" is accepted" for all arguments in the extended argumentation framework.

Extended argumentation contains Dung's theory as special case. A representation of extended abstract argumentation frameworks contains Dung's theory as a special case. For example, in preference based argumentation Dung's framework is the special case where all arguments are equally preferred, and in multiagent argumentation, Dung's framework is the special case in which there is only one agent.

In this case, meta-argumentation is argumentation. If the set of meta-arguments contains only the representation corresponding to a basic Dung's framework, then the extensions of the meta-argumentation correspond to the extensions of the basic argumentation framework.

2.4. A unified methodology based on acceptance functions

Our methodology of meta-argumentation uses the idea of acceptance functions. They were introduced by Baroni and Giacomin, because they needed them to define principles of argumentation in Dung's theory.

2.4.1. Baroni and Giacomin's formal framework

In this paper we use four ideas from the recently introduced formal framework for the evaluation of extension-based argumentation semantics introduced by Baroni and Giacomin [7]. The first idea we adopt is that the set A represents the set of arguments produced by a reasoner at a given instant of time. Baroni and Giacomin therefore assume that A is finite, independently of the fact that the underlying mechanism of argument generation admits the existence of infinite sets of arguments. Like in Dung's original framework, they consider argumentation framework as a pair $\langle A, \rightarrow \rangle$ where A is a set and $\rightarrow \subseteq (A \times A)$ is a binary relation on A, called attack relation.

Baroni and Giacomin thus observe that the set of all arguments can be generated, which is a second idea which we explore in meta-argumentation. In the following it is useful to explicitly refer to the set of all arguments which can be generated, which we call $\mathcal{U}$ for the universe of arguments.

The third idea we adopt from Baroni and Giacomin is the use of a function $\mathcal{E}$ that maps argumentation frameworks $\langle A, \rightarrow \rangle$ to its set of extensions, i.e., to a set of sets of arguments. Since Baroni and Giacomin do not give a name to the function $\mathcal{E}$, and it maps argumentation frameworks to the set of accepted arguments, we call $\mathcal{E}$ the *acceptance function.*

Definition 5. *Let $\mathcal{U}$ be the universe of arguments. An acceptance function $\mathcal{E} : \mathcal{U} \times 2^{\mathcal{U} \times \mathcal{U}} \rightarrow 2^{2^{\mathcal{U}}}$ is*

1. *a partial function which is defined for each argumentation framework $\langle A, \rightarrow \rangle$ with finite $A \subseteq \mathcal{U}$ and $\rightarrow \subseteq A \times A$, and*

2. *which maps an argumentation framework $\langle A, \rightarrow \rangle$ to sets of subsets of A: $\mathcal{E}(\langle A, \rightarrow \rangle) \subseteq 2^A$.*

The first three principles make the formal framework of Baroni and Giacomin also well suited for the dynamics of argumentation [17, 16], because a single acceptance function can represent the sequence of argumentation frameworks built up during a dialogue, together with the extensions of accepted arguments at each step of the dialogue.

The fourth idea we adopt is the use of argumentation principles. Baroni and Giacomin identify the following two fundamental principles underlying the definition of extension-based semantics in Dung's framework, the *language independent* principle and the *conflict free* principle. See Baroni and Giacomin [7] for a discussion on these principles. Note that the language independence principle cannot be expressed in Dung's theory, since it compares argumentation frameworks, and in Dung's setting, the argumentation framework is supposed to be fixed.

Definition 6 (Language independence). *Two argumentation frameworks $\mathcal{AF}_1 = \langle A_1, \rightarrow_1 \rangle$ and $\mathcal{AF}_2 = \langle A_2, \rightarrow_2 \rangle$ are isomorphic if and only if there is a bijective mapping $m : A_1 \rightarrow A_2$, such that $(\alpha, \beta) \in \rightarrow_1$ if and only if $(m(\alpha), m(\beta)) \in \rightarrow_2$. This is denoted as $\mathcal{AF}_1 \doteq_m \mathcal{AF}_2$.*

A semantics $\mathcal{S}$ satisfies the language independence principle *if and only if $\forall AF_1 = \langle A_1, \rightarrow_1 \rangle$, $\forall AF_2 = \langle A_2, \rightarrow_2 \rangle$ such that $AF_1 \doteq_m AF_2$ then $\mathcal{E}_{\mathcal{S}}(AF_2) = \{M(E) \mid E \in \mathcal{E}_{\mathcal{S}}(AF_1))\}$, where $M(E) = \{\beta \in A_2 \mid \exists \alpha \in E, \beta = m(\alpha)\}$.*

Definition 7 (Conflict free). *Given an argumentation framework $AF = \langle A, \rightarrow \rangle$, a set $S \subseteq A$ is* conflict free, *denoted as $cf(S)$, iff $\not\exists \alpha, \beta \in S$ such that $a \rightarrow \beta$. A semantics $\mathcal{S}$ satisfies the CF principle if and only if $\forall AF, \forall E \in \mathcal{E}_{\mathcal{S}}(AF) E$ is conflict free.*

A principle is a set of argumentation semantics. Reinstatement [23] is also a principle which can be accepted or rejected, and an argumentation framework can be represented by any binary graph, i.e., as in dominance theory. The graph theoretical properties of an argumentation graph are discussed also by Dunne [28]. In this paper the effect of a number of graph-theoretic restrictions is considered: k-partite systems in which the set of arguments may be partitioned into k sets each of which is conflict-free; systems in which the numbers of attacks originating from and made upon any argument are bounded, planar systems and so on. For the class of bipartite graphs, it is shown that determining the acceptability status of a specific argument can be accomplished in polynomial-time under both credulous and skeptical semantics.

Principles describe properties that can be written using a logic of argumentation [15]. Which logic of argumentation is most suited to represent principles is an open problem.

2.4.2. Acceptance functions in meta-argumentation

At first sight it may seem that the Baroni and Giacomin framework is not much different from Dung's framework. However, the use of acceptance functions give us additional expressive power lacking in Dung's framework, and which we explore in the techniques of meta-argumentation in the following section. One example we already mentioned is the fact that reinstatement is no longer built in, but it is a defined property. Another example is the fact that there can be many isomorphic argumentation frameworks, whereas in Dung's framework, isomorphic frameworks cannot be distinguished.

We use the existence of isomorphic argumentation frameworks, by demanding that the function f from extended argumentation frameworks to basic argumentation frameworks can be inverted. It means that f is an injective or one-to-one function, i.e. it is a function which associates distinct extended argumentation frameworks with distinct basic argumentation frameworks, such that every unique extended argumentation framework produces a unique basic argumentation framework. However, we do not require that all basic argumentation frameworks must be mapped, such that the inverse may be a partial function. We do assume that each extended argument is mapped onto a distinct argument, i.e., the inverse is not a multi-valued function.

The acceptance function may encode information about arguments. For example, for an argument, we can identify all the argumentation frameworks in which it occurs, because only for these argumentation frameworks the acceptance function is defined:

$$domain(\mathcal{E}) = \{AF \mid \mathcal{E}(AF) \text{ is defined}\}$$

$$framework(a) = \{\langle A, \rightarrow\rangle \in domain(\mathcal{E}) \mid a \in A\}$$

Then, we can use these definitions to identify arguments which are never attacked by other arguments as those elements for which the function f is well-defined:

$$unattacked = \{a \in \mathcal{U} \mid \forall \langle A, \rightarrow\rangle \in framework(a) \forall b \in A : \neg(b \rightarrow a)\}$$

In principle we could as well have said that distinct extended argumentation frameworks are mapped to the same basic argumentation framework, such that the inverse would be a multi-valued function. However, we believe that the use of standard one-valued functions is conceptually clearer here.

2.4.3. Meta-argumentation methodology

Using acceptance functions, we can make the application of Dung's theory of abstract argumentation to itself more precise. In particular, we further formalize the four steps of defining extended argumentation as an instance of abstract argumentation, as introduced in Section 2.3.2.

Meta-argumentation is Dung's theory. $\mathcal{E}$ is a function from argumentation frameworks to sets of extensions of arguments.

Meta-arguments "accept(a)" for all arguments a. There is a surjective or one-to-one function from the arguments of the extended argumentation framework to the set of meta-arguments.

Extended argumentation contains Dung's theory as special case. There is a case in which f maps the extended argumentation framework to itself.

In this case, meta-argumentation is argumentation. In this case in which the extended argumentation framework is a basic argumentation framework, the functions f and g are bijections.

2.5. Summary

Abstraction is represented using acceptance functions by the language independence assumption: the set of accepted arguments is the same for isomorphic argumentation frameworks, such that they depend only on the attack relation. Instantiation means that we describe the structure of arguments, such that the attack relation is derived from it. Extended argumentation does not directly describe the structure of the arguments, but describes it indirectly by other relations among arguments, such as preferences or higher order attack relations. The meta-argumentation methodology means that arguments in Dung's framework are interpreted as meta-arguments which are mapped to "argument a is accepted" for some argument a.

An apparent distinction between structured arguments and extended argumentation is that the function f may introduce auxiliary arguments, such that an instantiation of a basic Dung framework may lead to less arguments in the extended argumentation framework than in the basic argumentation framework. To explain this phenomenon, we have to discuss the techniques of meta-argumentation in the following section.

3. Meta-argumentation techniques

In this section, we explain three techniques used in meta-argumentation modeling: flattening of extended argumentation frameworks, representation of Dung's basic argumentation frameworks by extended argumentation frameworks, and specification languages for Dung's basic argumentation frameworks. We illustrate these new techniques by preference-based and higher order argumentation.

3.1. The meta-argumentation techniques: informal introduction

The meta-argumentation methodology is based on the idea that we can instantiate Dung's basic argumentation frameworks with extended argumentation frameworks, as discussed in Section 2. The techniques of meta-argumentation show *how* to instantiate basic argumentation frameworks. The first technique to define and study instantiation functions or algorithms is called flattening.

3.1.1. Flattening

Flattening may be seen as the inverse of instantiating a basic argumentation framework with an extended argumentation framework, because a flattening algorithm takes as input an extended argumentation framework, with for example attacks on attack relations or preferences among arguments, and produces as output a basic argumentation framework with attack relations only. Abstractly, flattening is a function f from a set of extended argumentation frameworks to the set of basic argumentation frameworks:

$$f : \mathcal{EAF} \to \mathcal{AF}$$

Such flattening functions or algorithms can be very simple, but they can also be more involved. For example, relatively simple flattening functions can be found in the flattening of preference based argumentation frameworks to basic argumentation frameworks, by defining the attack in the basic argumentation framework as the intersection of the attack and the preference relation of the extended argumentation framework: an argument attacks an argument in basic abstract argumentation when it attacks it in extended abstract argumentation and the attacker is preferred to the attacked. For the same preference based argumentation frameworks also other flattening functions can be defined, an issue we discuss in more detail in Section 3.2.1 of this paper. We call this flattening algorithm simple, because there is no need to introduce auxiliary arguments in the basic argumentation framework: its arguments are precisely the arguments of the extended argumentation framework. However, if we flatten a higher order argumentation framework, then the arguments of the basic argumentation framework contain not only the arguments of the extended argumentation framework, but also auxiliary attack arguments, as we discuss in more detail in Section 3.2.3. We call the arguments which occur both in the extended and basic argumentation framework the *primary arguments*, and we call the remaining auxiliary arguments in the basic argumentation framework the *secondary arguments.*

For a given flattening function, the acceptance function of an extended abstract argumentation theory can be defined using the acceptance function of the basic abstract argumentation theory: an argument of an extended argumentation framework is accepted if and only if it is accepted in the flattened basic argumentation framework. We call this the derived acceptance function for the extended abstract argumentation framework (for the given flattening function).

$$\mathcal{E}(f(EAF))$$

Roughly, we can use flattening functions or algorithms to define instantiations of Dung's argumentation in the following way:

1. Define a set of extended argumentation frameworks, which contains basic argumentation frameworks as special cases. For example, all arguments are equally preferred, there are no higher order attacks, there is only one agent, or the support relation is empty.
2. Define a flattening function or algorithm to flatten the extended argumentation frameworks to basic argumentation frameworks.
3. The set of all flattened argumentation frameworks gives the set of all descriptions of extended argumentation frameworks, together with constraints that hold among them. For example, if there is a description "argument A attacks argument B", then there must also be descriptions "argument A is accepted" and "argument B is accepted".
4. Invert the flattening function, which gives a function from basic argumentation frameworks to extended argumentation frameworks. Each combination of a set of extended argumentation frameworks together with a flattening function gives an instantiation of Dung's abstract argumentation theory.

The main challenge to this approach to define instantiations of Dung's theory using the flattening approach is to make it conceptually more clear.

Any modeling technique crucially depends on the simplicity and intuitiveness of its basic concepts, and the inverse flattening approach as we have discussed it thus far is too abstract to be used effectively. In the above analysis, the confusing point is that we describe arguments by itself. When an extended argumentation framework is flattened, the arguments of the extended argumentation framework are also (primary) arguments of the basic argumentation framework. Though this is done without much problems when extended argumentation theories are flattened, it becomes conceptually more complicated when we instantiate basic argumentation frameworks. It is strange for many modelers to instantiate something with itself.

Meta-argumentation is a way to solve this conceptual confusion. From the perspective of flattening, if an argument a of the extended argumentation framework also occurs in the flattened basic abstract argumentation framework, then we do not call it argument a anymore, but we call it the meta-argument "argument a is accepted." It is confusing if the object and meta-level are identified if we instantiate an abstract argument by the same argument, and thus we solve it by making the abstraction levels explicit.

In other words, when we instantiate abstract arguments, we interpret them as meta-arguments, and then some of the meta-arguments are instantiated by "argument ... is accepted", and some of the meta-arguments are instantiated by other relations among arguments, for example, "... supports ..." or "... attacks ...". More abstractly, there is a complete function that maps arguments in the extended argumentation framework to the basic abstract argumentation framework, and a partial function of abstract arguments to extended arguments.

A technical issue that comes up is the question whether we can distinguish primary and secondary arguments when we instantiate arguments. In other words, if we flatten an extended argumentation framework we introduce auxiliary arguments, then how can we recognize these auxiliary arguments in the basic argumentation framework? As we discuss in Section 3.2.3, in the case of higher order argumentation we can identify auxiliary arguments using the notion of *critical subsets*. The idea is that the labeling value of the auxiliary arguments is determined by the labeling value of the primary arguments [31, 30].

3.1.2. Representation

When an extended argumentation theory instantiates a basic argumentation theory, we say that the basic theory represents the instantiated theory, and that the instantiated theory is represented by the basic theory. In other words, when a set of extended argumentation frameworks is flattened to a set of basic argumentation frameworks, we say that the basic argumentation theory represents the extended argumentation theory, or that the extended argumentation theory is represented by the basic theory.

In many cases, a set of extended argumentation frameworks is represented by all basic argumentation frameworks, and the notion of representation may not seem very useful. For example, we can always instantiate a basic argumentation framework with a preference based argumentation framework, by choosing the same attack relation, and the universal preference relation. In other words, when we flatten a preference based argumentation framework to a basic argumentation framework, there is always a basic argumentation framework to which an extended argumentation framework is flattened, namely the argumentation framework with the same attack relation, and with the universal preference relation.

However, in general, a problem with the flattening technique is that there can be basic argumentation frameworks which cannot be instantiated, because there is no extended argumentation framework that is flattened to it. For example, suppose the domain of a flattening function is the set of extended argumentation frameworks that contain a symmetric attack relation together with a transitive preference relation, and the co-domain is the set of argumentation frameworks in which the attack relation is acyclic [36, 37]. In that case, there is no extended argumentation framework that is flattened to a cyclic argumentation framework, in other words, if we have a cyclic argumentation framework, we cannot instantiate it with an extended argumentation framework. This is a problem, since it means that the instantiation is not defined for a universal domain, but only for some fragments of abstract argumentation. Moreover, there can be abstract argumentation frameworks, for which there are two extended argumentation framework that are mapped to it. In that case, the problem disappears on closer inspection. When building refinements of models, it is common practice that there are several options in which a model can be refined.

$$\{AF \mid \exists EAF \in \mathcal{EAF} : AF = f(EAF)\}$$

If the instantiation is a complete function, i.e. defined for all basic argumentation frameworks, then we can add principles to the attack relation, such that we can define representation results. In our example, when we add the symmetry principle to the preference based argumentation framework, then we have to add the acyclicity principle to the basic argumentation framework. Thus, the principles which we add to the basic and extended argumentation frameworks do not have to be the same! This is not surprising by closer inspection, because it is precisely due to this property that preferences have been added to the symmetric argumentation frameworks, as explained in Section 3.3.

We now encounter our second conceptual problem. When we instantiate a acyclic attack relation by a symmetric one, it becomes confusing. Therefore we prefer not to use the name attack relation in the extended argumentation framework, but rather use a different name. In this particular case, the name "conflict relation" for the extended argumentation framework seems to be better suited. This has been observed before, and others like Prakken [46] have used the name "defeat' for the basic attack relation, and "attack" for the attack relation in the extended argumentation framework with preferences among the arguments. However, we prefer in our meta-argumentation approach to maintain Dung's terminology and reserve "attack" for the attack relation in the basic argumentation framework.

3.1.3. Specification of Dung's basic argumentation frameworks

Specification formalisms are a natural tool used in all areas of modeling. Often the formalisms which are best to do reasoning are less intuitive to be used by humans. There may be several reasons. Sometimes the specification formalisms are based on a visual language like UML or entity relationship diagrams, and the reasoning formalisms are based on description logic or first order logic. In other cases the specification formalisms are more compact than the reasoning formalisms, such as languages to describe multi criteria decision problems.

Extended argumentation frameworks may be seen as specification formalisms, because they may be more compact or more intuitive descriptions of a basic argumentation framework, namely the basic argumentation framework to which they are flattened. For example, a preference based argumentation framework may be seen as a specification of a basic argumentation framework. In other words, an extended argumentation framework may be seen as a specification of a basic argumentation theory, when the basic argumentation theory is represented by the extended theory.

The distinction between representation and specification is a subtle one. Most of the extended argumentation theories may be seen as representations of basic argumentation frameworks, in the sense that flattening algorithms have been defined, but they are also more ambitious than specification formalisms, in the sense that independent acceptance functions for these extended argumentation theories have been defined. Such an independent acceptance function does not make sense if we consider the extended argumentation frameworks as specification formalisms: in that case, the acceptance function of the extended argumentation theory is the derived acceptance function from the flattening function.

As an analogy, consider the representation of the preferences of a rational agent in the foundations of statistics, for example in the representation theorems of Savage [48]. In this theory, the preferences of the agent (as revealed by his actions) are represented by a probability distribution together with a utility function, and the preferences can be computed from these two functions by the expected utility decision rule. In such a case, we can interpret the extended theory of probability and utility as independently motivated, or we can consider them as theoretical constructs to specify the agent's preferences.

Note that a specification formalism is distinct from a logic of argumentation, of which several have been defined recently Boella *et al.* [15]. A logic of argumentation can be best seen as a language to define principles of argumentation, since it has as its models a set of argumentation frameworks. It case be used for argumentation compliance, in the sense that procedures can be defined to check whether a model satisfies a formula, i.e., whether an argumentation framework satisfies a principle.

3.1.4. Scope of the meta-argumentation techniques

In principle, we can also flattening an extended framework to another extended framework, such that we can combine extended argumentation frameworks. Consequently, we can design argumentation theories by starting from Dung's abstract theory and have a sequence of instantiations. In Villata [52], we show how to use meta-argumentation to merge argumentation frameworks, in which a meta-argument ca be instantiated by "agent i knows argument a" and the acceptable arguments reflect the arguments accepted by the multi-agent system. Moreover, we illustrate how a subsumption relation can be defined among arguments, and we show how the Toulmin scheme can be represented using meta-argumentation.

However, we believe that there are also limitations to the approach. On the one hand there are extensions which are more easily defined in another way. E.g., if we introduce audiences [10] in our meta-argumentation theory, then the distinction between objective and subjective acceptance seems more difficult to make. Moreover, if we add negotiation among the agents in a multiagent argumentation theory, then it seems better to use a game theoretic extension of Dung's theory than to model it using meta-argumentation.

3.2. Flattening

The use of meta-arguments can be seen as a particular case of the well known flattening process [39] in logic and algebra. Flattening consists in the *translation* of a specification into an atomic specification with the same meaning. In the flattening process, constructs such as rename and forget lead to some minor problems of a syntactical nature. Flattening has been studied for initial specifications and for deriving so-called normal forms of structured specifications. In our model, we translate an argumentation network into an atomic specification where arguments as substituted by meta-arguments.

3.2.1. Flattening preference based argumentation frameworks

The first step of our approach is to define the set of extended argumentation frameworks. In this section extended argumentation frameworks with besides the attacks also preferences among arguments. Abstractly, in this section the set of extended argumentation frameworks $\mathcal{EAF}$ contains all preference based argumentation frameworks $EAF = \langle A, \rightarrow, \succeq \rangle$ where A is a subset of the universe of arguments, $\rightarrow$ is a binary relation on A, and $\succeq$ is a reflexive relation on A. We consider the case in which the relations satisfy additional principles in Section 3.3.

The second step of our approach is to define flattening algorithms as a function from this set of extended argumentation frameworks to the set of *all* basic argumentation frameworks: $f : \mathcal{EAF} \rightarrow \mathcal{AF}$. The flattening in Definition 8 defines the attack in the basic argumentation framework as the intersection of the attack and the preference relation of the extended argumentation framework: an argument attacks an argument in basic abstract argumentation when it attacks it in extended abstract argumentation and the attacker is preferred to the attacked.

For a given flattening function f, the acceptance function of the extended argumentation theory $\mathcal{E}'$ is defined using the acceptance function of the basic abstract argumentation theory $\mathcal{E}$: an argument of an extended argumentation framework is accepted if and only if it is accepted in the flattened basic argumentation framework. We call $\mathcal{E}'$ the derived acceptance function for the extended abstract argumentation framework (for the given flattening function).

Definition 8. *An extended argumentation framework EAF is a tuple $\langle A, \rightarrow, \succeq \rangle$ where $A \subseteq \mathcal{U}$ is a set of arguments and $\rightarrow \subseteq A \times A$ is a binary relations over A, and $\succeq \subseteq A \times A$ is a binary reflexive relation over A.*

The universe of meta-arguments is $MU = \{accept(a) \mid a \in U\}$ and the *flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments $MA \subseteq MU$ is*

$$\{accept(a) \mid a \in A\}$$

and the attack relation $\longmapsto \subseteq MA \times MA$ is a binary relation on MA such that

$$accept(a) \longmapsto accept(b) \text{ if and only if } a \rightarrow b \text{ and } a \succeq b \text{ and not } b \succeq a$$

i.e., $a \rightarrow b$ and $a \succ b$.

For a set of arguments $B \subseteq MU$, the unflattening function g is given by $g(B) = \{a \mid accept(a) \in B)\}$, and for sets of arguments $AA \subseteq 2^{MU}$, it is given by $g(AA) = \{g(B) \mid B \in AA\}$.

Given an acceptance function $\mathcal{E}$ for basic argumentation, the extensions of accepted arguments of an extended argumentation framework are given by $\mathcal{E}'(EAF) = g(\mathcal{E}(f(EAF)))$ The derived acceptance function $\mathcal{E}'$ of the extended argumentation framework is thus $\{(a, b) \mid f^{-1}(a), g(b)\}$.

For the same preference based argumentation frameworks also other flattening functions can be defined. Definition 9 introduces another way to flatten the extended argumentation framework. In this case there does not seem to be a straightforward reason to prefer one way over the other, but when we add principles the distinction may be more substantial, as we discuss in Section 3.3. Besides a conceptual analysis of which flattening function is better suited for our modelling purposes, there are various ways in which flattening functions can be compared or composed, and we can define rationality properties for the flattening function. We give some properties about flattening functions in Section 3.4.

Definition 9. *Let an extended argumentation framework EAF and the universe of meta-arguments MU be as in Definition 8, and the flattening function f be given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments $MA \subseteq MU$ is again $\{accept(a) \mid a \in A\}$, but the attack relation $\longmapsto \subseteq MA \times MA$ is a binary relation on MA such that*

$$accept(a) \longmapsto accept(b) \text{ if and only if } a \rightarrow b \text{ and not } b \succeq a$$

Moreover, let the unflattening function g and the acceptance function $\mathcal{E}'$ of the extended argumentation framework be as in Definition 8.

The third step of the approach determines the set of all possible arguments in the meta-argumentation framework, and relations among them. In this case, the arguments in the meta-argumentation framework correspond directly to the arguments in the extended argumentation framework, and there are no additional constraints, so this step can be skipped.

3.2.2. Instantiating with preferences among arguments

In the fourth and final step of our approach, we consider the instantiation of a basic argumentation framework as a preference-based argumentation framework. As explained in Section 2, the motivation for such instantiations is that it give a more expressive representation formalism to model examples of argumentation. Instantiating a basic argumentation framework with a preference based argumentation framework goes as follows. Assume that we use extended argumentation framework with a preference relation, and a flattening method where the attack relation of the basic argumentation framework is the intersection of the attack and preference relation of the extended argumentation framework. For each two arguments a and b such that a attacks b, we have to decide for the extended argumentation framework, that either:

1. Argument a attacks argument b, and they are equally preferred, or
2. Argument a attacks argument b, and argument a is preferred to argument b, or
3. Argument a attacks argument b and vice versa, and argument a is preferred to argument b.

Note that our meta-argumentation methodology forces us to distinguish the sets of arguments from the set of meta-arguments. In this simple example, where there is a direct one-to-one mapping from the set of arguments to meta-arguments, this may seem superfluous, but it becomes important in the following sections.

3.2.3. Flattening higher order argumentation frameworks

The first step of our approach is to define the set of extended argumentation frameworks. In this section we consider extended argumentation frameworks with besides the attacks also attacks among attacks. Abstractly, in this section the set of extended argumentation frameworks $\mathcal{EAF}$ contains all second order argumentation frameworks $EAF = \langle A, \rightarrow, \rightarrow^2 \rangle$ where A is a subset of the universe of arguments, $\rightarrow$ is a binary relation on A, and $\rightarrow^2$ is a reflexive and transitive relation on $(A \cup \rightarrow) \times \rightarrow$.

The second step of our approach is to define the flattening function f. The flattening in Definition 10 defines the attack using two auxiliary meta-arguments X and Y. Given an argumentation network with atomic arguments a, we introduce the meta-arguments $Y_{a,b}$ which means that a has attack capability on b, and $X_{a,b}$ which means that a does not have attack capability on b. We use the meta-arguments in the following way. Each attack relation $a \rightarrow b$ is replaced by $accept(a) \longmapsto X_{a,b} \longmapsto Y_{a,b} \longmapsto accept(b)$. We call the arguments a and $accept(a)$ the *primary arguments*, and we call the remaining auxiliary arguments in the basic argumentation framework the *secondary arguments*.

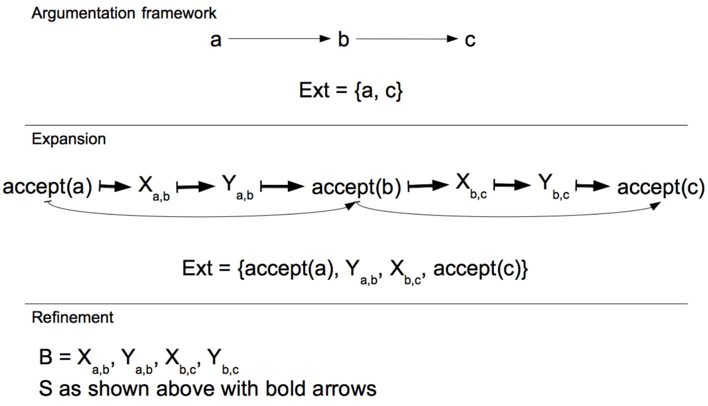

Argumentation framework
a
b
c
Ext = {a, c}
Expansion
accept(a)
X a,b
Y a,b
accept(b)
X b,c
Y b,c
accept(c)
Ext = {accept(a), Y a,b, X b,c, accept(c)}
Refinement
B = X a,b, Y a,b, X b,c, Y b,c
S as shown above with bold arrows

Figure 12. The notions of refinement and abstraction of an AF.

For a given flattening function f, the acceptance function of the preference-based argumentation theory $\mathcal{E}'$ is defined as in Section 3.2.1.

DEFINITION 10. *An extended argumentation framework EAF is a tuple $\langle A, \rightarrow, \rightarrow^2 \rangle$ where $A \subseteq U$ is a set of arguments and $\rightarrow \subseteq A \times A$ is a binary relation over A, and $\rightarrow^2$ is a binary relation on $(A \cup \rightarrow) \times \rightarrow$.*

The universe of meta-arguments is extended with X and Y meta arguments $MU = \{accept(a) \mid a \in U\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in U\}$, and the flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments $MA \subseteq MU$ is

$$\{accept(a) \mid a \in A\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in A\}$$

and $\longmapsto \subseteq MA \times MA$ *is a binary relation on* MA *such that*

$$X_{a,b} \longmapsto Y_{a,b}, Y_{a,b} \longmapsto accept(b)$$

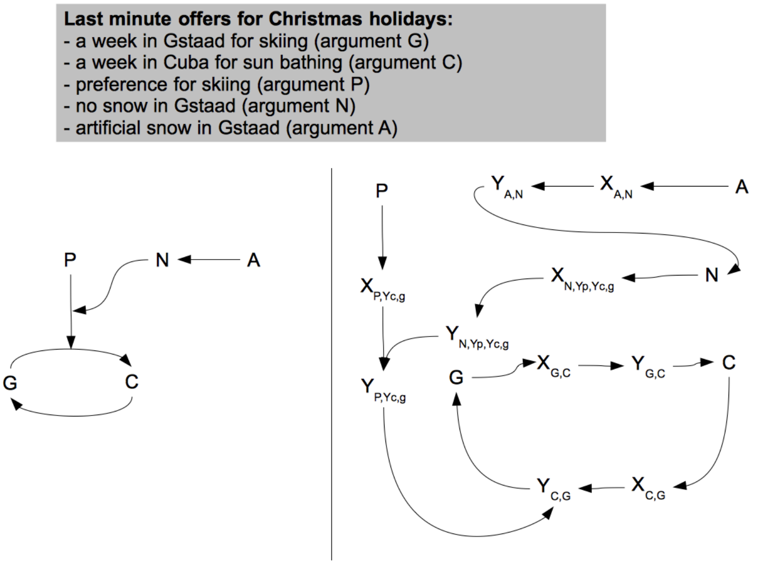
$$accept(a) \longmapsto X_{a,b} \textit{ if and only if } a \rightarrow b$$

$$accept(a) \longmapsto Y_{b,c} \textit{ if and only if } a \rightarrow^2 (b \rightarrow c)$$

$$Y_{a,b} \longmapsto Y_{c,d} \textit{ if and only if } (a \rightarrow b) \rightarrow^2 (c \rightarrow d)$$

The unflattening function g and the acceptance function $\mathcal{E}'$ of the extended argumentation framework are defined as in Definition 8.

Let us consider the example proposed by Baroni *et al.* [6] and represented in Figure 13. In this example, higher-order attacks are considered. In our model, they are represented by means of attacks from the "active" meta-arguments Y which attack the Y meta-arguments of the attacked attack relations.



Last minute offers for Christmas holidays:
- a week in Gstaad for skiing (argument G)
- a week in Cuba for sun bathing (argument C)
- preference for skiing (argument P)
- no snow in Gstaad (argument N)
- artificial snow in Gstaad (argument A)

Figure 13. The representation of the example proposed by Baroni *et al.* [6] in our meta-argumentation model.

Again there are more alternatives to define the flattening. For example, Definition 11 reduces the number of X and Y meta-arguments to the ones we really need.

DEFINITION 11. *Let an extended argumentation framework EAF and the universe of meta-arguments MU be as in Definition 10, and the flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments $MA \subseteq MU$ is*

$$\{accept(a) \mid a \in A\} \cup \{X_{a,b}, Y_{a,b} \mid a \rightarrow b\}$$

and $\longmapsto \subseteq MA \times MA$ is a binary relation on MA such that

$$accept(a) \longmapsto X_{a,b}, X_{a,b} \longmapsto Y_{a,b}, Y_{a,b} \longmapsto accept(b) \text{ if and only if } a \rightarrow b$$

$$accept(a) \longmapsto Y_{b,c} \text{ if and only if } a \rightarrow^2 (b \rightarrow c)$$

$$X_{a,b} \longmapsto Y_{c,d} \text{ if and only if } (a \rightarrow b) \rightarrow^2 (c \rightarrow d)$$

The unflattening function g and the acceptance function $\mathcal{E}'$ of the extended argumentation framework are defined as in Definition 8.

A more general concept is higher order attack. The idea is a straightforward generalization of the notion of second order attack, where now also the second order attacks can attack other attack relations, or be attacked. For the details, see Gabbay [31, 30]. Here we illustrate the use of higher order argumentation to model argumentation by some examples.

The graphical representation of the meta-arguments is presented in Figure 14. The upper part of the figure represents the argumentation network given as input while the lower one is the flattened argumentation network with meta-arguments. Argument a attacks argument b but argument c attacks the attack relation between a and b. We flatten it adding four meta-arguments, two for each attack relation, and meta-arguments $accept(a)$. We compute the following extension, for all argumentation semantics:

$$\{accept(a), accept(c), Y_{c,Y_{a,b}}, accept(b)\}$$

Where meta-arguments $X_{c,Y_{a,b}}$ and $Y_{c,Y_{a,b}}$ represent the attack of argument c to the attack meta-argument represented by $Y_{a,b}$, as shown in Figure 14.

As discussed in Section 2, an attack can itself attack by a higher-order attack another argument, as shown in Figure 15(a). Argument c is attacked by the attack $a \rightarrow b$. This attack is raised by meta-argument $Y_{a,b}$ which is the meta-argument representing the "active" state of the attack $a \rightarrow b$. The extension of this argumentation framework is $\{accept(a)\}$.

Another example is shown in Figure 15(b) where, starting from Figure 15(a), we add a new attack from the new argument d to argument a. This example shows a case in which without meta-arguments it does not make

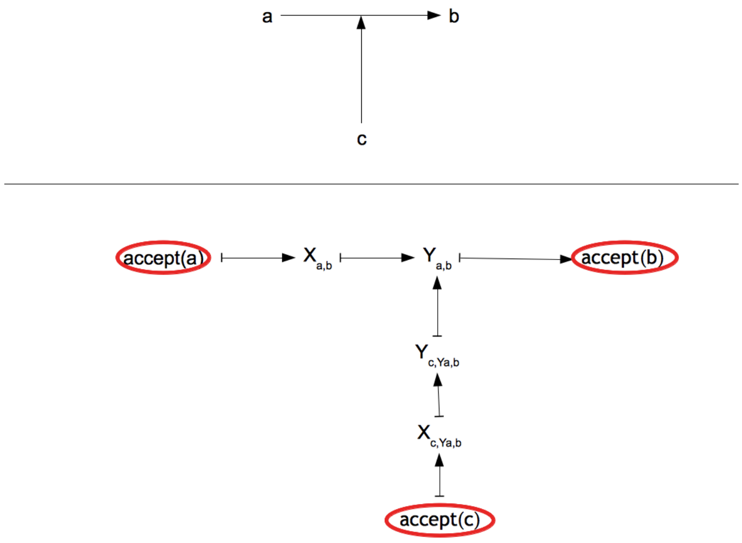


Figure 14. Graphical representation of the extended argumentation network and the flattened one.

sense. The attack of d is translated in the object level in an attack of d to the two meta-arguments representing its attack on $accept(a)$, $X_{d,a}$ and $Y_{d,a}$. The extension of this example is as follows: $\{accept(d), Y_{d,a}, X_{a,b}, accept(b)$, $accept(c)\}$ since the attack $a \rightarrow b$, represented by $Y_{a,b}$, is not in the extension being $accept(a)$ not in the extension too.

Figure 16 represents another example of translation from an argumentation network to the flattened one. The represented case consists in an attack between two arguments a and b and another attack from the attack $a \rightarrow b$ to argument c. The flattened version represents the attack of the attack as an attack from meta-argument $Y_{a,b}$ to argument $accept(c)$. The computation of the extension for the flattened argumentation network is as follows: $\{accept(a), Y_{a,b}\}$.

Finally a more complex argumentation network is presented in Figure 17. This argumentation network depicts argument a which attacks argument b and this attack is attacked by argument c. The attack from argument c to $a \rightarrow b$ attacks also argument b. This argumentation network is flattened in Figure 17(b). The extended argumentation framework has the following extension: $\{accept(c), accept(a)\}$.

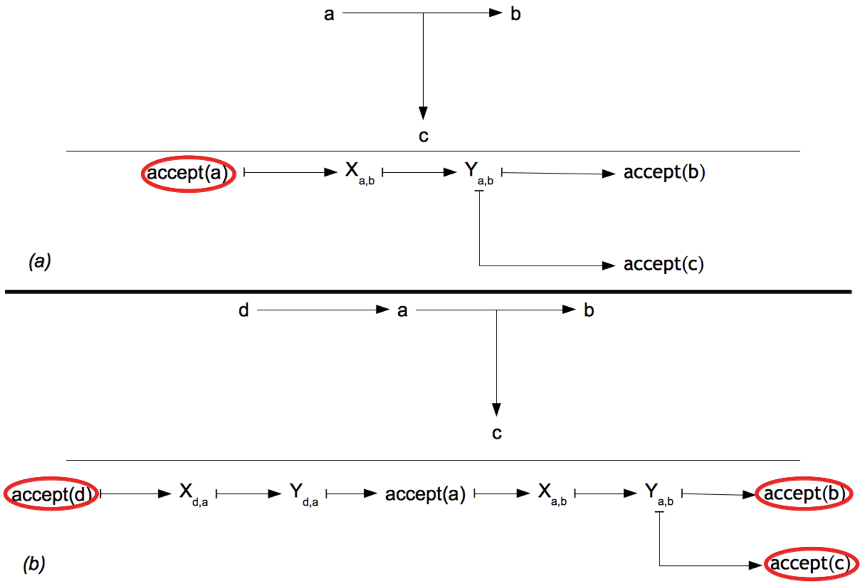


Figure 15. Two examples of higher-order attack in the flattened argumentation network.

In order to give a procedural way of building the meta-argumentation network from a complex argumentation framework obtaining an abstract Dung's based argumentation framework, we define a flattening algorithm. The algorithm works as follows.

The algorithm uses three main functions: function $add()$ adds new arguments to the flattened argumentation framework under the form of refinement $[\mathcal{B}, \mathcal{S}]$ of the starting argumentation framework, function $newAttack()$ adds a new attack relation to the refinement $[\mathcal{B}, \mathcal{S}]$ of the argumentation framework and $findAcc()$ returns the Y meta-arguments of the given attack relation. Algorithm FLATTENING_ALGORITHM is composed by four fundamental steps: the first one consists in flattening the attack relations between arguments of the starting argumentation framework, the second one consists in flattening the attacks from an argument to another attack, the third one considers the attacks from an attack to an argument and, finally, the fourth one consists in flattening the attacks from attack relations to attack relations.

The set of all flattened argumentation frameworks gives the set of all descriptions of extended argumentation frameworks, together with constraints that hold among them. For example, if there is a description "argument a

```
Input: An argumentation network $\langle A, R \rangle$.
Output: A flattened argumentation network $\langle N \cup A, E \rangle$
forall $a \times b \in R$ with $a, b \in A$ do
    $add(X_{a,b}, Y_{a,b})$;
    $newAttack(accept(a), X_{a,b})$;
    $newAttack(X_{a,b}, Y_{a,b})$;
    $newAttack(Y_{a,b}, accept(b))$;
end
forall $a \times y \in R$ with $a \in A$ and $y \in R$ do
    $y_{acc} = findAcc(y)$;
    $add(X_{accept(a),y_{acc}}, Y_{a,y_{acc}})$;
    $newAttack(accept(a), X_{a,y_{acc}})$;
    $newAttack(X_{a,y_{acc}}, Y_{a,y_{acc}})$;
    $newAttack(Y_{a,y_{acc}}, y_{acc})$;
end
forall $a \times b \in R$ with $a \in R$ and $b \in A$ do
    $a_{acc} = findAcc(a)$;
    $newAttack(a_{acc}, X_{a_{acc},b})$;
    $newAttack(X_{a_{acc},b}, Y_{a_{acc},b})$;
    $newAttack(Y_{a_{acc},b}, b)$;
end
forall $a \times b \in R$ with $a, b \in R$ do
    $a_{acc} = findAcc(a)$;
    $b_{acc} = findAcc(b)$;
    $newAttack(a_{acc}, X_{a_{acc},b_{acc}})$;
    $newAttack(X_{a_{acc},b_{acc}}, Y_{a_{acc},b_{acc}})$;
    $newAttack(Y_{a_{acc},b_{acc}}, b_{acc})$;
end
```

Algorithm 1: FLATTENING_ALGORITHM

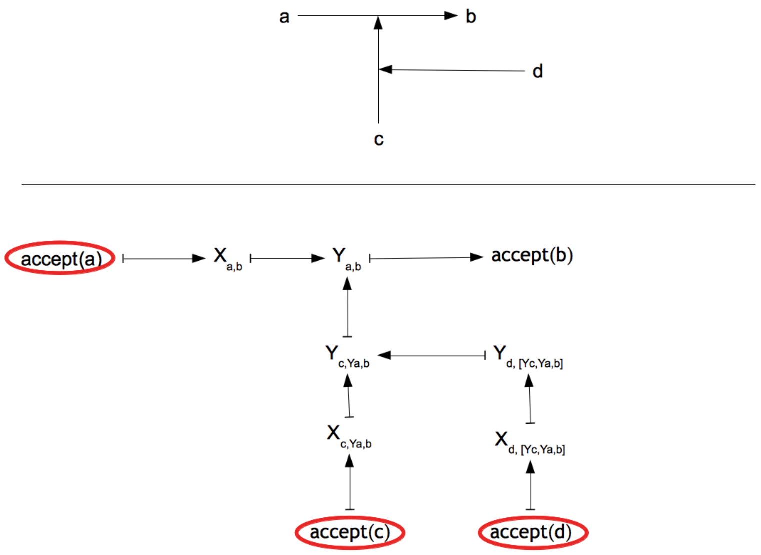
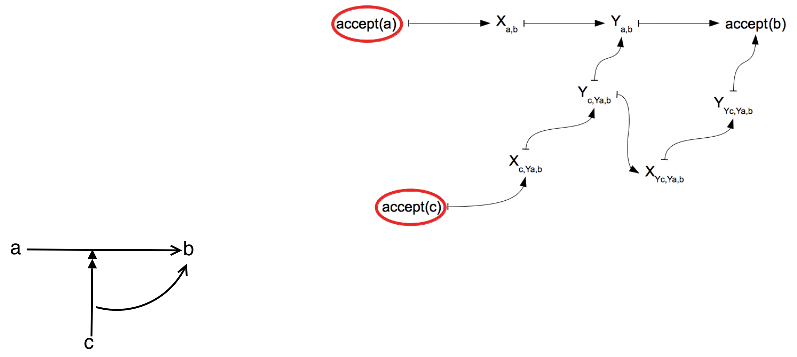


Figure 16. Example of higher-order attacks between four arguments.

attacks argument b", then there must also be descriptions "argument A is accepted" and "argument B is accepted" and the constraints represented by the attacks between meta-arguments $X_{a,b}$ and $Y_{a,b}$. This means to define a set of basic argument types, together with a number of constraints on this set of basic arguments and the attack relations between them. For example, if there are attack arguments, then there can be only attack arguments from basic arguments, or also from attack arguments. We constraint that, having an attack from a to b and the descriptions "argument a is accepted" and "argument b is accepted" and $X_{a,b}$, $Y_{a,b}$, argument $accept(a)$ [1] must attack argument $X_{a,b}$ which must attack argument $Y_{a,b}$ which, finally, must attack argument "argument b is accepted".

The third step of the approach determines the set of all possible arguments in the meta-argumentation framework, and relations among them. In the case of Definition 10, the universe of meta-arguments is extended with X and Y meta arguments $MU = \{accept(a) \mid a \in U\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in U\}$, and the attack relation is characterized by $\longmapsto \subseteq MA \times MA$ is a binary relation on MA such that $X_{a,b} \longmapsto Y_{a,b}, Y_{a,b} \longmapsto accept(b)$. For example, if there

[1] Using the short notation for "argument a is accepted".



(a) Meta-level (b) Object level

Figure 17. An argumentation network in the meta level (a) and object level (b).

is a meta-argument $X_{a,b}$ if and only if there is a meta-argument $Y_{a,b}$. For the flattening function in Definition 11, we have that $X_{a,b}$ implies $accept(a) \in A$ and $accept(b) \in A$, but not vice versa.

3.2.4. Instantiating abstract arguments

In the fourth and final step of our approach, we consider the instantiation of a basic argumentation framework as a higher order argumentation framework. Instantiating a basic argumentation framework with a second order argumentation framework goes as follows. For each two arguments a and b such that a attacks b, we have to decide for the extended argumentation framework, that either:

1. Argument a attacks argument b, and this attack is not attacked itself, or
2. Argument a attacks argument b, and the attack is attacked by an argument which is itself not attacked, or
3. Argument a attacks argument b and vice versa, and the attack of argument b to argument a is attacked by another argument or attack which is accepted.

We can recognize auxiliary or secondary arguments like the X and Y arguments by the acceptance function. For example, in the flattening function of Definition 11, and argument $X_{a,b}$ is accepted if the argument $accept(a)$

is not accepted, and $Y_{a,b}$ is accepted if the argument *accept*(a) is accepted too. In general, the auxiliary arguments are not part of the critical set, see Gabbay [31, 30].

3.3. Representation

The meta-argumentation techniques become more interesting when the argumentation framework satisfy some principles. The following definitions and results for preference based argumentation are taken from Kaci *et al.* [36, 37], and they show that if the attack relation in the extended argumentation framework is symmetric, and the preference relation is transitive, then the attack relation of the flattened argumentation framework is acyclic. Moreover, they show that the two flattening functions of Definition 8 and Definition 9 give rise to two distinct acyclicity or loop principles. To distinguish the attack relation in the extended argumentation framework from the attack relation in the basic argumentation framework, we call the former an incompatibility relation.

Definition 12 (Incompatibility+preference argumentation framework [37]). *An incompatibility+preference argumentation framework is a triplet* $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ *where* $\mathcal{A}$ *is a set of arguments,* $\mathcal{C}$ *is a symmetric binary incompatibility relation on* $\mathcal{A} \times \mathcal{A}$*, and* $\succeq$ *is a preference relation on* $\mathcal{A} \times \mathcal{A}$*.*

Definition 13 ([37]). *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an argumentation framework and* $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ *an incompatibility+preference argumentation framework. We say that* $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ *represents* $\langle \mathcal{A}, \mathcal{R} \rangle$ *iff for all arguments* A *and* B *of* $\mathcal{A}$*, we have* $A \mathcal{R} B$ *iff* $A \mathcal{C} B$ *and not* $B \succ A$*. We say also that* $\mathcal{R}$ *is represented by* $\mathcal{C}$ *and* $\succeq$*.*

Definition 14 (Acyclic argumentation framework [36]). *An argument* A *strictly attacks* B *if* A *attacks* B *and* B *does not attack* A*. A strict acyclic argumentation framework is an argumentation framework* $\langle \mathcal{A}, \mathcal{R} \rangle$ *in which there is no sequence of arguments* $\langle A_1, \ldots, A_n \rangle$ *such that* A_1 *strictly attacks* A_2*,* A_2 *strictly attacks* A_3*, ...,* A_{n-1} *strictly attacks* A_n*, and* A_n *attacks* A_1*.*

Summarizing, strictly acyclic argumentation frameworks are characterized by incompatibility+preference argumentation frameworks.

Theorem 1 ([37]). $\langle \mathcal{A}, \mathcal{R} \rangle$ *is a strictly acyclic argumentation framework (in the sense of Definition 14) if and only if there is an incompatibility+preference argumentation framework* $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ *that represents it (in the sense of Definition 13).*

Definition 15 ([36]). *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework and $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ a conflict+preference argumentation framework. We say that $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ represents $\langle \mathcal{A}, \mathcal{R} \rangle$ iff for all arguments A and B of $\mathcal{A}$, we have $A \mathcal{R} B$ iff $A \mathcal{C} B$ and $A \succeq B$. We also say that $\mathcal{R}$ is represented by $\mathcal{C}$ and $\succeq$.*

Definition 16 (Acyclic argumentation framework). *An acyclic argumentation framework is an argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ in which the attack relation $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ satisfies the following property:*

> *If there is a set of attacks $A_1 \mathcal{R} A_2$, $A_2 \mathcal{R} A_3$, $\cdots$, $A_n \mathcal{R} A_1$ then we have that $A_2 \mathcal{R} A_1$, $A_3 \mathcal{R} A_2$, $\cdots$, $A_1 \mathcal{R} A_n$.*

Summarizing, acyclic argumentation frameworks are characterized by conflict+preference argumentation frameworks.

Theorem 2 ([37]). *$\langle \mathcal{A}, \mathcal{R} \rangle$ is an acyclic argumentation framework if and only if there is a conflict+preference argumentation framework $\langle \mathcal{A}, \mathcal{C}, \succeq \rangle$ that represents it.*

See the original papers by Kaci *et al.* [36, 37] for further details and discussions. What is important for the meta-argumentation techniques is that principles on extended argumentation frameworks give rise to other principles for the basic argumentation framework. Therefore, if we instantiate Dung's argumentation theory with a preference based argumentation theory with a symmetric attack relation, the above results give us a criterium to decide among the two flattening functions in Definition 8 and 9. The choice depends on which kind of cycles we want to be able to model in the argumentation frameworks.

3.4. Specification formalisms

There exists another way of using the mappings from the extended representation, as shorthand notation for representing the argumentation framework. What we need at this point is a set of requirements which we have to satisfy in order to develop a flattening algorithm for this shorthand notation. The requirement of Modgil [41], and of Baroni and Giacomin [7], is to define an argumentation theory for the higher order case, and then to show that the flattened argumentation framework corresponds to the higher order one. But the thing is that this approach just seems to transfer the problem. The question what are the reasons to accept the higher order theory? For an extended discussion about the semantics for higher level attacks, see Gabbay [30].

We propose to find new requirements which have to be satisfied by the flattening algorithm. Some examples of such requirements are listed below. A first requirement of the flattening algorithm is the kind of inputs the algorithm accepts, i.e., the kind of higher order structures which can be flattened. For example, the algorithm allows for flattening attacks attacking attacks (Baroni *et al.*[6] do not, in their approach only arguments can attack attacks), and so on. The minimal higher order structures which must be flattened are given by the Argumentation Framework with Recursive Attacks of [6].

For this knowledge representation language, there are at least three possible solutions:

- the Baroni *et al.*[6] flattening, which considers only $Y_{a,b}$ arguments;
- the Boella *et al.*[32] flattening, which uses only X_a meta-arguments instead of $X_{a,b}$;
- the flattening proposed in this paper, which uses both $X_{a,b}$ and $Y_{a,b}$ meta-arguments.

A second requirement is that the argumentation framework output has to contain at least the arguments of the input. A third requirement is that if the argumentation framework is already flattened, then the flattening algorithm returns the original framework. A weaker variant of the third requirement is that if the original argumentation framework is already flattened, then the extensions of this framework are the same as the extensions of the flattened argumentation framework given by the algorithm. Maybe more precisely, this should hold if we filter out the atomic arguments. For example, if we have arguments a and b, and $a \rightarrow b$, then the flattened argumentation framework is $\{a, X_{a,b}, Y_{a,b}, b\}$ with $a \rightarrow X_{a,b}, X_{a,b} \rightarrow Y_{a,b}, Y_{a,b} \rightarrow b$. The extension of the first argumentation framework is $\{a\}$ while the extension of the second one is $\{a, Y_{a,b}\}$. This weak constraint does not hold, unless some constraints on the *semantics* are imposed. For example, consider again the argumentation framework $\{a, X_{a,b}, Y_{a,b}, b\}$ with $a \rightarrow X_{a,b}, X_{a,b} \rightarrow Y_{a,b}, Y_{a,b} \rightarrow b$. Suppose there is a semantics which outputs arguments $\{a, b\}$ from such a framework, then clearly the constraint is violated.

A fourth requirement is on the output. The output must be a Dung style argumentation framework, but it seems that none of the above flattenings returns precisely a Dung style argumentation framework. In particular, the problem consists in the names given to the arguments in the flattened framework. We could simply define the output to be such that the names are filtered out, but then we do not know what the extension is, because we need to filter the atomic arguments from the output.

An fifth requirement is that the flattening algorithm should be reversible. Thus, given a flattened argumentation framework, we can somehow recover the original higher order argumentation framework. A sixth requirement, which is very important, is on the compositionality of the flattening algorithm. E.g., if we add an attack or an argument, then we only have to flatten this additional attack or argument. A seventh requirement is on the complexity of the algorithm since a compositional algorithm should have low complexity.

A final requirement could be based on the dynamic properties, see for example Boella *et al.* [17, 16].

3.5. Summary

The discussion on the techniques of meta-argumentation highlighted several guidelines for meta-argumentation modeling.

First, instead of instantiating arguments by themselves, we distinguish argument and meta-arguments. From the perspective of flattening, if an argument a of the extended argumentation framework also occurs in the flattened basic abstract argumentation framework, then we do not call it argument a anymore, but we call it the meta-argument “argument a is accepted.” In other words, when we instantiate abstract arguments, we interpret them as meta-arguments, and then some of the meta-arguments are instantiated by “argument ... is accepted”, and some of the meta-arguments are instantiated by other relations among arguments, for example, “... supports ...” or “... attacks ...”. Such auxiliary arguments can be identified in the acceptance function, because they do not belong to a critical set.

Second, if both the basic and the extended argumentation framework contain an attack relation, but they satisfy distinct principles, as can be shown by representation theorems, then we choose another name for the attack relation in the extended argumentation framework. In the particular case of preference based argumentation, the name “incompatibility relation” for the extended argumentation framework seems to be better suited.

Third, abstract properties of the flattening functions are to be defined. If extended argumentation frameworks are used as specifications for basic argumentation frameworks, then the used extensions and flattening functions have to be motivated independently.

4. Related work

In this paper we introduce the methodology of meta-argumentation to model argumentation itself. Bondarenko *et al.* [21] and Verheij [51] may be seen as predecessors of the meta-argumentation approach.

In some way, Dung and colleagues [21] propose already to *instantiate* his theory rather than to extend it, and abstract arguments have been instantiated by, for example, assumptions, default rules, or clauses from a logic program. One of the main reasons for the popularity of Dung is that such so-called extensions can also be modeled as instances of Dung's framework. However, Dung's framework is seen as an abstract reference model into which less abstract models can be mapped, but is not meant to be the "starting point" of a modeling activity. Bondarenko *et al.* [21] refers to Dung's framework as an abstraction of logic programming semantics interpretation, and the assumption-based approach proposed is not introduced as an instantiation of Dung's framework but rather as a sort of intermediate abstraction with respect to various non-monotonic logics.

Verheij [51] presents the argument assistance system, DEFLOG, which can be used to keep track of diverging positions and assist in the evaluation of opinions, in the research area of the dialogical theories of reasoning. The first consideration towards DEFLOG's logical language is the recognition of the warrants of argument steps as logically compound sentences. Since warrants connect two statements, they can be expressed in a logical style using binary connectives. On the one hand, the warrant of a supporting step in which the statement that j is a reason for the statement that y, is denoted using a binary connective, $\leadsto$. On the other hand, the warrant of an attacking step in which the statement that j is a counterargument to the statement that y is denoted using the combination of the binary connective and a unary connective. The defeat of a statement is expressed using the unary connective $\times$. A sentence $\times j$ expresses that the statement that j is defeated. As a result, it becomes possible to define attack in terms of conditional justification and defeat: the statement that $j \rightarrow y$ can be defined as the statement that if j is justified, then y is defeated, it is expressed by $j \leadsto \times y$.

Meta-argumentation has been treated in an explicit way in the following works. Jakobovitz and Vermeir [34] show how to associate to an argumentation framework its so-called meta-argumentation framework in which meta-arguments represent labelings of the original framework. It turns out that the minimal semantics of the meta-framework characterizes the robust sets of the original framework, thus providing a simple procedure to compute robust sets. They defines a meta-argumentation framework as the tuple $\langle A^*, \leadsto^* \rangle$ where AF^* is the set of restricted labeling of AF such that $A^* = \{$ l such that l is a labeling of AF $|_S$ for some $S \subseteq A\}$ and $l' \leadsto^* l$ iff l' is an incompatible extension of l. All of the labelings and restricted labelings of AF, together with their attacks, are represented in the meta-argumentation framework.

Extending an argumentation framework with the support relation has been done by Cayrol and Lagasquie-Schiex [25] and Amgoud *et al.* [4] using meta-argumentation. The authors aim in defining support and defeat independently one from the other and they introduce an extension of an argumentation framework called bipolar argumentation framework. An abstract bipolar argumentation framework is an extension of the basic Dung's argumentation framework in which two kinds of interactions between arguments are used, having thus a bipolar representation of the interactions between arguments. At the meta level, they have arguments in favor of other arguments, i.e., the support relation, and also arguments against other arguments, i.e., the defeat relation.

A work which discusses another way of doing flattening of argumentation frameworks is presented by Gabbay [31, 30]. The author shows how to substitute one argumentation network as a node in another argumentation network, providing the notion of higher level networks. Substitution is treated as a purely logical operation. Given a network (S, R) with a node $x \in S$, Gabbay sees it as a variable for which we can substitute values. There are two immediate problems: give meaning to the substitution and generalize the notion of the network so that it is closed under substitution. Higher-level networks are networks with conjunctive and disjunctive attacks. The author introduces a new kind of Caminada [23] labelling thinking in terms of labels as functions and giving values to the nodes in some algebraic or numerical range (e.g., complex or real numbers). These equations are solved thanks to the addition of variables not present in the argumentation network. This work and our one are both concerned with the notions of abstraction and instantiation. In Gabbay [31, 30], an argumentation network could be abstracted and seen as a single node of another argumentation network and then the node is instantiated with all the nodes and attack relations of the networks which represent its refinement. Fibring seems more general than meta argumentation since the same argument can occur in the substituted network as well as in the original one, e.g. if we have $x \rightarrow a \rightarrow y$, and we replace a by $c \rightarrow x$. However, in our approach, we also can have the same arguments at distinct abstraction levels. The applied methods are different. While Gabbay [31, 30] uses collective arguments, we use meta argumentation producing from the original, complex argumentation network a new network in which it is simpler to compute the labelling. The two flattening approaches seem to suggest, i.e., in the section eliminating joint and disjunctive attacks, that the fibring approach can be reduced to a meta argumentation approach.

An approach to meta-argumentation is provided also by Wooldridge *et al.* [53]. The starting point of this work is the same of our one and consists in the view that arguments and dialogues are inherently meta-logical processes. The authors argue that rational argumentation also involves putting forward arguments about arguments, and it is in this sense that they are meta-logical. For example, a statement that serves as a justification of an argument is a statement about an argument: the argument for which the justification serves must itself be referred to in the justification. They construct a well-founded tower of arguments, where arguments, statements, and positions at a level n in the hierarchy may refer to arguments and statements at levels m, for $0 \leq m < n$. In the bottom of the hierarchy there are object level statements about the domain of discourse. The presented hierarchical first-order meta-logic is a type of first-order logic in which individual terms in the logic can refer to terms in another language. This formalization enables to give a clean formal separation between object-level statements, arguments made about these object level statements, and statements about arguments. Similarly as our approach, the authors argue that any proper formal treatment of logic-based argumentation must be a meta-logical system. This is because formal arguments and dialogues do not just involve asserting the truth or falsity of statements about some domain of discourse: they involve making arguments about arguments, and potentially higher-level references (i.e., arguments about arguments about arguments). The main difference in comparison with our approach consists in the modeling perspective by which we present and discuss meta-argumentation, without developing a new meta-logic.

Modgil and Bench-Capon [43] show how hierarchical second-order argumentation can be represented in Dung's theory using attack arguments. The authors present an extension of Dung's argumentation framework enabling the integration of meta-level reasoning about which arguments should be preferred. The extended argumentation framework introduced by them is similar to our one since they introduce meta-arguments for preferences which can be compared to our X and Y meta-arguments. They show how meta-level argumentation about values can be captured by the extended argumentation frameworks they defined showing also that these extended argumentation frameworks can be rewritten as Dung argumentation frameworks. In particular, they used a hierarchical approach with three levels such that binary attacks are between arguments within a given level, and defence attacks originate from arguments in the immediate meta-level. In the case of attacks such as $a \to b$ they add two intermediate meta-arguments which operate like our X and Y meta-arguments but they do not use meta-arguments like "argument a is accepted".

Baroni *et al.* [6] investigate the generalization the argumentation framework notion of attack by allowing an attack, starting from an argument, to be directed not just towards an argument but also towards any other attack. This is be achieved by a recursive definition of the attack relation leading to the introduction and preliminary investigation of a formalism called argumentation framework with recursive attacks.

Second and higher order argumentation have been discussed in a modeling approach to argumentation by Boella *et al.* [20]. In this work, a new way to analyze cooperation using argumentation networks is presented. The authors introduce different modelling decisions which can be adopted by the coalitions, represented as arguments, in order to be formed and to survive to the attacks of the other coalitions. In [20], the idea is that first and second order attacks do not depend directly on the coalitions, in the sense that a coalition cannot invent them if they are not already available for it. Concerning second order attacks, the coalition can decide to attack or not, but it can only decide to attack if there is this possibility of attack. This choice is modeled considering the following two alternatives: removing the second order attack from the argumentation framework or adding a higher order attack for representing that the coalition decides to not attack. The first solution presents a problem, particularly in iterative design, since, in this case, it is necessary to refine different argumentation frameworks, due to the removal of the second order attack which means also the removal of the dynamic dependency underlying it. The authors adopt the second alternative, introducing higher-order attacks to model the choice not to attack at the coalition level of the iterative design process, without having to change the level below. In fact, the dynamic dependency still exists if the coalition either chooses not to attack (i.e., adding a higher order attack) or to attack at the higher level (i.e., not adding an higher order attack).

5. Conclusions

In this paper we introduce the meta-argumentation viewpoint on argumentation, which conceptualizes argumentation together with arguing about argumentation. Our meta-argumentation viewpoint assumes that meta-argumentation has to be able to mirror argumentation, for example, lawyers should be able to mirror the argumentation of suspects, and political commentators should be able to mirror the argumentation of politicians. Moreover, our meta-argumentation viewpoint assumes that the common pattern in argumentation and meta-argumentation is conflicts resolution, and that the relation of argumentation and meta-argumentation is argument instantiation, which both can be modeled using Dung's theory of abstract argumentation. In meta-argumentation, arguments of Dung's framework are interpreted as meta-arguments which are mapped to "argument a is accepted" for some argument a.

We show how to use meta-argumentation as a general methodology for modeling argumentation. Our meta-argumentation methodology is a way to use Dung's argumentation theory by guiding how it can be instantiated with extended argumentation theories. We need some more general concepts than introduced by Dung, for which we use the Baroni and Giacomin framework [7] of – what we call – acceptance functions and argumentation principles. In this framework, abstraction is represented by the notion of isomorphic argumentation frameworks and the language independence assumption. This assumption says that the set of accepted arguments is the same for isomorphic argumentation frameworks, such that they depend only on the attack relation. Therefore we can define the flattening of the acceptance function of an extended argumentation theory to Dung's acceptance functions as a bijection, such that we can use the inverse function as the instantiation of Dung's theory.

The technique of meta-argumentation applies Dung's theory of abstract argumentation to itself, by instantiating Dung's abstract arguments with meta-arguments using the flattening techniques. Such auxiliary arguments can be identified in the acceptance function, because they do not belong to a critical set. Representation techniques are used to show that the attack relation of the basic and the extended argumentation framework may satisfy distinct principles, and therefore we choose another name for the attack relation in the extended argumentation framework, for example "incompatibility relation" for the preference based argumentation framework. Extended argumentation frameworks are used as specifications for basic argumentation frameworks, in the sense that they are a way to model argumentation. The used extended argumentation frameworks and flattening functions therefore have to be motivated independently from a modeling perspective, for which we define abstract properties of the flattening functions.

There are various topics for further research. A first topic for further research is a study of the relation between fibring argumentation frameworks and meta-argumentation, where the former instantiates abstract arguments with other argumentation frameworks, and the latter instantiates meta-arguments. Despite their apparent differences, they use similar techniques, in particular flattening functions. Such a comparison could lead to a more general formal framework for formal argumentation, which has fibring and meta-argumentation as special cases. This could incorporate not only flattening, representation and specification techniques discussed in this paper, but it would incorporate also other new ideas in formal argumentation like logics of argumentation and dynamic approaches to argumentation.

A second topic for further research is the use meta-arguments. For the X and Y meta-arguments discussed in this paper, we can distinguish two modeling challenges. First, if we like to model something, then when do we introduce attacks among these X and Y meta-arguments? Second, if we have a meta-argumentation framework with X and Y meta-arguments, then how can or should we read the attacks among these meta-arguments? These questions are addressed in Villata [52] for merging argumentation frameworks in multiagent argumentation, subsumption relations in bipolar argumentation, and combining micro-arguments using Toulmin's scheme.

References

[1] Leila Amgoud. An argumentation-based model for reasoning about coalition structures. In Parsons et al. [44], pages 217–228.

[2] Leila Amgoud and Philippe Besnard. Bridging the gap between abstract argumentation systems and logic. In Lluis Godo and Andrea Pugliese, editors, *SUM*, volume 5785 of *Lecture Notes in Computer Science*, pages 12–27. Springer, 2009.

[3] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Ann. Math. Artif. Intell.*, 34(1-3):197–215, 2002.

[4] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and P. Livet. On bipolarity in argumentation frameworks. *Int. J. Intell. Syst.*, 23(10):1062–1093, 2008.

[5] Katie Atkinson and Trevor J. M. Bench-Capon. Legal case-based reasoning as practical reasoning. *Artif. Intell. Law*, 13(1):93–131, 2005.

[6] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Encompassing attacks to attacks in abstract argumentation frameworks. In Sossai and Chemello [49], pages 83–94.

[7] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

[8] Howard Barringer, Dov M. Gabbay, and John Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning*, volume 2605 of *Lecture Notes in Computer Science*, pages 59–98. Springer, 2005.

[9] T.J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Logic and Computation*, 13(3):429–448, 2003.

[10] Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In Salem Benferhat and Enrico Giunchiglia, editors, *NMR*, pages 443–454, 2002.

[11] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[12] Trevor J.M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *JURIX*, pages 5–20, 1998.

[13] A. Bochman. *Explanatory Nonmonotonic Reasoning.* World Scientific Publishing, 2005.

[14] Alexander Bochman. Collective argumentation and disjunctive logic programming. *J. Log. Comput.*, 13(3):405–428, 2003.

[15] Guido Boella, Joris Hulstijn, and Leendert W. N. van der Torre. A logic of abstract argumentation. In Parsons et al. [44], pages 29–41.

[16] Guido Boella, Souhila Kaci, and Leendert van der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In Sossai and Chemello [49], pages 107–118.

[17] Guido Boella, Souhila Kaci, and Leendert van der Torre. Dynamics in argumentation with single extensions: attack refinement and the grounded extension. In Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman, editors, *AAMAS (2)*, pages 1213–1214. IFAAMAS, 2009.

[18] Guido Boella, Leendert van der Torre, and Serena Villata. Attack relations among dynamic coalitions. In *BNAIC 2008*, pages 25–32, 2008.

[19] Guido Boella, Leendert van der Torre, and Serena Villata. Social viewpoints for arguing about coalitions. In The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors, *PRIMA*, volume 5357 of *Lecture Notes in Computer Science*, pages 66–77. Springer, 2008.

[20] Guido Boella, Leendert van der Torre, and Serena Villata. Analyzing cooperation in iterative social network design. *Journal of Universal Computer Science. (To appear)*, 2009.

[21] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.*, 93:63–101, 1997.

[22] Felix Brandt and Paul Harrenstein. Characterization of dominance relations in finite coalitional games. *Theory and Decision*, To appear, 2009.

[23] Martin Caminada. On the issue of reinstatement in argumentation. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *JELIA*, volume 4160 of *Lecture Notes in Computer Science*, pages 111–123. Springer, 2006.

[24] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[25] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of

arguments in bipolar argumentation frameworks. In Lluis Godo, editor, *ECSQARU*, volume 3571 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.

[26] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.

[27] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[28] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[29] Gabbay, Johnson, Ohlbach, and Woods, editors. *Handbook of the logic of argument and inference: the turn towards the practical.* Elsevier Science, 2002.

[30] Dov Gabbay. Semantics for higher level attacks in extended argumentation frames. part 1: overview. *Studia Logica. (This issue)*, 2009.

[31] Dov M. Gabbay. Fibring argumentation frames. *Studia Logica (This issue)*, 2009.

[32] Serena Villata Guido Boella, Leendert van der Torre. On the acceptability of meta-arguments. In *IAT*, 2009.

[33] Sven Ove Hansson. Preference logic. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, pages 319–387. Kluwer Academic Publishers, 2001.

[34] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9(2):215–261, 1999.

[35] Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *Int. J. Approx. Reasoning*, 48(3):730–751, 2008.

[36] Souhila Kaci, Leendert W. N. van der Torre, and Emil Weydert. Acyclic argumentation: Attack = conflict + preference. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *ECAI*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 725–726. IOS Press, 2006.

[37] Souhila Kaci, Leendert W. N. van der Torre, and Emil Weydert. On the acceptability of incompatible arguments. In Mellouli [40], pages 247–258.

[38] Antonis C. Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *AAMAS*, pages 883–890. ACM, 2003.

[39] J. Loeckx, H.-D. Ehrich, and M. Wolf. Algebraic specification of abstract data types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic and Computer Science*, pages 219–309. Oxford Science Publications, 2000.

[40] Khaled Mellouli, editor. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007, Hammamet, Tunisia, October 31 - November 2, 2007, Proceedings*, volume 4724 of *Lecture Notes in Computer Science*. Springer, 2007.

[41] Sanjay Modgil. An abstract theory of argumentation that accommodates defeasible reasoning about preferences. In Mellouli [40], pages 648–659.

[42] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[43] Sanjay Modgil and Trevor Bench-Capon. Integrating object and meta-level value based argumentation. In *COMMA*, volume 172, pages 240–251, 2008.

[44] Simon Parsons, Nicolas Maudet, Pavlos Moraitis, and Iyad Rahwan, editors. *Argumentation in Multi-Agent Systems, Second International Workshop, ArgMAS 2005,*

Utrecht, The Netherlands, July 26, 2005, Revised Selected and Invited Papers, volume 4049 of *Lecture Notes in Computer Science*. Springer, 2006.

[45] H. Prakken and G. Vreeswijk. *Logics for defeasible argumentation.* Handbook of Philosophical Logic, Kluwer Academic Publishers, 2002.

[46] Henry Prakken. An abstract framework for argumentation with structured arguments. Technical Report UU-CS-2009-019, Department of Information and Computing Sciences, Utrecht University, 2009.

[47] Henry Prakken and Giovanni Sartor. A system for defeasible argumentation, with defeasible priorities. In *Artificial Intelligence Today*, pages 365–379. 1999.

[48] L.J. Savage. *The Foundations of Statistics.* 1954.

[49] Claudio Sossai and Gaetano Chemello, editors. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 10th European Conference, ECSQARU 2009, Verona, Italy, July 1-3, 2009. Proceedings*, volume 5590 of *Lecture Notes in Computer Science.* Springer, 2009.

[50] Stephen Toulmin. *The Uses of Argument.* Cambridge University Press, 1958.

[51] Bart Verheij. Artificial argument assistants for defeasible argumentation. *Artif. Intell.*, 150(1-2):291–324, 2003.

[52] Serena Villata. *Meta-argumentation for MAS: coalition formation, merging views, support relations and dependence networks.* PhD thesis, University of Turin (To appear), 2010.

[53] Michael Wooldridge, Peter McBurney, and Simon Parsons. On the meta-logic of arguments. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *AAMAS*, pages 560–567. ACM, 2005.

Guido Boella
Department of Computer Science
University of Turin
Torino, Italy
guido@di.unito.it

Dov M. Gabbay
Department of Computer Science
King's College, London
Bar-Ilan University, Israel
University of Luxembourg
dov.gabbay@kcl.ac.uk

Leendert van der Torre
Faculty of Sciences, Technology and Communication
University of Luxembourg
L-1359 Luxembourg
leon.vandertorre@uni.lu

Serena Villata
Department of Computer Science
University of Turin
Torino, Italy
`villata@di.unito.it`