Università degli Studi di Torino

Scuola di Dottorato in Scienza e Alta Tecnologia

Tesi di Dottorato di Ricerca in Scienza e Alta Tecnologia

Indirizzo: Informatica

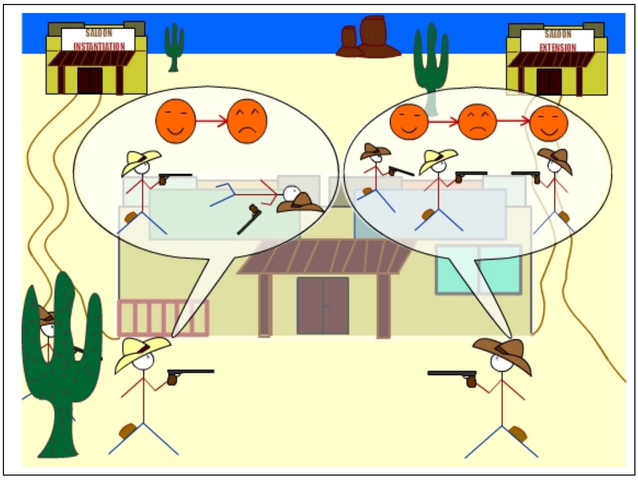
Meta-argumentation for MAS: Coalition Formation, Merging Views, Subsumption Relations and Dependence Networks

Serena Villata

Supervisors:
Prof. Guido Boella
Prof. Leendert van der Torre

XXII Ciclo, Gennaio 2010

Università degli Studi di Torino
Scuola di Dottorato in Scienza e Alta Tecnologia



Meta-argumentation for MAS: Coalition Formation, Merging Views, Subsumption Relations and Dependence Networks

Serena Villata

To my husband and my parents with love and gratitude

Abstract

In this thesis, 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 it 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. We illustrate the methodology and techniques of meta-argumentation on three challenges in formal argumentation: the representation of subsumption relation among arguments in argument ontologies and bipolar argumentation, the merging of argumentation frameworks in multi-agent argumentation and dialogue and the arguing about reciprocity-based coalitions that may emerge in social networks.

Acknowledgements

If we knew what it was we were doing,
it would not be called research, would it?
A. Einstein

Many people have contributed to the growth of this thesis. Firstly, I would like to thank my supervisors Guido Boella and Leon van der Torre. Thanks for showing me what research is. Thanks for the patience, the suggestions, the million of comments you gave me. Thanks for helping me in improving myself every time. I could not have hoped to have been supervised by two finer people, on both an academic and a personal level. Guido, Leon, you have both contributed so much towards making the last two years of my PhD two fantastic years and for this I am forever indebted to you both.

Secondly, I would thank my *third* supervisor, Dov Gabbay. Thanks for all the hints you have given to me, thanks for helping me with the big picture about research you have in your mind. I have learnt so much from you through engaging in many stimulating discussions with you and you have provided me with invaluable feedback about my work.

Thirdly, the support I have received from my family and friends throughout has been overwhelming. Mum, dad, I cannot express how very grateful I am to you both for the love and support you have given to me every day in these three years. Thanks!

My PhD started as a revolution in my life. The main result I found is my husband, Marco. We started together the PhD and, now, at the end of it, we are together in our own home fixing the last points of the thesis. My words cannot express how much your love, support and patience have meant to me throughout this time and always.

I would like to thank my academic colleagues. I am grateful to all members of the Interaction Models Research Group of Department of Computer Science at the University of Turin and of the ICR at the University of Luxembourg who have taken an interest in my work and helped me throughout the course of my PhD by providing me with feedback and engaging in discussions with me about my research. In particular, I would like to thank Martin Caminada, Valerio Genovese and Gabriella Pigozzi.

Thanks go to some people in particular who have provided me with help and engaged in discussions with me about my work: Joris Hulstijn of the Vrije Universiteit Amsterdam, Sanjay Modgil of the King's College London, Henry Prakken of the Utrecht University, Luigi Sauro of the University of Naples, Patrice Caire of the University of Luxembourg and Jan Broersen of the Utrecht University.

I would also like to thank my fellow PhD students and friends at Turin with whom I have shared the trials and tribulations of our PhD program. In particular, I would thank Elena Roglia, Roberto Grenna, Dino Ienco and Roberto Furnari.

Last but not the least, I thank my beloved cat, Ninni, for being next to me every time I was writing a paper, supporting me with his loving purr.

List of Figures

1.1 The position of argumentation theory and other research fields. 18

1.2 A dialogue between two lawyers about suspect's arguments. 22

1.3 A child arguing about his own arguments. 23

1.4 Instantiation of an abstract argument. 27

2.1 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 $O = \mathcal{E}'(I) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(I)))$. 37

2.2 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 2.1, the argumentation output is a function of the argumentation input $O = \mathcal{E}'(I)$, derived from the two transformations and the acceptance function, $O = \mathcal{E}'(I) = g(AA) = g(\mathcal{E}(EAF)) = g(\mathcal{E}(f(I)))$. 39

2.3 Extended argumentation framework as an instantiation: a function f transforms an extended argumentation framework AF to a basic argumentation framework AF. As in Figure 2.1, 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, $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$. 41

2.4 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 2.1, 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 5.1, 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 $O = \mathcal{E}''(I) = g'(AA') = g'(\mathcal{E}'(EAF)) = g'(\mathcal{E}'(f'(I))) = g'(g(\mathcal{E}(f(f'(I)))$. 42

2.5 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 $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a, b) \in \mathcal{E}\}$. . . 43

2.6 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 $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a, b) \in \mathcal{E}\}$. 45
2.7 Weakly $\mathcal{E} - equivalence$ between two AF. 51
2.8 Examples of extended argumentation frameworks. . . . 57

3.1 The notions of refinement and abstraction of an AF. . . 80
3.2 The representation of the example proposed by Baroni *et al.* [BCGG09] in our meta-argumentation model. . . 82
3.3 Graphical representation of the extended argumentation network and the flattened one. 84
3.4 Two examples of higher-order attack in the flattened argumentation network. 85
3.5 Example of higher-order attacks between four arguments. 86
3.6 An argumentation network in the meta level (a) and object level (b). 88

4.1 Example of attack from a subsumption argument to another argument. 106
4.2 Example of attack from a subsumption argument to another argument in the meta model. 107
4.3 Example of attack from an argument to a subsumption argument. 108

4.4 Example of attack from an argument to a subsumption argument in the meta model. 109
4.5 Example of attacks on the subsumption relation. 110
4.6 Example of attack on the subsumption relation in the meta model ($b \rightarrow c$). 111
4.7 Example of attack on the subsumption relation in the meta model ($c \rightarrow b$). 112
4.8 Toulmin scheme. 113
4.9 Modelling Toulmin scheme using subsumption relation. 116
4.10 Modelling Toulmin scheme using attack relations only. 117
4.11 Toulmin scheme and implications. 118

5.1 Extended argumentation framework as an instantiation: a function f transforms an extended argumentation framework AF to a basic argumentation framework AF. As in Figure 2.1, 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, $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$. 123
5.2 An example of merging with technique 1. 125
5.3 An example of merging with technique 2. 127
5.4 An example of merging with technique 3. 132
5.5 An example of merging with technique 3 attack, non-attack and ignorance. 134
5.6 Example of cycle analysis using technique 2. 137
5.7 Example of merging with the three degrees of distrust. 141
5.8 Example of merging with an agent attacking a non-attack relation of another agent. 142
5.9 An example of merged 2^{nd} order AFs. 144
5.10 Merged 2^{nd} order AFs in meta-argumentation. 145

5.11 An example of introduction of agents in argumentation. 148
5.12 Graphical representation of the above dialogue. 150
5.13 An example of merging with a non-attack relation. . . 151

6.1 UML diagram of the ontology of our model. 160
6.2 Iterative social network design. 161
6.3 Coalition view. 162
6.4 Dynamic dependence view. 162
6.5 Power view. 163
6.6 Agent view. 163
6.7 (a) Modgil - Bench-Capon scheme, (b) Higher-order argumentation. 165
6.8 Example of second-order arguments. 168
6.9 Flattened version of the networks of Figure 6.7 with X and Y meta-arguments. 170
6.10 Candidate coalitions attacking each other from Example 12. 171
6.11 The legend of the graphical representation of the modeling activities of *dependency* and *dynamic dependency*. 175
6.12 Dependence Network of Example 14. 177
6.13 Agents forming a coalition (a) or not (b); the coalition view and the dynamic dependence view merged together (c) 178
6.14 Candidate coalitions sharing goals. 182
6.15 Two vulnerable coalitions attacking each other. 186

8.1 The mapping between dependence networks and voluntary argumentation networks. 208

List of Tables

1.1 Notation used in the thesis. 29

List of Algorithms

1 FLATTENING_ALGORITHM 87

Contents

1 Introduction **17**

1.1 Objective 21

1.2 Methodology 27

1.2.1 Subsumption relation 29

1.2.2 Merging views 30

1.2.3 Coalition formation and dependence networks . 31

1.3 Structure of the thesis 32

2 Meta-argumentation methodology **35**

2.1 An informal introduction 35

2.1.1 Unifying instantiations and extended argumentation 36

2.1.2 Meta-argumentation methodology 42

2.1.3 Meta argumentation viewpoint 45

2.2 Methodology 1: Instantiating arguments 48

2.2.1 Dominance as argumentation 48

2.2.2 Abstraction in meta-argumentation 51

2.2.3 Instantiating abstract arguments 54

2.3 Methodology 2: Extending Dung's framework 56

2.3.1 Some examples of extending Dung's basic framework . . . 57
2.3.2 Applying Dung's theory of abstract argumentation to itself . . . 60
2.4 A unified methodology . . . 61
2.4.1 Baroni and Giacomin's formal framework . . . 61
2.4.2 Acceptance functions in meta-argumentation . . . 64
2.4.3 Meta-argumentation methodology . . . 65
2.5 Summary . . . 66

3 Meta-argumentation techniques 67
3.1 An informal introduction . . . 67
3.1.1 Flattening . . . 68
3.1.2 Representation . . . 71
3.1.3 Specification of Dung's basic argumentation frameworks . . . 73
3.1.4 Scope of the meta-argumentation techniques . . . 75
3.2 Flattening . . . 75
3.2.1 Flattening preference based argumentation frameworks . . . 76
3.2.2 Instantiating with preferences among arguments . . . 78
3.2.3 Flattening higher order argumentation frameworks . . . 79
3.2.4 Instantiating abstract arguments . . . 89
3.3 Representation . . . 90
3.4 Specification formalisms . . . 92
3.5 Summary . . . 95

4 Argument subsumption 97
4.1 The concept . . . 97
4.1.1 Semantics (*without attacks on subsumption*) . . . 98

4.1.2 Semantics (*with attacks on subsumption*) 104
4.2 Toulmin scheme 113

5 Merging argumentation networks 121
5.1 Three ways of modelling merging 122
5.1.1 Technique 1: no agent meta-arguments 122
5.1.2 Technique 2: agent meta-arguments 126
5.1.3 Technique 3: trust arguments 130
5.2 Applications 135
5.2.1 Technique 1 135
5.2.2 Technique 2 136
5.2.3 Technique 3 139
5.3 Merging 2^{nd} order AFs 142
5.4 Merging in dialogues 147
5.5 Discussion on limitations of abstract merging 152

6 Coalition Formation 155
6.1 Iterative social network design 158
6.1.1 Coalitions in a grid-based scenario 158
6.1.2 Ontology 159
6.1.3 Iterative design: refining viewpoints on grid-based coalitions 160
6.2 Arguing on abstract coalitions models 164
6.3 Analyzing reciprocity based coalitions 172
6.3.1 Dependence Networks 172
6.3.2 Refining coalitions with dynamic dependencies 177
6.3.3 Maintaining or destroying coalitions 181

7 Related Work **187**
7.1 Subsumption relation and Toulmin scheme 191
7.2 Merging views 193
7.3 Coalition formation and dependence networks 197

8 Future Work **201**
8.1 Subsumption relation 202
8.2 Merging views 203
8.3 Coalition formation 204
8.4 Dependence networks 206

9 Conclusions **211**

Chapter 1

Introduction

Argumentation theory involves different ways for analyzing arguments and their relationship, in this thesis we are interested only in Dung's style formal abstract argumentation [Dun95] which sees each argument as an abstract entity and in which arguments are related to each other by means of attack relations. In everyday life arguments are "reasons to believe and reasons to act". Until recent years, the idea of "argumentation" as the process of creating arguments for and against competing claims, was a subject of interest to philosophers and lawyers. In recent years, however, there has been a growth of interest in the subject from formal and technical perspectives in Computer Science (CS) and Artificial Intelligence (AI), and a wide use of argumentation technologies in practical applications. In Computer Science and Artificial Intelligence, argumentation is viewed as a mechanical procedure for interpreting events, organizing and presenting documents and making decisions about actions. From a theoretical perspective, argumentation offers a novel framework casting new light on classical forms of reasoning, such as logical deduction, induction, abduction and plausible reasoning, communication explanations of advice, supporting discussion and negotiation in computer-supported cooperative work, and learning. From a human-computer interaction point of view argumentation is a versatile technique that facilitates natural system behaviour and is more easily understood by human users and operators. Generally speaking, argumentation has the potential to add value to any computer-assisted system that provides information and advice to human users or other agents. Figure 1.1 summarizes the relation between argumentation theory and other fields, particularly logic programming and multiagent systems. Roughly, there exists an interesting overlap between abstract argumentation and logic programming, which is also reflected in the similarity between argumentation and logic programming semantics. For instance, the grounded extension in abstract argumentation corresponds to the well-founded model in logic programming, and the stable extensions in abstract argumentation correspond to the stable models in logic programming. For more details about the relationship between logic programming and argumentation, see Amgoud and Besnard [AB09] and Wu and Caminada [WC09].

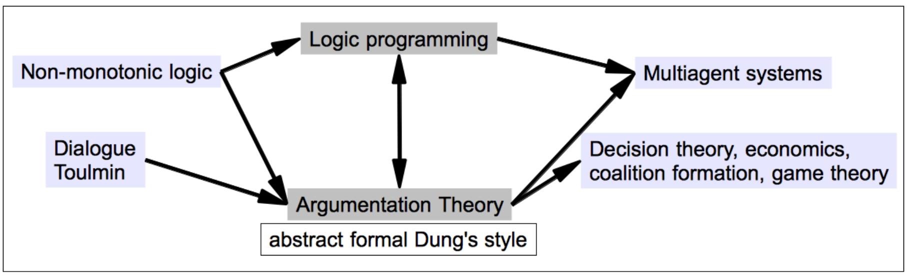


Figure 1.1: The position of argumentation theory and other research fields.

Complex technical systems and services increasingly require several autonomous agents that have to collaborate and communicate in order to achieve required objectives, because of the inherent interdependencies and constraints that exist between their goals and tasks. Increasingly they depend upon complex conversations concerned with negotiation, persuasion and trustworthiness where agents have different capabilities and viewpoints. Such dialogues have at their heart an exchange of proposals, claims or offers. What distinguishes argumentation-based discussions from other approaches is that proposals can be supported by the arguments that justify, or oppose, them. This permits greater flexibility than in other decision-making and communication schemes since, for instance, it makes it possible to persuade agents to change their view of a claim by identifying information or knowledge that is not being considered, or by introducing a new relevant factor in the middle of a negotiation or to resolve an impasse.

Argumentation is the process by which arguments are constructed and handled. Thus argumentation means that arguments are compared, evaluated in some respect and judged in order to establish whether any of them are warranted. Each argument is a set of assumptions that, together with a conclusion, is obtained by a reasoning process [BH09a]. The layout of an argument has been studied by Toulmin in 1958 [Tou58] who identified the pieces of information composing an argument. These key components are the data, the claim, the warrant and the rebuttal. A claim is a conclusion which is drawn if the warrant holds and the rebuttal does not hold. The data, supported by the warrant, imply the claim.

Argumentation as exchange of pieces of information and reasoning about them involves groups of agents. Besnard and Hunter [BH09a] assume that each argument has a proponent, the person who puts forward the argument, and an audience, the person who receipts the argument. Two kinds of views on argumentation can be highlighted in multiagent systems, monological and dialogical. In the former, a single agent or a group of agents with the same role has the knowledge to construct arguments to support and attack a conclusion while, in the latter, a group of agents interacts to construct arguments supporting or attacking a particular claim. For a deeper discussion about the different classes of argumentation in multiagent systems, see Besnard and Hunter [BH09a].

There are, at the higher level, two ways to formalize a set of arguments and their relationships, abstract argumentation and logical argumentation. Abstract argumentation has been introduced by Dung [Dun95] and it names only the arguments without describing them at all and represents that an argument is attacked by another one. Logical argumentation [Pra09] is a framework in which more details about the arguments are considered. In particular, each argument is seen as composed by the premises, the claim and the inference rules used to achieve the claim from the premises. In this thesis, we introduce the methodology of meta-argumentation as modeling technique for different kinds of abstract argumentation. A discussion about pros and cons of using abstract argumentation and a comparison with logical argumentation are provided in the following.

We apply the methodology of meta-argumentation to three interdisciplinary challenges that in recent years have involved the research area of argumentation theory: coalition formation, merging views and support relations. Our approach regarding these challenges consists in a design perspective, in which different modeling techniques are necessary to model a particular concept. The first point follows from the works of Amgoud [Amg05] and Bulling *et al.* [BDC08], using argumentation to decide what coalitions should be formed. In this challenge, not only arguments are abstract entities but, being abstract entities they can represent everything, also a coalition for instance. Coalitions are, thus, viewed as abstract entities too and we highlight their composition using dependence networks, introduced by Conte and Sichman [SC02]. The second point sees arguments as abstract entities and, following Coste-Marquis *et al.* [CMDK$^+$07], different kinds of relationships between the arguments are considered, particularly attack, non-attack and ignorance. The third point sees an argumentation framework as composed by abstract arguments and two kinds of relations, attack and subsumption, as introduced by Cayrol and Lagasquie-Schiex [CLS05] in bipolar argumentation.

1.1 Objective

Consider the dialogue between the two lawyers in Figure 1.2. 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 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

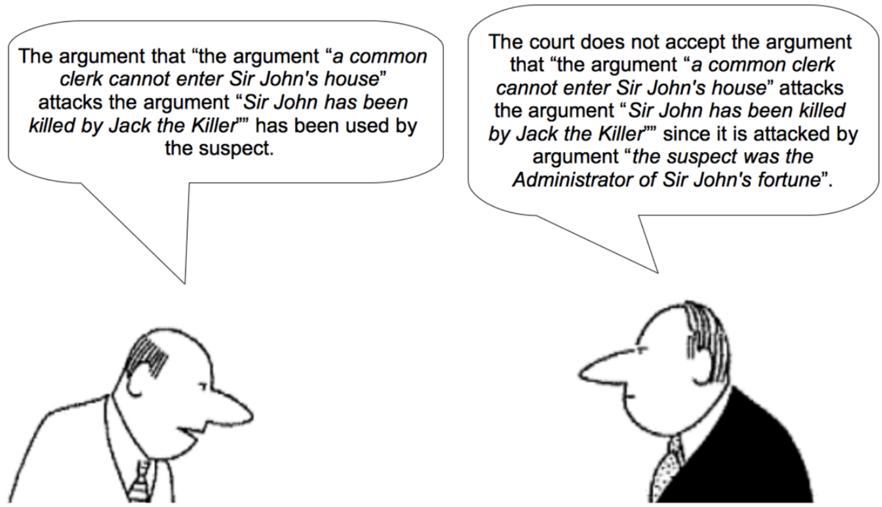


Figure 1.2: A dialogue between two lawyers about suspect's arguments.

one hand, Jakobovits and Vermeir [JV99] 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 [CLS05] 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 [MBC08] 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 Chapter 7.

In this thesis 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 1.3. 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*".

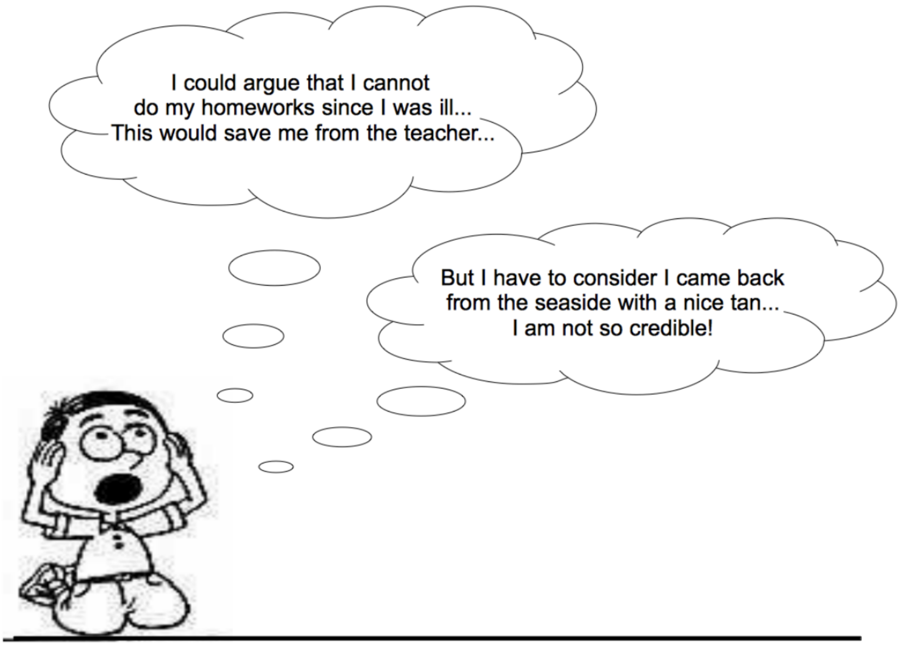


Figure 1.3: A child arguing about his own arguments.

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 [BCD07], decision making [KM03], coalition formation [Amg05], combining Toulmin's micro arguments [Tou58], normative reasoning [ABC05], 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 [Pra09].

Extending Dung's framework. Alternatively, the description of argumentation frameworks is extended, for example with preferences among abstract arguments [AC02, KvdT08], abstract value arguments [BC03], second- and higher-order attack relations [Mod07, BGW05, Mod09], support relations among abstract arguments [CLS05], or priorities among abstract arguments [PS99].

In this thesis, we argue 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 [BvdTV09e], 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 thesis, 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 subquestions:

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 [Gab09b, Gab09a], representation techniques for extended argumentation [KvdTW06, KvdTW07], and specification formalisms and logics of argumentation [BHvdT05b, GJOW02, Boc05, WMP05].

3. How to model bipolar argumentation, introduced by Cayrol and Lagasquie-Schiex [CLS05], using meta- argumentation, such that the acceptance of one argument is a reason to accept another one? Note that this is the opposite of Dung's theory, where the attack relation represents the negative relation that the acceptance of one argument is a reason to *reject* another argument.

4. How to model Toulmin's scheme [Tou58] using meta- argumentation? Toulmin models the process of defending a particular claim against a challenger, raising several challenges such as the modeling of micro arguments together with their relationships of defeat and support.

5. How to model multi-agent argumentation using meta-argumentation? Whereas agents are explicit in dialogue proof theories, in Dung's theory they are abstracted away. We are interested in particular in the merging of argumentation frameworks [CMDK+07], because of its application in multiagent systems. Each argumentation framework represents the set of beliefs of the agents of a multiagent systems, such that the merged argumentation framework represents the beliefs accepted by the group.

6. How to analyze the reciprocity-based coalitions that may emerge in social networks at various degrees of abstraction using meta-argumentation? We are interested in the analysis based on cooperation which emerges in 'small' social networks in order to achieve a greater number of goals. As a measure of cooperation, we analyze the coalitions [SK98] that emerge in a social network assuming reciprocity, for example measuring the number of coalitions [BvdTV09d], the kinds of coalitions [BvdTV08d], or the stability of the coalitions.

Figure 1.4 provides an abstract example of argument instantiation. Argument $a \to b$ 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 $b \to a$ in such a way to establish the preference of a.

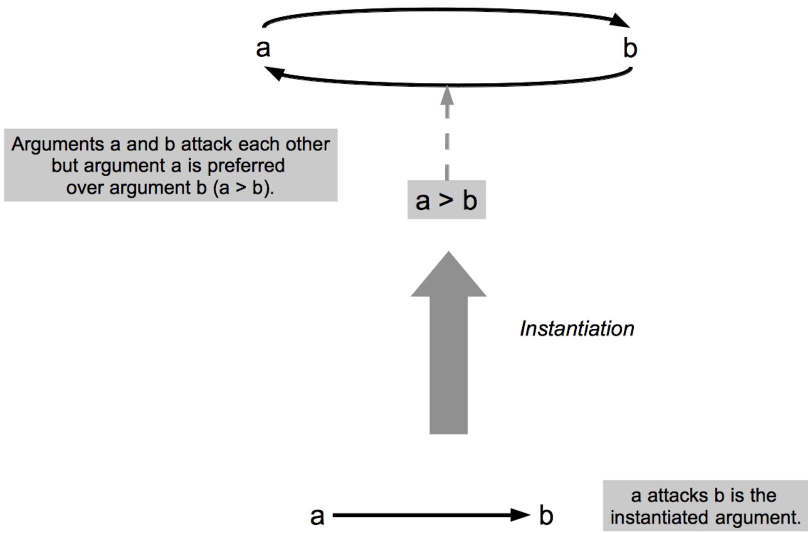


Figure 1.4: Instantiation of an abstract argument.

1.2 Methodology

We consider three techniques used in meta-argumentation: flattening, representation and specification languages. For higher-order attacks, in Boella *et al.* [BvdTV09e] we use the Jakobovits-Vermeir [JV99] and Caminada [Cam06] labeling to introduce meta-arguments like 'argument A is accepted' or 'argument A is undecided'. Following several similar proposals in the recent literature [Mod09, Gab09b, Gab09a], 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 [Gab09b, Gab09a], which may be seen as a generalization of our earlier work, as well as a growing body of other earlier work [BCGG09, MBC08, Mod07, BvdTV08d, BvdTV08a]. 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 [BvdTV09e] 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 thesis, we explain the methodology and techniques using these two examples. Following several similar proposals in the recent literature by Modgil [Mod09] and Gabbay [Gab09b], we use X and Y meta-arguments to model second and higher order attacks.

In this thesis, 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, and finally the formalization of the coalition formation process in the context of iterative design of social dependence networks. Table 1.1 summarizes the notation of meta-argumentation used in this thesis.

NOTATION	MEANING
U	universe of all generated arguments
$A \subset U$	a finite set of arguments
$a, b, c, \ldots \in A$	elements of A
$\rightarrow$	binary relation on A representing attack
MU	universe of all meta-arguments
$accept(a)$	"argument a is acceptable"
MA	a set of meta-arguments
$\longmapsto$	a relation on MA
EAF	an extended AF
$\mathcal{EAF}$	a set of possible EAF
f	function from EAF to AF
AF	a pair of A and $\rightarrow$
$\mathcal{AF}$	a set of possible AF
$\mathcal{E}$	mapping from $\langle A, \rightarrow \rangle$ to sets of subsets of A
g	function from accepted MA to accepted A
$\Rightarrow$	binary relation on A representing subsumption
$\nrightarrow$	binary relation on A representing non-attack
X	meta-argument for attack (de-active)
Y	meta-argument for attack (active)

Table 1.1: Notation used in the thesis.

1.2.1 Subsumption relation

First, we provide a representation of subsumption relation, in order to model a kind of support relation, as done by Amgoud *et al.* [ACLSL08] for bipolar argumentation frameworks. In this framework, Dung's argumentation framework is extended with a new kind of binary relation representing support. Although the extended argumentation framework presented by Amgoud *et al.* [ACLSL08] is similar to our one, we analyze also the consequences of an attack from and to the arguments belonging to the subsumption relation and on the subsumption relation itself in such a way to know if new attack relations arise from the existing ones.

Second, the representation of the subsumption relation between arguments allows us to model the Toulmin scheme [Tou58], using the warrant as subsumption relation argument between the data and the claim. Rebuttals are represented as attack relations on the claim and the absence of a warrant is equal to an attack on the subsumption relation.

1.2.2 Merging views

Conflict resolution is at the basis of Dung's argumentation theory, and also of the merging of argumentation frameworks. In the latter case, one agent may argue that one argument attacks another one, whereas another agent argues for the opposite. For the resolution of this conflict, the society or multiagent system has to decide whether both arguments attack each other, they do not attack each other, or one attacks the other, but not vice versa. Possible solutions of this conflict problem are trust, authority, and so on, but from an abstract level of analysis a solution is merging the different argumentation frameworks.We aim at providing different techniques which can be used in order to merge argumentation frameworks coming from different agents. This characterization of merging uses the idea of meta-argumentation, because there are also arguments about the existence of attack relations. Other approaches to merging argumentation frameworks are given by [CMDK$^+$07, CP09, PTG08, BE09]. We can see coalition formation as a merging argumentation frameworks problem: the merged argumentation frameworks represent the way coalitions will operate. The epistemic merging of agents' beliefs would lead to a more stable coalition thus merging personal argumentation frameworks of the agents belonging to the same coalition helps in maintaing the coalition's stability. Using different techniques for doing merging, we obtain various degrees of stability depending on the merged argumentation framework resulting from the merging process.

1.2.3 Coalition formation and dependence networks

Small social networks are analyzed in software engineering, for example by the TROPOS methodology [BPG+04], developed for agent-oriented design of software systems. At the highest level of abstraction, coalitions are purely abstract and we only specify whether the creation of one coalition will block the creation of another coalition. We say that two coalitions are attacking each other and the second-order argument sets a preference of the first coalition over the second one, and we use abstract argumentation theory [Dun95] to determine the acceptable coalitions. At the second level of abstraction, we detail the composition of a coalition which is seen as a set of agents and a set of dependencies between them. Our notion of coalition is based on the concept of reciprocity which constraints each node to contribute something, and to get something out of it. At the third level of abstraction, we detail the powers and goals of the individual agents. At the fourth level of abstraction, we also detail the beliefs, decisions and goals of the agents. For the analysis we focus on the coalition and dependence views, and leave a detailed analysis of the power and agent views for further research.

We illustrate our approach using a grid scenario. Consider, for example, a virtual organization for e-Science composed by nodes belonging to academic institutions such as universities and research centers. Inside the virtual organization, sub-groups can be formed with the aim to collaborate in order to achieve a greater number of goals, i.e., if node a cannot store a file but it can help node b in doing a computation and b can store a's file, these two nodes form a reciprocity based coalition in order to achieve both goals. It would be possible that two or more candidate coalitions share the same goals, e.g. two nodes can do the storage for node a and thus it becomes necessary to have a mechanism to decide what coalition can be formed.

Using social dependence networks to represent the multiagent system, as in TROPOS [BPG+04], allows us to model, particularly for the requirements analysis phase of the design process, the domain stakeholders. The analysis of cooperation in this context is relevant since agents can form coalitions with the aim to achieve more goals than what they can achieve alone. As in well known game theoretic approaches to cooperation [SK98], we face with problems of incompatibilities between the possible coalitions which can be formed. We manage these incompatibilities using an argumentation framework treating each candidate coalition as an argument, the incompatibilities as the attacks between the arguments and, finally, using the extensions to find out the acceptable coalitions.

1.3 Structure of the thesis

The thesis follows the research questions and is organized as follows. Chapter 2 introduces the methodology of meta-argumentation, starting with a general introduction, introducing Dung's argumentation framework and abstraction, various extended argumentation frameworks proposed in the literature and reductions to Dung's basic theory, and finally Baroni and Giacomin's framework [BG07] and acceptance functions.

Chapter 3 introduces the techniques by first giving an informal introduction, then introducing 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.

In chapter 4, we provide a definition of the subsumption relation in the context of argumentation networks, comparing it with the notion of support of bipolar argumentation. We model various kinds of attacks on the arguments involved in the subsumption relation and, finally we analyze the possibility to attack the subsumption relation itself.

Moreover, we concentrate our efforts in defining the well-known Toulmin scheme in which a claim is supported by a warrant and would be attacked by a rebuttal. The Toulmin scheme is modeled, thanks to the meta-argumentation methodology, using the subsumption relation. Two different representations of this scheme using two different meta-argumentation languages are provided.

Chapter 5 presents three merging techniques and various kinds of applications of this techniques. Merging is analyzed from a multiagent point of view, involving a dialogue perspective in which the agents interact with each other by means of arguments and the relationships between their arguments are identified by attack relations, non-attack relations and ignorance.

In chapter 6 a social network approach to coalition formation is presented. In the iterative design perspective, a coalition is firstly seen as a dependence network respecting the reciprocity constraints while, at a more abstract level of abstraction, it is seen as an argument and higher-order attacks between coalitions hold. Coalition formation thus is represented in the coalition view by using meta-argumentation in order to know which coalitions should be formed.

In chapter 7, we relate our methodology with some results in the field of argumentation theory. First, we compare our approach introducing subsumption in argumentation with the approach extending the argumentation frameworks with the support relations. Second, we compare our approach to merging to the recent works in this field both in the argumentation community and in the MAS one. Third, we provide a survey about other works approaching coalition formation using argumentation theory and we compare them with our approach.

Chapter 8, we discuss possible research lines to further improve the meta-argumentation methodology and its application to the three challenges.

Conclusions end the thesis.

Chapter 2

Meta-argumentation methodology

In this chapter, 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 Chapter 3.

2.1 An informal introduction

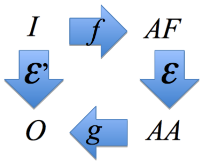
We start with an informal introduction about meta-argumentation theory, highlighting the two well known methodologies of extending and instantiating argumentation.

2.1.1 Unifying instantiations and extended argumentation

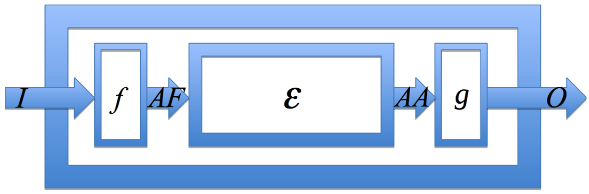
Dung's argumentation theory formalizes the reasoning leading to accepted arguments, on the basis of attacks among arguments. In Dung's terminology, it is a theory of argumentation semantics, which relates attack relations among arguments to acceptable arguments. In our terminology, it is a theory of 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 2.1. Using elementary mathematics, Figure 2.1(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 2.1(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 thesis.



(a) Categories



(b) Basic Argumentation System

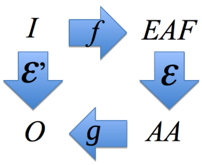
Figure 2.1: Instantiating Dung's basic argumentation theory: a function f transforms an argumentation input I to an argumentation framework AF, whose extensions of accepted arguments $AA = \mathcal{E}(AF)$ are transformed back into the argumentation output O. The argumentation output is a function of 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 2.1. 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 2.1 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.* [DMT07], 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.* [BDKT97] are not sets of arguments but sets of assumptions and in [DMT07] it is shown that an equivalent fully argument-based formulation, as introduced in [Dun95], 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.

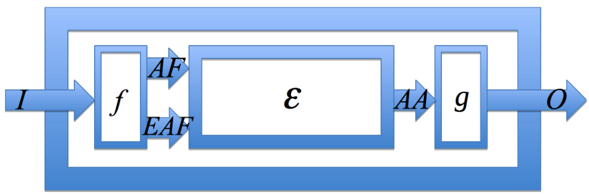
Another way to use the diagram is for cases when we have an input I and an output O, but we do not have the relation between them, i.e. we do not have the function $\mathcal{E}'$. The function may be partially known, for example we want the relation between input and output to satisfy some principles, 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 2.1 has been modified by extending Dung's argumentation framework with other relations among abstract arguments, such as preference-based relations [AC02], value-based relations [BC03], support relations in bipolar argumentation [CLS05], second- and higher-order attack relations [Mod07, BGW05, Mod09] and priorities relations among abstract arguments [PS99].

The use of an extended argumentation framework is visualized in Figure 2.2. Figure 2.2(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 2.2(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

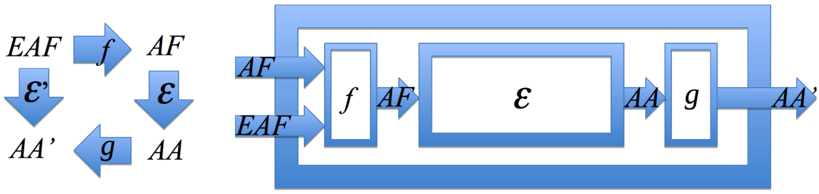
Figure 2.2: Extending Dung's theory: a function f transforms an argumentation input I to an extended argumentation framework EAF, which contains besides attack relations among arguments represented in AF also other kind of relations among arguments. As in Figure 2.1, the argumentation output is a function 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 5.1. Figure 5.1(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 5.1(b) describes the related instantiation as an instantiated argumentation system.

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

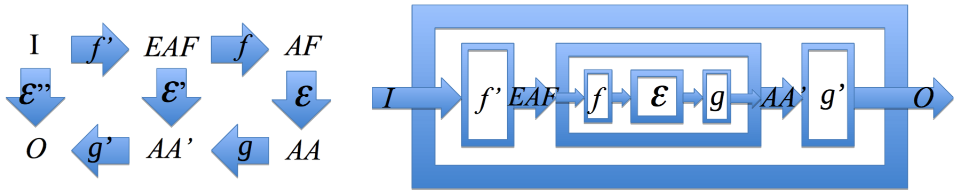


(a) Categories (b) Argumentation System

Figure 2.3: Extended argumentation framework as an instantiation: a function f transforms an extended argumentation framework AF to a basic argumentation framework AF. As in Figure 2.1, 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, $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$.

and instantiations is visualized in Figure 2.4. Figure 2.4(a) describes the instantiation using again the various functions by combining the functions from Figure 2.1(a) and Figure 5.1(a). Figure 2.4(b) describes combination as an instantiated argumentation system, which replaces the component E of Figure 2.1(b) by the whole argumentation system of Figure 5.1(b).

Summarizing, the functional compositions and the combination of argumentation systems in Figure 2.4 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.



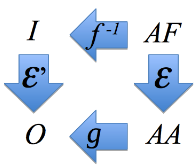
(a) Categories (b) Argumentation System

Figure 2.4: 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 2.1, 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 5.1, 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 $O = \mathcal{E}''(I) = g'(AA') = g'(\mathcal{E}'(EAF)) = g'(\mathcal{E}'(f'(I))) = g'(g(\mathcal{E}(f(f'(I)))$.

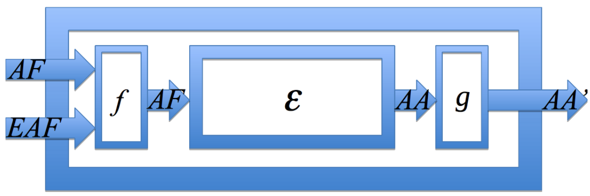
2.1.2 Meta-argumentation methodology

The general methodological problem we consider in this thesis 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 2.5. It is the same figure as the instantiation problem of Dung's theory in Figure 2.1, 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



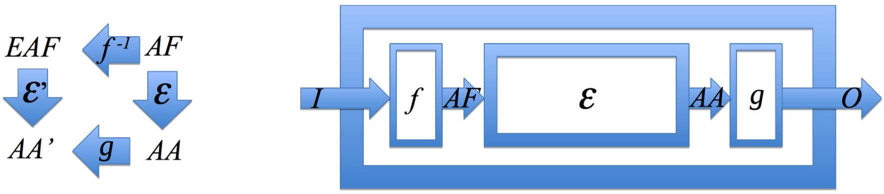
(b) Basic Argumentation System

Figure 2.5: 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 2.6.



(a) gori

(b) Argumentation System

Figure 2.6: 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.* [WMP05] 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 chapter 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 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 [BH09b]. 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 [Han01]. 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 [Dun95] 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 2.7. We therefore prefer to refer to acceptance functions over argumentation semantics.

$$A_1 = a, b, c$$
$$A_2 = a, b, c$$

$$R_1\colon\ a \longrightarrow b \longrightarrow c$$
$$R_2\colon\ c \longrightarrow b \longrightarrow a$$

$$Ext_1 = \{a, c\}$$
$$Ext_2 = \{a, c\}$$

Figure 2.7: Weakly $\mathcal{E} - equivalence$ between two AF.

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 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 [Dun95]:

> "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 [PV02] and Bench-Capon and Dunne [BCD07]. 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 [Pra09] 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 [Pra09] 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 [CA07], 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 [AB09] and Caminada and Amgoud [CA07].

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 Chapter 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 framework

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 [BCD07], Toulmin schemes [Tou58] or examples from normative reasoning [ABC05], the language is typically extended, for example with preferences among arguments [AC02, KvdT08], value arguments [BC03], second- and higher-order attack relations [Mod07, BGW05, Mod09], support relations among arguments [CLS05], or priorities among arguments [PS99]. 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 [Gab09b, Gab09a].

2.3.1 Some examples of extending Dung's basic framework

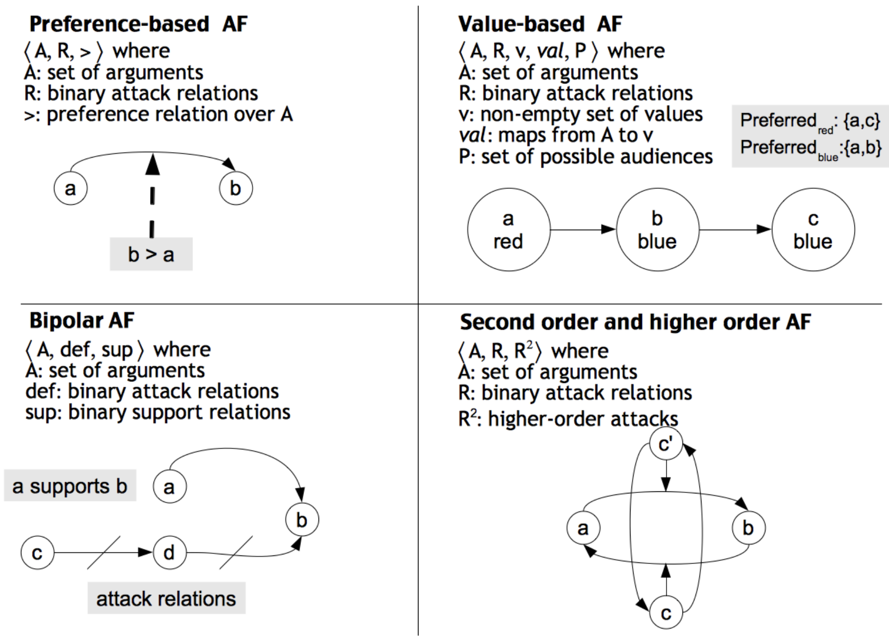


Figure 2.8: Examples of extended argumentation frameworks.

Four examples of extended argumentation frameworks are illustrated in Figure 2.8. Preference-based argumentation introduces a preference relation between the arguments. For example, as shown in Figure 2.8, Amgoud [AC02] 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 2.8. See Kaci and van der Torre [KvdT08] 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 [Mod07] 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 2.8 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 [MBC08], the authors show how hierarchical second-order argumentation can be represented in Dung's theory using attack arguments. Moreover, Barringer *et al.* [BGW05] argue that the attack of b to $d \rightarrow c$ can itself be attacked.

Abstract argumentation networks were generalized by Bench-Capon [BC03], 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 2.8 from Bench-Capon [BC03], 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 [CLS05]. 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 2.8.

Toulmin [Tou58] 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 [Tou58], Bench-Capon [BC98] 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 [Boc03]. 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

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 thesis, we use four ideas from the recently introduced formal framework for the evaluation of extension-based argumentation semantics introduced by Baroni and Giacomin [BG07]. 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 [BKvdT09b, BKvdT09a], 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 [BG07] 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* $\nexists \alpha, \beta \in S$ *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 [Cam06] 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 [Dun07]. 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 [BHvdT05b]. 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 chapter. 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:

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

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

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 Chapter 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 chapter.

Chapter 3

Meta-argumentation techniques

In this chapter, 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 An 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 Chapter 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:

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

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 thesis. 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 [Gab09b, Gab09a].

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 [KvdTW06, KvdTW07]. 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 [Pra09] 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 [Sav54]. 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.* [BHvdT05b]. 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 this thesis, 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 [BC02] 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 [LEW00] 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 chapter extended argumentation frameworks with besides the attacks also preferences among arguments. Abstractly, in this chapter the set of extended argumentation frameworks $\mathcal{EAF}$ contains all preference based argumentation frameworks $EAF = \langle A, \rightarrow, \succ \rangle$ where A is a subset of the universe of arguments, $\rightarrow$ is a binary relation on A, and $\succ$ 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 $\langle A, \rightarrow, \succ \rangle$ where $A \subseteq \mathcal{U}$ is a set of arguments and $\rightarrow \subseteq A \times A$ is a binary relations over A, and $\succ \subseteq A \times A$ 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

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

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*

$$accept(a) \longmapsto accept(b)\ if\ and\ only\ if\ a \rightarrow b\ and\ not\ b \succ a$$

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 Chapter 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 chapters.

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 chapter we consider extended argumentation frameworks with besides the attacks also attacks among attacks. Abstractly, in this chapter 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 25 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*.

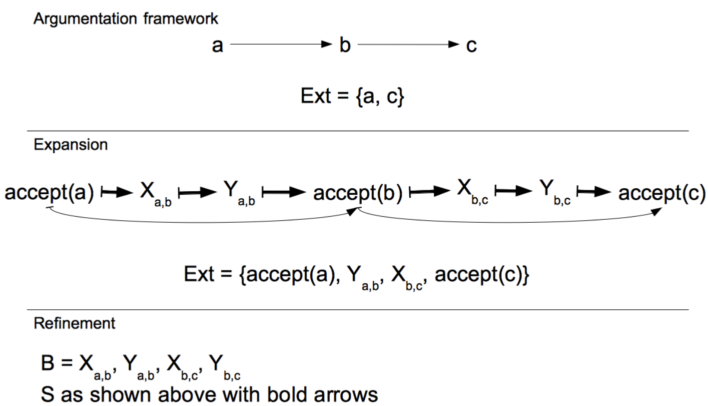


Figure 3.1: The notions of refinement and abstraction of an AF.

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

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

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

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

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

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.* [BCGG09] and represented in Figure 3.2. 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. At first sight, the first network of the example could seem simpler than the other one built with the flattening algorithm but the advantage of our meta-argumentation methodology, apart from the discussion about instantiation and abstraction provided in the previous chapters, consists in an easier way to get the accepted arguments, given an argumentation framework. For complex argumentation networks, our flattening algorithm allows

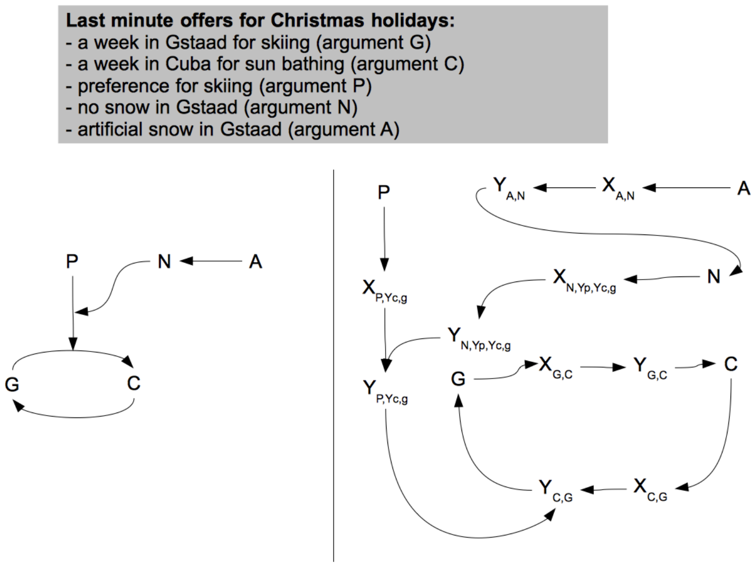


Figure 3.2: The representation of the example proposed by Baroni *et al.* [BCGG09] in our meta-argumentation model.

to build a Dung's network in which higher order attacks are represented as meta-arguments and the labeling can be computed in an easier way.

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 25, 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*

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

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

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

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 [Gab09b, Gab09a]. 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 3.3. 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 3.3.

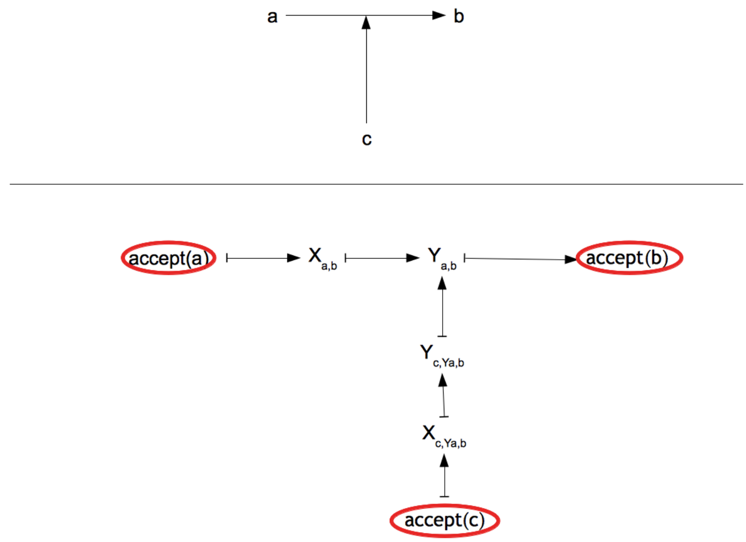


Figure 3.3: Graphical representation of the extended argumentation network and the flattened one.

As discussed in Chapter 2, an attack can itself attack by a higher-order attack another argument, as shown in Figure 3.4(a). Argument c is attacked by the attack $a \to b$. This attack is raised by meta-argument $Y_{a,b}$ which is the meta-argument representing the "active" state of the attack $a \to b$. The extension of this argumentation framework is $\{accept(a)\}$.

Another example is shown in Figure 3.4(b) where, starting from Figure 3.4(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 sense. The attack of d is translated in the object level in an attack of d to the two meta-arguments representing its

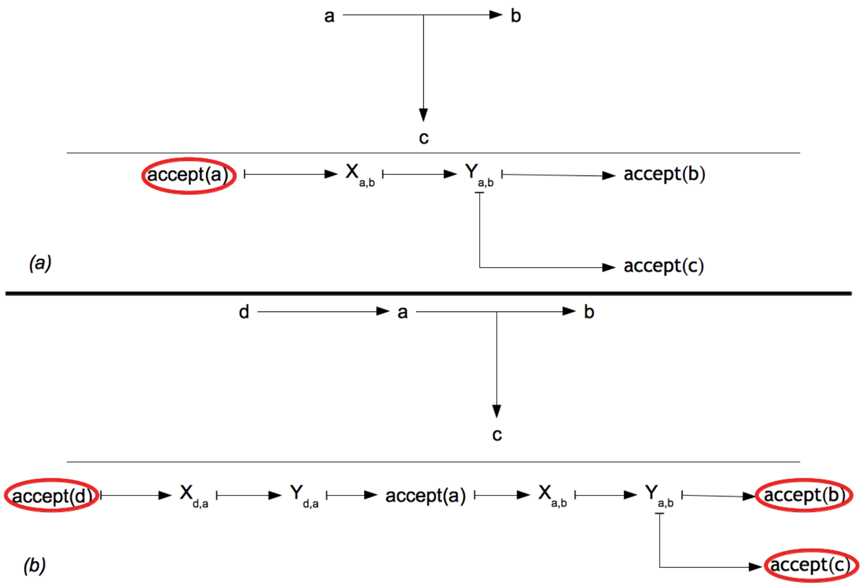


Figure 3.4: Two examples of higher-order attack in the flattened argumentation network.

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 $a \longrightarrow b$, represented by $Y_{a,b}$, is not in the extension being $accept(a)$ not in the extension too.

Figure 3.5 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 $a \longrightarrow b$ 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: $\{accept(a), Y_{a,b}\}$.

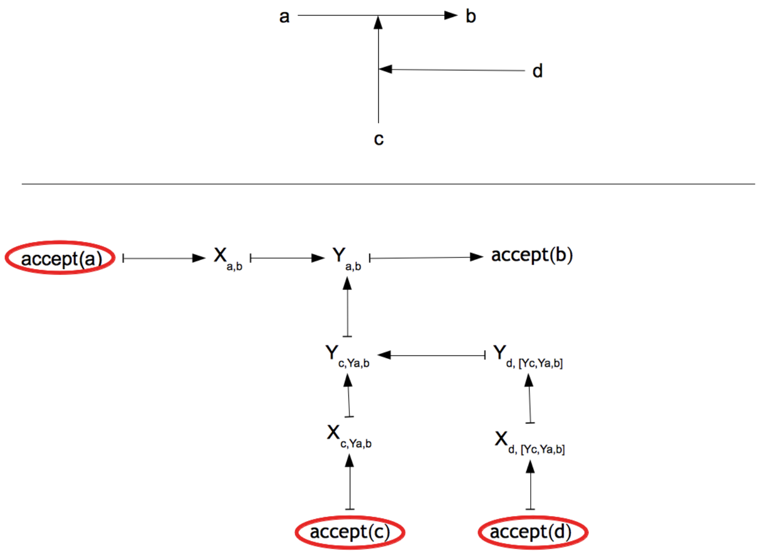


Figure 3.5: Example of higher-order attacks between four arguments.

Finally a more complex argumentation network is presented in Figure 3.6. 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 3.6(b). The extended argumentation framework has the following extension: $\{accept(c), accept(a)\}$.

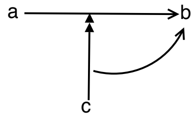
In order to give a procedural way of building the meta-argumentation network from a complex argumentation framework obtaining an abstract Dung's based argumentation framework, we define a flattening algorithm. The algorithm works as follows.

The algorithm uses three main functions: function $add()$ adds new arguments to the flattened argumentation framework under the

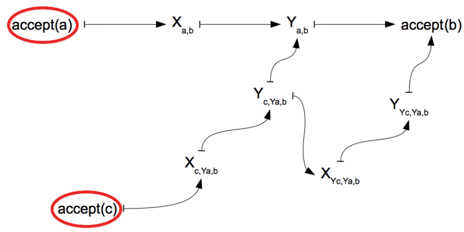
Input: An argumentation network $\langle A, R \rangle$.
Output: A flattened argumentation network $\langle N \cup A, E \rangle$

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



(a) Meta-level



(b) Object level

Figure 3.6: An argumentation network in the meta level (a) and object level (b).

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 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 25, 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 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

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

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 [Gab09b, Gab09a].

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.* [KvdTW06, KvdTW07], 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 AF [KvdTW07]) *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 ([KvdTW07]) *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 AF [KvdTW06]) *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 ([KvdTW07]) $\langle \mathcal{A}, \mathcal{R} \rangle$ *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 ([KvdTW06]) *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 AF) *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 ([KvdTW07]) *$\langle \mathcal{A}, \mathcal{R} \rangle$ 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.* [KvdTW06, KvdTW07] 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 [Mod07], and of

Baroni and Giacomin [BG07], 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 [Gab09a].

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.*[BCGG09] 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 [BCGG09].

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

- the Baroni *et al.*[BCGG09] flattening, which considers only $Y_{a,b}$ arguments;
- the Boella *et al.*[BvdTV09e] flattening, which uses only X_a meta-arguments instead of $X_{a,b}$;
- the flattening proposed in this thesis, 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's style argumentation framework, but it seems that none of the above flattenings returns precisely a Dung's 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.* [BKvdT09b, BKvdT09a].

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.

Chapter 4

Argument subsumption

4.1 The concept

The notion of subsumption is usually related to the well known structure of ontologies, inheritance networks and syllogisms. Roughly, by subsumption we mean that, given an ontology O and two classes A and B where A is subsumed by B, we verify whether the interpretation of A is a subset of the interpretation of B in every model of O. Taxonomies based on a partial-ordering relation commonly known as $is{-}a$, or subsumption, have become an important conceptual modeling tool for knowledge-based systems and semantic lexicons. Even when arguments are abstract, we may still assume that there is an ontology of arguments, for example when one argument is a sub-argument of a longer argument. Subsumption relations among arguments are used to describe such an ontology, without describing the internal structure of the arguments. For example:

- The argument that "agent A accepts argument b" is subsumed by an argument that "agent A knows argument b".
- The argument that "bridge is a game for four players which is

complete in four deals" is subsumed by "bridge is a game in which there are four players in two fixed partnerships" in the context *chicago* but the same does not hold in the context *rubber bridge*.

- The argument that "icing and baking powder are necessary for making birthday cakes" is subsumed by an argument that "baking powder is necessary for making cakes".

If the internal structure of the arguments is known, then such a subsumption relation among arguments can be partly derived from this internal structure. For example, as suggested by the last example above, if an argument is represented by a propositional formula, then an argument a is subsumed by an argument b if the propositional sentence associated with argument a implies the propositional sentence associated with argument b. However, we do not consider such instantiations in this thesis, and restrict our discussion to the abstract level.

4.1.1 Semantics (*without attacks on subsumption*)

The notion of subsumption in this context is similar to the notion of support as discussed by Amgoud *et al.* [ACLSL08] in the context of bipolar argumentation. 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.

We propose the following meaning for a subsumption relation among arguments: if argument a is subsumed by an argument b, then argument a cannot be accepted without argument b being accepted too. In other words, if we have both that argument a is subsumed by argument b, and argument a is accepted, then we are forced to accept argument b too. For the examples above, our semantics gives the following interpretation to the subsumption relations:

- If you accept the argument that "agent A accepts argument b" then you should also accept the argument that "agent A knows argument b".

- If you accept the argument that "icing and baking powder are necessary for making birthday cakes" then you should also accept the argument that "baking powder is necessary for making cakes".

This semantics makes it explicit that "argument a is subsumed by argument b" is intuitively a stronger notion than "argument a supports argument b", because if argument a supports argument b and there is another argument c such that argument c attacks argument b, then we may have that argument a is accepted without argument b being accepted. In such a case, intuitively, argument a supports argument b, but the support was not strong enough for argument b to be accepted too. In the case of subsumption relations, if argument a is subsumed by argument b and argument a is accepted, then argument b will be accepted too, regardless of other attacks on argument b. The only way to have argument a accepted without accepting argument b is to attack the subsumption relation between the two arguments itself, but that is an issue we defer to Section 4.1.2.

There are some logical properties such subsumption relations have to obey. In particular, the following transitivity property: if a is subsumed by b and b is subsumed c, then a is subsumed by c. This follows from the semantics: if accepting b implies that c must be accepted, and accepting a implies that b must be accepted, then accepting c implies that a must be accepted.

There are two fundamental logical principles which intuitively follow from this semantics:

1. If a is subsumed by b and b attacks c, then a attacks c. For example, if the argument "agent A knows argument b" attacks the argument that "agent A does not know anything", then "agent A accepts argument b" also attacks the argument that "agent A does not know anything". Likewise if "baking powder is necessary for making cakes" attacks "every cake must be cooked without baking powder" then "icing and baking powder are necessary for making birthday cakes" attacks the argument that "cakes are cooked without baking powder" too.

2. If a is subsumed by b and c attacks b, then c attacks a. For example, if the argument "agent A knows only arguments c and d" attacks the argument that "agent A knows argument b", then "agent A knows only arguments c and d" also attacks the argument that "agent A accepts argument b". Likewise if "fruit tarts are cooked without baking powder" attacks "baking powder is necessary for making cakes" then you should also accept the argument that "fruit tarts are cooked without baking powder" attacks the argument that "icing and baking powder are necessary for making birthday cakes".

The following principles are intuitively *not* valid:

1. If a attacks b and b is subsumed by c, then a attacks c. For example, if "icing and baking powder are necessary for making birthday cakes" is subsumed by "baking powder is necessary for making cakes" and "birthday cakes can be done with baking powder and chocolate only" attacks "icing and baking powder

are necessary for making birthday cakes", then you should also accept that "birthday cakes can be done with baking powder and chocolate only" attacks "baking powder is necessary for making cakes" but this principle does not hold.

2. If b is subsumed by c and b attacks a, then c attacks a. For example, if "icing and baking powder are necessary for making birthday cakes" is subsumed by "baking powder is necessary for making cakes" and "icing and baking powder are necessary for making birthday cakes" attacks "birthday cakes are cooked without icing" then you should also accept that "baking powder is necessary for making cakes" attacks "birthday cakes are cooked without icing" but this principle does not hold.

This list of valid and invalid properties raises two questions. First, is there another reason, besides intuition for these examples, why these principles are valid or invalid? Second, even more ambitiously, what is the set of all the valid principles? To answer these questions, we turn to the logic of argumentation. We can represent that "a attacks b" by "accept(a) implies not accept(b)" and "a is subsumed by b" by "accept(a) implies accept(b)", but the question is which kind of implication is used here. For subsumption relation we can use the material implication $\supset$ from classical logic, but for attack we cannot use material implication, because from the property of contraposition it would follow from a attacks b that b attacks a: $(accept(a) \supset \neg accept(b)) \supset (accept(b) \supset \neg accept(a))$. So we use a weaker kind of implication $>$ here for representing the attack relation. Thus we have the transitivity relation:

$$(accept(a) \supset accept(b)) \wedge (accept(b) \supset accept(c)) \supset (accept(a) \supset accept(c))$$

and the fundamental properties that a is subsumed by b and b attacks c, then a attacks c:

$$(accept(a) \supset accept(b)) \wedge accept(b) > \neg accept(c)) \supset (accept(a) > \neg accept(c))$$

and if a is subsumed by b and c attacks b, then c attacks a:

$$(accept(a) \supset accept(b)) \wedge (accept(c) > \neg accept(b)) \supset (accept(c) > \neg accept(a))$$

Likewise, the logic of argumentation shows why the other principles are invalid, such as if a attacks b and b is subsumed by c, then a attacks c:

$$(accept(a) > \neg accept(b)) \wedge (accept(b) \supset accept(c)) \supset (accept(a) > \neg accept(c))$$

and if b is subsumed by c and b attacks a, then c attacks a:

$$(accept(b) > \neg accept(a)) \wedge (accept(b) \supset accept(c)) \supset (accept(c) > \neg accept(a))$$

Subsumption relations can be restricted to a context: x is subsumed by y in context C. We can restrict the conditional to a set of arguments. The logical principles hold only if they refer to arguments in the context. For example, the argument "icing and baking powder are necessary for making birthday cakes" is subsumed by argument "baking powder is necessary for making cakes" in the context C, *european cuisine*, but the same subsumption relation does not hold in the contexts C_1, *thai cuisine* or C_2, *south africa cuisine*.

Given an argumentation framework together with a set of subsumption relations, we can extend the argumentation framework using the logical principles above. We can define an extended argumentation framework $EAF = \langle A, \rightarrow, \Rightarrow \rangle$ where $\Rightarrow$ represents the subsumption relation.

Definition 17 *Let $EAF = \langle A, \rightarrow, \Rightarrow \rangle$ be an extended argumentation framework where A is the set of arguments, $\rightarrow$ is a binary attack relation and $\Rightarrow$ is a binary subsumption relation. This EAF is a meta argumentation framework $MAF = \langle MA, \longmapsto \rangle$ where:*

- MA : *a finite set of meta-arguments;*
- $\longmapsto$: *if $(b \Rightarrow a) \in \Rightarrow$, then:*
  - *if $(c \rightarrow b)$ then $(c \rightarrow a) \in \longmapsto$*
  - *if $(b \rightarrow c)$ then $(a \rightarrow c) \in \longmapsto$*
  - *if $(c \rightarrow a)$ then $(c \rightarrow b) \notin \longmapsto$*
  - *if $(a \rightarrow c)$ then $(b \rightarrow c) \notin \longmapsto$*

Finally, consider the case of dynamic argumentation. Suppose one agent has an argument and he extends it using the four principles above. Then the agent adds a new argument to the argumentation framework. The result is that he has to reconsider all subsumption relations again to see which arguments must be added now. This point highlights why efficient incremental algorithms are needed, as stated in Chapter 3. Let us consider again the *cakes example*. We have for instance that "icing and baking powder are necessary for making birthday cakes" (argument a) is subsumed by "baking powder is necessary for making cakes" (argument b) and "fruit tarts are cooked without baking powder" (argument c) attacks "baking powder is necessary for making cakes" (argument b) and "icing and baking powder are necessary for making birthday cakes" (argument a) attacks "birthday cakes are cooked without icing" (argument d). Then, according to principle 2 and to not valid principle 2, we have that $c \rightarrow a$ but only $a \rightarrow d$. If we add a new argument e "fruit tarts are cooked with less baking powder than cakes", attacking argument c, not only argument b is supported by this new argument but argument a is supported by e, too.

4.1.2 Semantics (*with attacks on subsumption*)

The real challenge of subsumption relations among arguments is to define an extension of the above for subsumption relations which themselves can be attacked too. This is a very natural operation, in particular in the Toulmin scheme we discuss in the following section. For example, if we have argument "icing and baking powder are necessary for making birthday cakes" which is subsumed by argument "baking powder is necessary for making cakes", we could have also argument "unless the birthday cake is a profiterol" attacking the subsumption relation between the other two arguments.

First, we consider the attacks from the arguments belonging to the subsumption relation to another argument, as analyzed in the previous subsection. Argument a is subsumed by argument b and argument b attacks argument c (i.e., the meta argument $Y_{a,b}$), then argument a attacks argument c (i.e., there is no meta argument $X_{a,c}$), and if argument a is subsumed by argument b and argument c attacks argument b (i.e., the meta argument $Y_{c,b}$), then argument c attacks argument a (i.e., there is no meta argument $X_{c,a}$):

Definition 18 *For all atomic arguments a and b, the meta argumen-*

tation network contains the arguments $X_{a,b}$ and $Y_{a,b}$, and the attack relation of the meta argumentation framework contains $X_{a,b}$ attacks $Y_{a,b}$, and $Y_{a,b}$ attacks b.

We say that argument a is subsumed by argument b in the set of arguments S if for all arguments $c \in S$, we have $Y_{b,c}$ attacks $X_{a,c}$ and $Y_{c,b}$ attacks $X_{c,a}$.

Let us consider the following examples of attacks. Example 1 illustrates the evolution, given that argument a is subsumed by argument b, of an attack from b to c, considering also the addition of a new attack from argument d to the attack relation $b \to c$.

Example 1 *Figure 4.1 presents the consequences of an attack from an argument b, subsuming argument a, to argument c. The dashed arrow represents the subsumption relation and the grey color means that the attack is no more valid. This attack brings to the addition of another attack from argument a to argument c, as stated before. What happens if another argument d attacks the attack between b and c? Intuitively, the consequence is that the attack of argument d attacks also the attack from argument a to argument c, due to the subsumption relation.*

Figure 4.2 provides a representation of the cases analyzed in Figure 4.1 using meta-argumentation. If a is subsumed by b and b attacks c then also a attacks c, this is represented in meta-argumentation in the following way. The subsumption relation is represented by means of an attack from meta-argument $X_{a,b}$ to meta argument $Y_{a,b}$ and another attack from meta-argument $Y_{a,b}$ to meta-argument "b" is accepted, b. The attack from b to c is represented in the usual way and this attack "activates" the attack from a to c due to subsumption. Meta-argument $Y_{b,c}$ attacks meta argument $X_{a,c}$ in order to "activate" the attack from a to c constrained to the activation of the attack from b to c (meta-argument $Y_{b,c}$ has to be accepted in order to make accepted also meta-argument $Y_{a,c}$). The extension is $\{a, b\}$. If a is subsumed by b and *b attacks c and d attacks the attack between b and d, then the attack from a to c has to be deleted. This is obtained in a natural way since argument d attacks meta-argument* $Y_{d,Y_{b,c}}$ *and this attack involves also the attack from a to c, which is now made out. The extension is* $\{a, b, c, d\}$.

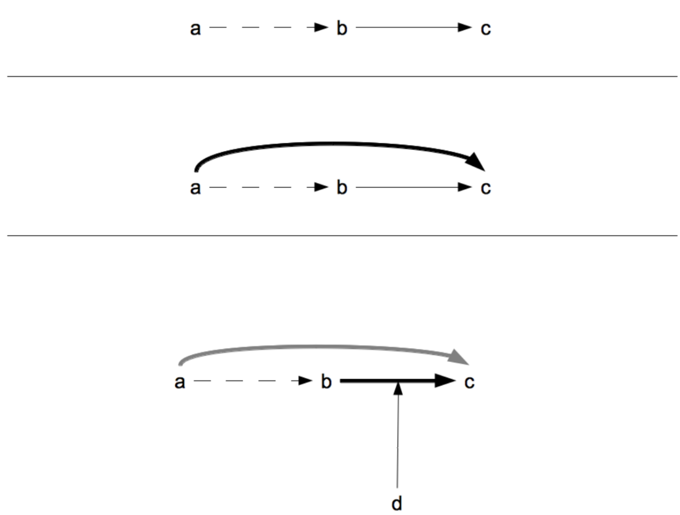


Figure 4.1: Example of attack from a subsumption argument to another argument.

Example 2 illustrates, instead, the evolution, given that a is subsumed by b, of an attack from argument c to b, also considering the additional attack from argument d to the attack relation $c \to b$.

Example 2 *Figure 4.3 presents the consequences of an attack from an external argument c to an argument b where argument a is subsumed by b. This means that argument c attacks also argument a, as stated before. This attack, then, should be deleted if another attack from argument d to the attack* $c \to b$ *is raised, due to the subsumption*

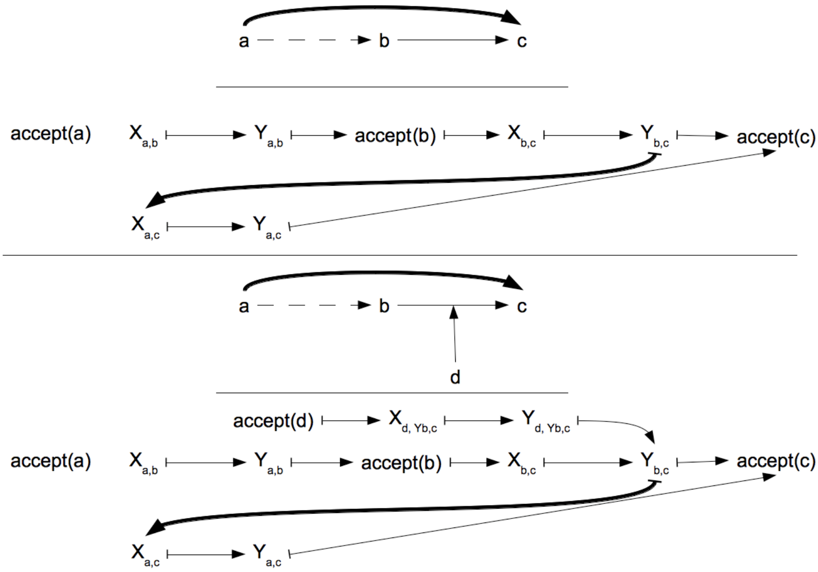


Figure 4.2: Example of attack from a subsumption argument to another argument in the meta model.

relation. In Figure 4.4, the meta-argumentation network is provided. The attack from c to b is characterized by meta-argument $Y_{c,b}$*. This meta-argument attacks meta-argument* $X_{c,a}$*, in order to "activate" the attack from argument c to argument a as a consequence of the activation of the attack from c to b. The extension is* $\{c\}$*. If there is another argument d which attacks the attack from argument c to argument b then also the attack from argument c to argument a has to be attacked. This is modeled in the following way: meta-argument* $Y_{d,Y_{c,b}}$ *attacks meta-argument* $Y_{c,b}$*, making it not accepted in all possible extension. The consequence of this "deactivation" is that also meta-argument* $Y_{c,a}$ *is made not accepted, deleting in this way the attack from argument c* *to argument a. The extension is* $\{a, b, c, d\}$.

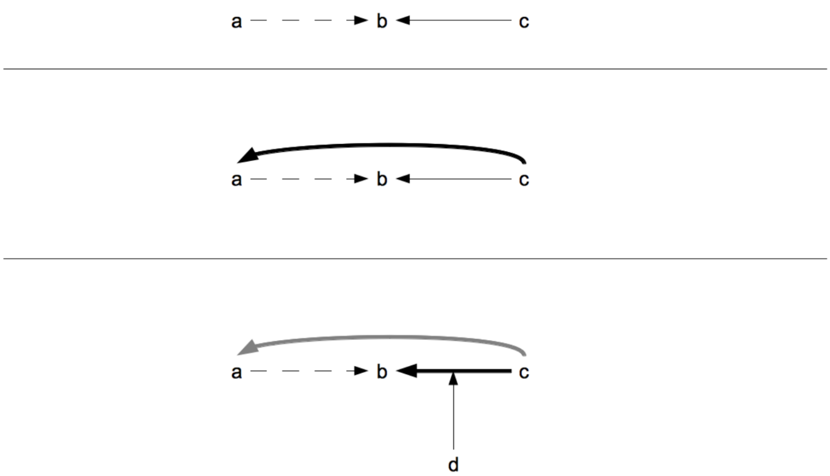


Figure 4.3: Example of attack from an argument to a subsumption argument.

The new kind of attack which should be introduced consists in the attack from an argument d to the subsumption relation itself, effecting all the attacks discussed above. We say that argument a is subsumed by argument b and argument c attacks the subsumption relation, thus we have the following attacks:

- $Y_{b,c} \rightarrow X_{a,c}$, $Y_{c,b} \rightarrow X_{c,a}$
- $Y_{a,b} \rightarrow X_{Y_{a,b},X_{a,c}} \rightarrow Y_{Y_{b,c},X_{a,c}} \rightarrow X_{a,c}$, $Y_{c,b} \rightarrow X_{Y_{c,b},X_{c,a}} \rightarrow Y_{Y_{c,b},X_{c,a}} \rightarrow X_{c,a}$
- $c \rightarrow Y_{Y_{b,c},X_{a,c}}$, $c \rightarrow Y_{Y_{c,b},X_{c,a}}$

Example 3 illustrates the evolution of an attack on a subsumption relation, in which argument a is subsumed by argument b, considering the existence of attacks like what discussed by Examples 1 and 2.

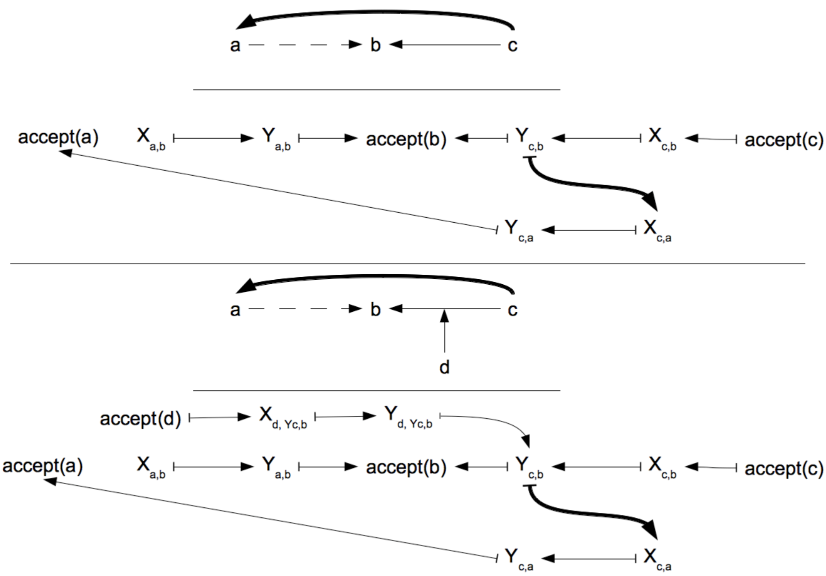


Figure 4.4: Example of attack from an argument to a subsumption argument in the meta model.

Example 3 *Figure 4.5 introduces the constraints set by an attack on a subsumption relation. There are at least two cases which should be discussed. First, if argument a is subsumed by argument b and argument c attacks argument b and argument d attacks the subsumption relation, the link between* $c \rightarrow b$ *and* $c \rightarrow a$*, here represented by means of a dotted arrow, should be deleted. Second, if argument a is subsumed by argument b and argument b attacks argument c and argument d attacks the subsumption relation, the link between* $b \rightarrow c$ *and* $a \rightarrow c$ *should be deleted.*

Example 4 illustrates how the evolution described in Example 3 is

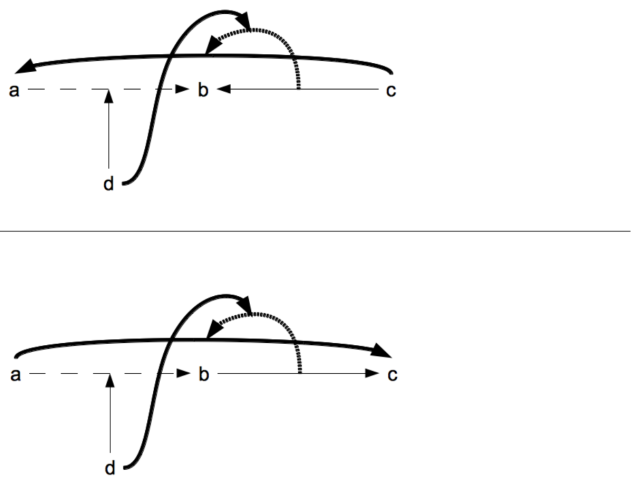


Figure 4.5: Example of attacks on the subsumption relation.

translated in our meta argumentation model.

Example 4 *Figure 4.6 provides the following example: argument a is subsumed by argument b and it attacks argument c. Argument d attacks the subsumption relation, thus the link between the attack from b to c and the attack from a to c does not hold anymore. The figure presents this example by means of our meta argumentation model. In particular, meta argument $Y_{a,c}$ which represents the attack from a to c is attacked by meta argument $X_{c,d}$, representing the new argument d. The extension is $\{a, b, d\}$.*

Conversely, the case in which argument a is subsumed by argument b and it is attacked by argument c. Argument d attacks the subsumption relation, thus the link between the attack from c to b and the attack from c to a do not hold anymore. Figure 4.7 presents this in our meta argumentation model. Meta-argument $X_{c,d}$ attacks the meta argument which represents the attack from argument c to argument a,

$Y_{c,a}$, allowing to argument a to be in the extension of this argumentation framework. The extension is $\{a, c, d\}$.

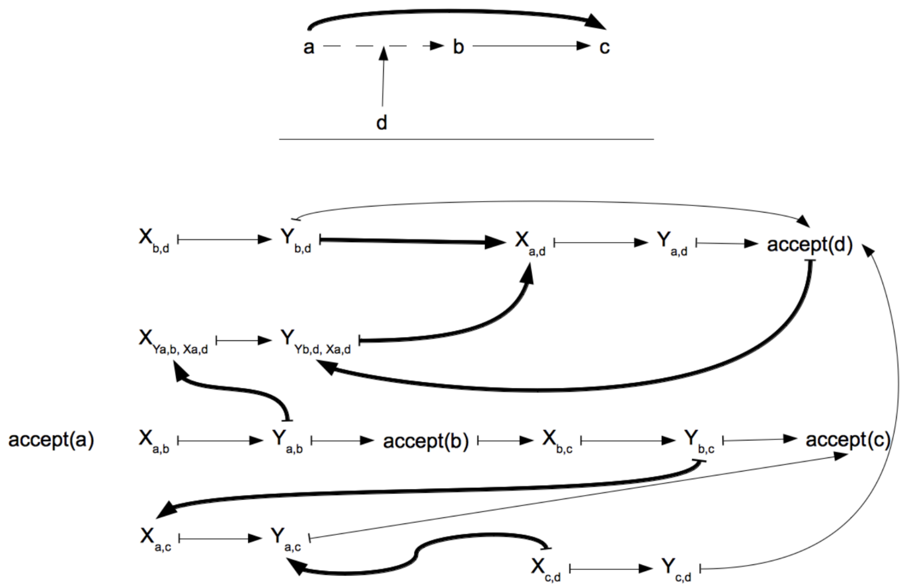


Figure 4.6: Example of attack on the subsumption relation in the meta model ($b \rightarrow c$).

Representing with meta argumentation the subsumption relations and the attacks which can be raised by the arguments involved in this relation gives the opportunity to discuss the meaning of the attacks between X and Y meta-arguments. In particular, attacks on the Y meta-arguments lead to an attack on the attack relation while attacks on the X meta-arguments lead to attacks on the subsumption relations and on the links between the attacks, when an attack has as consequence the "activation" of another attack. It remains the open question: *Can a subsumption relation attack another argument?* From

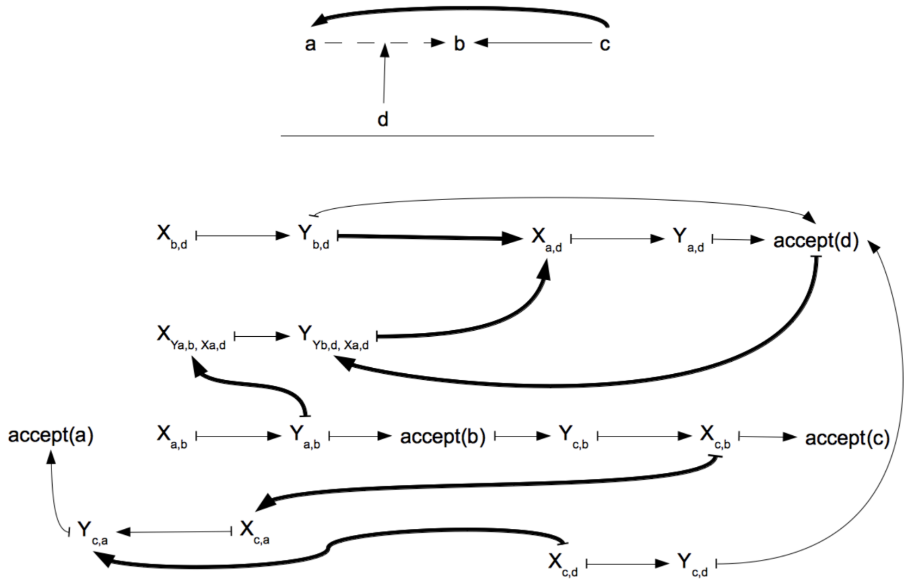


Figure 4.7: Example of attack on the subsumption relation in the meta model ($c \rightarrow b$).

our point of view, it would be possible. This additional kind of attack will change all the cases considered above. For example, if argument a is subsumed by argument b and the subsumption relation is attacked by argument d, the attack from the subsumption relation to argument c does not hold thus the extension would be $\{a, b, c, d\}$. The intuition behind this solution of this problem seems to be in the addition of support arguments, just like what we have for attack relations and agents and to model the attack on the subsumption relation as a direct attack on that kind of argument. The analysis and representation of this kind of attack in meta-argumentation is left for future research.

4.2 Toulmin scheme

Loui [Lou07] finds that Toulmin is ninth in total number of citations for philosophers of science and logic between 1988 and 2004. He concludes that after paradigm shifts and methods (Kuhn, Lakatos, Feyerabend), fuzzy logic (Zadeh), illocutionary force (Austin), the analytic-synthetic distinction (Quine), supervenience (Putnam), deductive-nomological explanation (Hempel), Toulmin's scheme must be mentioned next, before, for example, Carnap, Church, Tarski and Russell-Whitehead. Hitchcock and Verhey [HV07b] explain Toulmin's scheme in Figure 4.8 as follows.

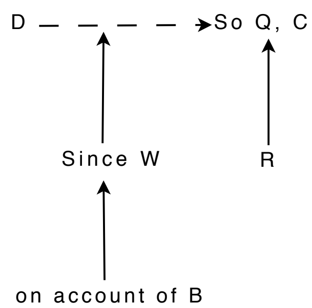


Figure 4.8: Toulmin scheme.

We have to understand the Toulmin structure in detail, and therefore consider the introduction to the recent book on the Toulmin model [HV07a]:

> "During this process of rational justification, we throw up what Toulmin called 'micro-arguments' [Tou58], for which he proposed a field-invariant pattern of analysis designed to do justice to the process of defending a particular claim against a challenger. This pattern, which has become to be known as the 'Toulmin model' or 'Toulmin scheme',

> differed radically from the traditional logical analysis of a micro-argument into premises and conclusion. "First we assert something, and thus make a *claim*. Challenged to defend out claim by a questioner who asks, "What have you got to go on?", we appeal to the relevant facts at our disposal, which Toulmin calls our *data* (D). [...] For the challenger may ask about the bearing of our data on our claim: "How did you get there?" Our response will be at our most perspicuous take the form: "Data such as D entitle one to draw conclusions, or make claims, such as C." [Tou58, p.98]. A proposition of this form Toulmin calls a *warrant* (W). Warrants, he notes, confer different degrees of force on the conclusions that justify, which may be signaled by qualifying our conclusion with a *qualifier* (Q) such as "necessarily", "probably" or "presumably". In the latter case, we may need to mention conditions of *rebuttal* (R) "indicating circumstances in which the authority of the warrant would have to be set aside" [Tou58, p.110]. Our task, however, is still not necessarily finished. For our challenger may question the general acceptability of our warrant: "Why do you think that?" Toulmin calls our answer to this question our *backing* (B)."

Different kinds of bakings are due to different fields while while warrants can be defended by appeal to a system of taxonomic classification, to a statute, to statistics from census, and so forth. All micro-arguments depend of the combination of data (D) and backing (B) and only in rare cases, checking the backing involves checking the claim.

There are various challenges in formalizing Toulmin's field-independent scheme. First, the scheme, like Dung's argumentation theory, differs radically from the traditional logical analysis of a micro-argument into premises and conclusion. However, Toulmin's argument against formalization of his scheme can be countered by the argument that over the past five decades, many new kinds of formalisms have been developed. The second challenge is that there are great differences between kind of backings in different fields, as emphasized by Toulmin, and thus backing B is abstract like arguments in Dung's theory. The third challenge is to represent the defense of C by D. Extensions of Dung's theory with a binary support relation among arguments [CLS05] do not allow for the support itself to be attacked, which is the core of Toulmin's scheme. The fourth challenge consists in providing a representation of the qualifier Q in abstract Dung's argumentation theory, analyzing its role differently from what is claimed by Verheij in [HV07a] who says "A qualifier is simply thought of as some kind of modal operator on statements. As a result, Toulmin's qualifier will be considered as being a part of the sentence that expresses the claim supported by the data."

We propose to represent argument D which supports the claim C with the warrant W by D is subsumed by C, where the absence of a warrant is equal to an attack on the subsumption relation. Figure 4.9 proposes a graphical representation of this approach. We do not represent the qualifier Q and the baking B can be represented as another support relation, from B to W, expressed by a subsumption relation. The subsumption relation, as proposed above, is represented by means of an attack from $X_{d,c}$ to $Y_{d,c}$ and an attack from $Y_{d,c}$ and C. The warrant is represented in this way by meta-argument $X_{d,c}$ which is the argument characterizing the subsumption relation. Rebuttals are modeled as standard attacks on the claim. In the meta-argumentation model, this is translated as an attack from meta-argument R to meta argument C. Following principle 2 above, the attack from meta-argument R to meta-argument C, characterized by meta-argument $Y_{r,c}$, actives an attack on meta-argument D from meta-argument R.

To further illustrate our solution to represent the Toulmin scheme,

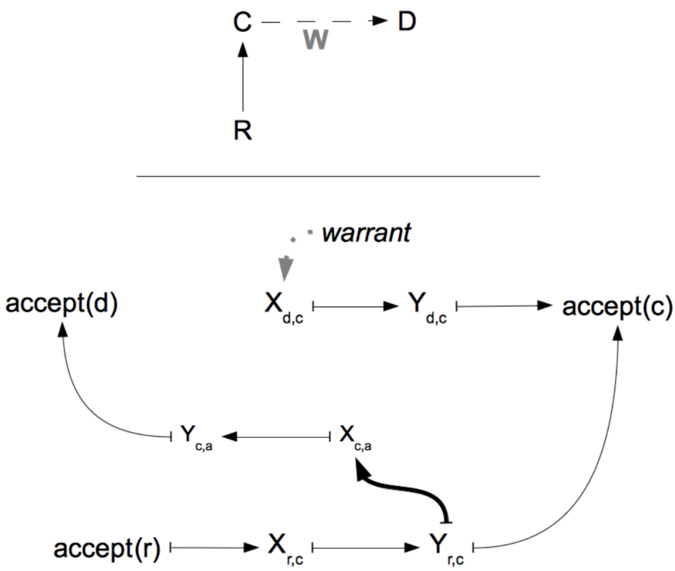


Figure 4.9: Modelling Toulmin scheme using subsumption relation.

we compare it with another representation using attack relations only, presented by Boella *et al.* [BvdTV09e]. Figure 4.10 visualizes this representation of Toulmin's scheme in abstract argumentation. Each square is a meta-argument, stating that the argument inside the square is accepted, and each circle is a meta-argument stating that the argument written inside is not accepted (neither is undecided). The qualifier Q is not represented, and rebuttal is represented by an optional counterargument R to C. If we have $\mathrm{D}_{\in}$ and $\mathrm{B}_{\in}$ then we have $\mathrm{W}_{\in}$ and accordingly $\mathrm{C}_{\in}$ for any of Dung's argumentation semantics. If we do not have $\mathrm{B}_{\in}$, then we don't have $\mathrm{W}_{\in}$, and consequently we don't have $\mathrm{C}_{\in}$. In the bottom left corner of the figure, a more convenient visualization is suggested. $\mathrm{C}_{\notin}$ and $\mathrm{R}_{\notin}$ are added for symmetry and to combine micro-arguments, but for a single micro-argument they could have been left out.

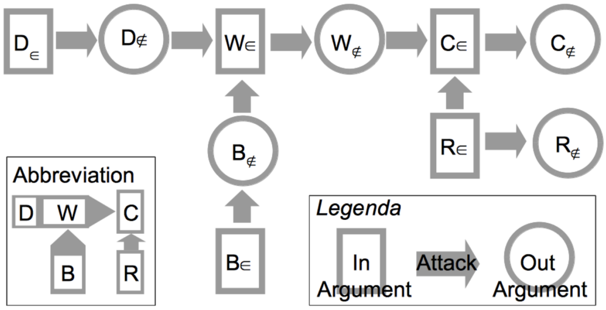


Figure 4.10: Modelling Toulmin scheme using attack relations only.

One may wonder whether there are other representations of Toulmin's scheme in the meta argumentation framework. For example, at first sight it may seem that if there is an attack from $D_\in$ to $D_\notin$, then there might also be an attack the other way around. However, this would not represent the defense of C by D, but a conditional defense: if D would be acceptable, then C would be acceptable too. However, we do not claim that our representation is the only one which can be used, and a more systematic exploration of the kind of schemes which can be represented in our meta-argumentation theory is a topic of further investigation.

The generation of meta-arguments and the condition on meta- argumentation frameworks are thus very simple, and formalized as follows.

Definition 19 *Let A_0 be a set of atomic arguments. Let the universe of arguments U of a Toulmin argumentation framework be the minimal set of arguments such that if a in A_0, then $a_\in$ and $a_\notin$ in U. A Toulmin argumentation framework is an argumentation framework $\langle A, \rightarrow \rangle$, where $a_\in$ in A iff $a_\notin$ in A, and if $a_\in$ in A, then $a_\in \rightarrow a_\notin$, and this is the only attack on $a_\notin$.*

Toulmin does not consider examples with cycles, so the formalization of his examples is straightforward (and all semantics coincide). A topic for further exploration concerning the effect of adding the meta-arguments is: how are the original semantics related to the semantics of the extended framework? How to define the influence from meta-arguments to original arguments?

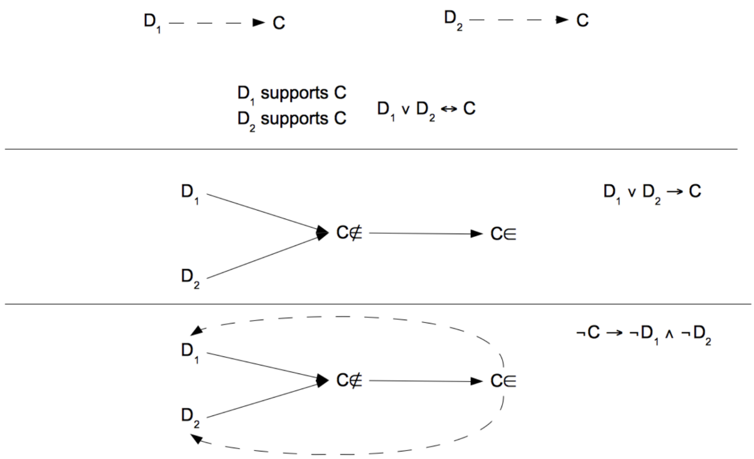


Figure 4.11: Toulmin scheme and implications.

The advantages of the new scheme can be summarized as follows. In the earlier approach, support is represented by means of an attack from argument D to the support relation, W, and an attack from the support relation to argument C. This means that if we have D then we have C, unless we have an attack on the support relation, W. The problem is that it seems only to express a week notion of support, since if we have that D supports C and R attacks C. The effect is that the extension $\{D, R\}$ does not satisfy that D supports C, though there is no attack on the support relation. Thanks to the new approach using the subsumption relation in order to express the support relation, we have that D is subsumed by C and if there is an attack from R to C, this is translated in an attack also on D, returning the extension $\{R\}$. This advantage is achieved thanks to the use of the meta argumentation model, in which support is represented using the subsumption relation.

Some other consideration may hold concerning Toulmin scheme. Let us discuss the examples provided by Figure 4.11 in which a simplified Toulmin scheme is presented. If argument D_1 supports argument C and also argument D_2 supports C, the following formula holds $D_1 \wedge D_2 \leftrightarrow C$. This formula gives us the following formulas: $D_1 \wedge D_2 \rightarrow C$ and $\neg C \rightarrow \neg D_1 \vee D_2$. We represent these two formulas by means of the earlier representation. The second formula makes clear what kind of relation holds between C and D_1, D_2, explicating the subsumption relation between them (dashed arrows). This subsumption relation states that if argument C is not accepted then also the two arguments D_1 and D_2, subsumed by it, cannot be accepted too.

Chapter 5

Merging argumentation networks

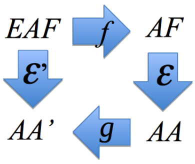
In this chapter, we answer the research question: *How to model merging of different abstract Dung's argumentation frameworks using meta-argumentation?* We start from the work of Coste-Marquis *et al.* [CMDK$^+$07], which describes an approach for merging argumentation frameworks and we propose three ways in which different AFs could be merged using meta-argumentation. We first introduce agents in the meta-argumentation model, then we propose a representation of the non-attack relation and of the ignorance relation, focussing on a dialogue perspective. In Coste-Marquis *et al.* [CMDK$^+$07], the non-attack relation is defined implicitly as the complement of the universe of attack relations and the ignorance one. The authors represent the partial argumentation framework as a quadruple $\langle A, R, I, N \rangle$ where A is the set of arguments, R is the binary attack relation, I is the binary ignorance relation such that $R \cap I = \emptyset$ and N is the binary non-attack relation such that $N = (A \times A) \diagdown (R \cup I)$.

5.1 Three ways of modelling merging

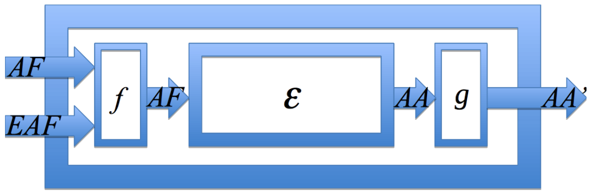
Merging different argumentation frameworks is particularly relevant in a multiagent perspective. Let us consider a multiagent system composed by two agents, 1 and 2. These two agents are associated to different argumentation frameworks, composed by different arguments and attack relations. In particular, agent 1 has the argumentation framework composed by $\langle \{a, b, c\}, \{a \rightarrow b, b \rightarrow c\} \rangle$ while agent 2 has the argumentation framework $\langle \{a, b\}, \{a \nrightarrow b\} \rangle$ where $\nrightarrow$ represents the non-attack relation. We define three different ways in which the merging of this two argumentation frameworks can be achieved. Our aim is not finding if one of these techniques would be better than the other ones but it is to present different ways of modelling the merging, in such a way to bring to different final results. The preference for one of these techniques is due to its application context. Moreover, these merging techniques provide a meaningful example of attacks between the X meta-arguments and the Y meta-arguments, justifying thus the introduction of these two kinds of meta-arguments in the model.

5.1.1 Technique 1: no agent meta-arguments

Figure 5.1 illustrates the meta-argumentation methodology introduced in Chapter 2. A flattening algorithm or function f flattens an extended argumentation framework EAF to an argumentation framework AF, and an argumentation semantics or acceptance function $\mathcal{E}$ gives the accepted arguments of the argumentation framework, and finally a function g gives the accepted arguments of the extended argumentation framework. Since the acceptance function $\mathcal{E}$ can be any of Dung's argumentation semantics, we have to define the extended argumentation framework EAF, the meta-argumentation framework AF, and the functions f and g.



(a) Categories



(b) Argumentation System

Figure 5.1: Extended argumentation framework as an instantiation: a function f transforms an extended argumentation framework AF to a basic argumentation framework AF. As in Figure 2.1, 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, $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$.

Technique 1 consists in a relatively simple kind of merging. Each agent is represented by a partial argumentation framework $\langle A, \rightarrow, \nrightarrow \rangle$ where both the non-attack relation and the attack relation are explicitly defined. The merging of these extended argumentation frameworks is done in the following way. The input of the function f is a sequence of partial argumentation frameworks $\langle \langle A_1, \rightarrow_1, \nrightarrow_1 \rangle, \ldots \rangle$, which are sets of arguments A_i with a binary attack relation $\rightarrow_i$ and a non-attack relation $\nrightarrow_i$.

Definition 20 *Given a set of arguments A and n agents, an extended argumentation framework EAF is a tuple $\langle A_1, \rightarrow_1, \nrightarrow_1, \ldots, A_n, \rightarrow_n, \nrightarrow_n \rangle$ where for each $1 \leq i \leq n$, $A_i \subseteq \mathcal{U}$ is a set of arguments and $\rightarrow, \nrightarrow_i \subseteq A_i \times A_i$ are two binary relations over A_i.*

The flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments $MA \subseteq MU$ is

$$\{accept(a) \mid a \in A_1 \cup \ldots \cup A_n\}$$

and the attack relation $\longmapsto \subseteq MA \times MA$ is a binary relation on MA such that $accept(a) \longmapsto accept(b)$ if and only if there is an agent $1 \leq i \leq n$ such that $a, b \in A_i$ and $a \rightarrow_i b$, and there is no agent $1 \leq j \leq n$ such that $a, b \in A_j$ and $a \nrightarrow_j b$.

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

Whereas there are agents in the extended argumentation theory, they are no longer present in the meta-argumentation theory. In other words, the merging has been done before the meta-argumentation theory is created, not by the meta-argumentation theory itself. This is illustrated by the following example.

Let us consider the example presented at the beginning of this section, represented in Figure 5.2.

Example 5 *Figure 5.2 presents an example of merging following technique 1. The EAF is as follows:*

$$\langle \{a, b, c\}, \{a \rightarrow_1 b, b \rightarrow_1 c\}, \emptyset, \{a, b\}, \emptyset, \{a \nrightarrow_2 b\} \rangle$$

The merging of the two partial argumentation frameworks is obtained with the union of the sets of arguments of both the partial argumentation frameworks and, according to Definition 20, by adding an at-

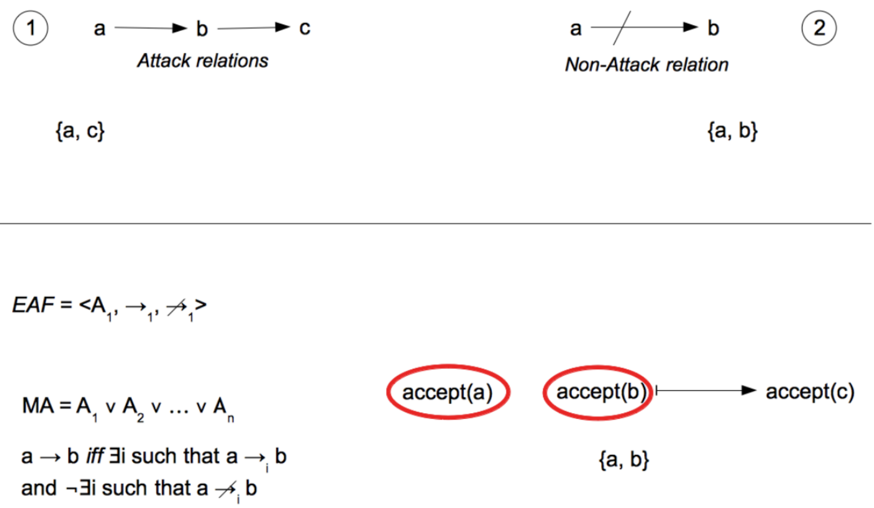


Figure 5.2: An example of merging with technique 1.

tack relation every time it is present in one of the partial argumentation frameworks and it is not present any non-attack relation between the same arguments. In the figure, the attack relation between b and c is added while the attack relation between a and b cannot be added due to the presence of the non-attack relation $a \nrightarrow b$ for agent 2. The unique extension of the meta-argumentation framework is $\mathcal{E}(f(EAF)) = \{accept(a), accept(b)\}$, thus the values returned by function g are $g(\{accept(a), accept(b)\}) = \{a, b\}$. The arguments inside a circle are the accepted ones.

The argumentation theory has to calculate the extension for arguments without agents only, but the flattening function is relatively complicated. This is due to the fact that the merger does not take the arguing of the individual agents into account, but only the structure of their argumentation framework. As an alternative, technique 2 incorporates also the argumentation of individual agents in the meta-argumentation framework.

5.1.2 Technique 2: agent meta-arguments

Technique 2 presents a merging modelling in which the arguing of each agent has an explicit representation in the merged model. We introduce meta-arguments $accept(a)$ for all arguments a that occur in the union of all the A_i, and meta-arguments $X_{a,b}$ and $Y_{a,b}$ for all arguments a and b occurring in the union of the A_i. Then we define that argument a attacks $X_{a,b}$ if and only if there is at least one agent i with $\rightarrow_i (a,b)$ and no agent j with $\nrightarrow_j (a,b)$. Moreover, we have that $X_{a,b}$ attacks $Y_{a,b}$ and $Y_{a,b}$ attacks $accept(b)$, see Chapter 3. The input of this technique is a sequence of partial argumentation frameworks EAF and the output is the set of extensions of acceptable arguments. Technique 2 is defined in the following way.

Definition 21 *Given a set of arguments, n agents and an extended argumentation framework* $EAF = \langle A_1, \rightarrow_1, \nrightarrow_1, \ldots, A_n, \rightarrow_n, \nrightarrow_n \rangle$*, see Definition 20, the flattening function* f *is given by* $f(EAF) = \langle MA, \longmapsto \rangle$*, where the set of meta-arguments* MA *is*

$$\{accept(i,a) \mid 1 \le i \le n, a \in A_i\} \cup \{X_{i,a,b}, Y_{i,a,b} \mid 1 \le i \le n, a, b \in A_i\} \cup$$

$$\{accept(a) \mid a \in A_1 \cup \ldots \cup A_n\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in A_1 \cup \ldots \cup A_n\}$$

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

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

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

$$Y(i,a,b) \longmapsto X_{a,b} \text{ and}$$

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

if and only if $a, b \in A_i$ *and* $a \rightarrow_i b$, *and* $X(i, a, b) \longmapsto Y_{a,b}$ *if and only if* $a, b \in A_i$ *and* $a \nrightarrow b$.

The unflattening function g *and the acceptance function* $\mathcal{E}'$ *are defined as in Definition 20.*

Let us consider the example presented in Figure 5.3.

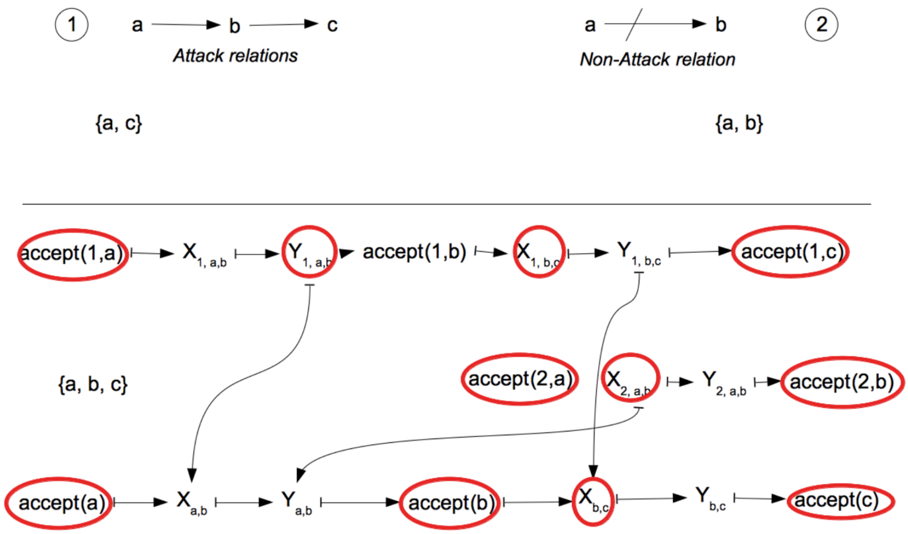


Figure 5.3: An example of merging with technique 2.

Example 6 *Figure 5.3, again starting with the previous EAF with agents 1 and 2, merges the partial argumentation frameworks of the agents using meta-arguments* $X_{a,b}$ *and* $Y_{a,b}$*, associating them to the agents. At the beginning, the merged argumentation framework is composed by the union of the arguments and of the attack relations of the EAF composing it. Then, each time an attack relation is present in the AF of the agent, the* Y *meta-argument of the attack relation attacks the* X *meta-argument of the same attack in the merged AF, e.g.,* *agent 1 attacks the $X_{a,b}$ with its $Y_{1,a,b}$. For each non-attack relation present in one of the partial argumentation frameworks, we add an attack from the X meta-argument of this non-attack to the attack relation, represented by argument Y, in the merged AF, e.g, $X_{2,a,b}$ attacks $Y_{a,b}$. Using this technique, we obtain that the unique extension of the meta-argumentation framework is $\mathcal{E}(f(EAF)) = \{accept(1,a), Y_{1,a,b}, X_{1,b,c}, accept(1,c), accept(2,a), X_{2,a,b}, accept(2,b), accept(a), accept(b), X_{b,c}, accept(c)\}$, thus the values returned by function g are $\mathcal{E}'(EAF) = g(\mathcal{E}(f(EAF))) = g(\{accept(a), accept(b), accept(c)\}) = \{a, b, c\}$.*

The differences between the two techniques are significant. Using technique 1, we accept arguments a and b while using the more complicated one we accept arguments a, b and c. The difference is that agent 1 does not accept argument b, and therefore also not does the attack of b to c, thus in the latter technique this has as result that argument c is accepted in the merged argumentation framework. Technique 2 explains where the attacks between the X meta-arguments and the Y ones hold. In this merging perspective, the attack of the kind a $X_{a,b} \to Y_{a,b} \to b$ means that the AF of the individual agent has a non-attack relation between arguments a and b, $a \nrightarrow b$, characterized by the $X_{a,b}$ meta-argument, and this meta-argument attacks the meta-argument $Y_{a,b}$, representing the same attack in the merged argumentation framework.

In our meta-argumentation theory, we introduce arguments such as, for example, *accept*(a). Accepting an argument is a natural notion in argumentation but we need to spend some words about the relation between accepting an argument and knowing an argument. We claim that, in order to accept an argument a, an agent should know argument a, otherwise he cannot accept it. For a further discussion on this issue, see [BGvdTV09a]. The following definition introduces the additional information that, in order to accept an argument, the agent needs to know it.

Definition 22 *Given a set of arguments, n agents and an extended argumentation framework $EAF = \langle A_1, \rightarrow_1, \nrightarrow_1, \ldots, A_n, \rightarrow_n, \nrightarrow_n \rangle$, see Definition 20, the flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments MA is*

$$\{know(i,a), Y_{i,know,accept,a}, accept(i,a) \mid 1 \le i \le n, a \in A_i\} \cup$$

$$\{X_{i,a,b}, Y_{i,a,b} \mid 1 \le i \le n, a, b \in A_i\} \cup$$

$$\{accept(a) \mid a \in A_1 \cup \ldots \cup A_n\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in A_1 \cup \ldots \cup A_n\}$$

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

$$know(i,a) \longmapsto Y_{i,know,accepts,a}, Y_{i,know,accepts,a} \longmapsto accept(i,a)$$

if and only if $a \in A_i$, and

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

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

$$Y(i,a,b) \longmapsto X_{a,b} \text{ and}$$

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

if and only if $a, b \in A_i$ and $a \rightarrow_i b$, and $X(i,a,b) \longmapsto Y_{a,b}$ if and only if $a, b \in A_i$ and $a \nrightarrow_i b$.

The unflattening function g and the acceptance function $\mathcal{E}'$ are defined as in Definition 20.

This new definition allows us to model a larger number of situations, particularly related to a multiagent system. Consider a multiagent system in which agents argue about other agents' arguments. For example, an agent could argue whether another agent accepts an argument, assuming that he knows that the other agent knows this argument. Or the agent could argue only that the other agent knows a particular argument but he does not accept it. In a more complicated scenario, agents can make coalitions and the choice to be part of a coalition or not depends on the arguments known by its members and not only on the accepted ones, although the latter ones depend on the former ones. Also in a game scenario, for example the contract bridge one, if you look at the possible moves as arguments, the agent which knows a particular move can also accept it (playing those cards), otherwise he cannot, and he can also have an idea about the opponents' moves, depending on what they know.

5.1.3 Technique 3: trust arguments

The third technique is more similar to the second one since it also considers the introduction of the agents in the argumentation framework. The difference consists in how agents are introduced. In this technique, agents are added as explicit arguments in the merged argumentation framework, attacking the attack relations of the merged AF due to the aim of expressing attack or non-attack. The input of this technique is again a sequence of partial argumentation frameworks EAF. We introduce meta-arguments "*agent i is trustable*" for all the agents i. Then we add meta-arguments "*argument a is accepted*" for all arguments that occur in the union of all the A_i, and meta-arguments $X_{a,b}$, $Y_{a,b}$ for all arguments a and b occurring in the union of the A_i. Then for each agent i, if EAF_i contains $a \rightarrow b$, then the meta-argument "*agent i is trustable*" supports the meta-argument $Y_{a,b}$, representing the attack relation, by attacking the meta-argument Z. This new meta-argument has the aim to represent a kind of support provided by the agent to its attacks and arguments. Otherwise, if EAF_i contains $a \nrightarrow b$, then the meta-argument "*agent i is trustable*" attacks directly the meta-argument $Y_{a,b}$. Moreover, each argument believed by the agents is supported, by means of the Z meta-argument, by the meta-argument $trust(i)$ when the argument is in A_i. In particular, the third technique is defined as follows:

Definition 23 *Given a set of arguments, n agents and an extended argumentation framework $EAF = \langle A_1, \rightarrow_1, \not\rightarrow_1, \ldots, A_n, \rightarrow_n, \not\rightarrow_n \rangle$, see Definition 20, the flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments MA is*

$$\{trust(i) \mid 1 \leq i \leq n\} \cup$$

$$\{accept(a) \mid a \in A_1 \cup \ldots \cup A_n\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in A_1 \cup \ldots \cup A_n\}$$

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

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

$$trust(i) \longmapsto Z, Z \longmapsto Y_{a,b}$$

if and only if $a, b \in A_i$ and $a \rightarrow_i b$, and moreover

$$trust(i) \longmapsto Y_{a,b}$$

if and only if $a, b \in A_i$ and $a \not\rightarrow b$, and finally

$$trust(i) \longmapsto Z, Z \longmapsto accept(a)$$

if and only if $a \in A_i$.

The unflattening function g and the acceptance function $\mathcal{E}'$ are defined as in Definition 20.

The idea is that there are two dimensions to which this merging technique can be viewed. The first dimension consists in the arguments and the attack relations of each agent. These elements are combined in a unique framework in which for each attack relation in a private AF the same attack relation is added in the merged AF. The second dimension consists in the expression of the agents' trust. Each agent

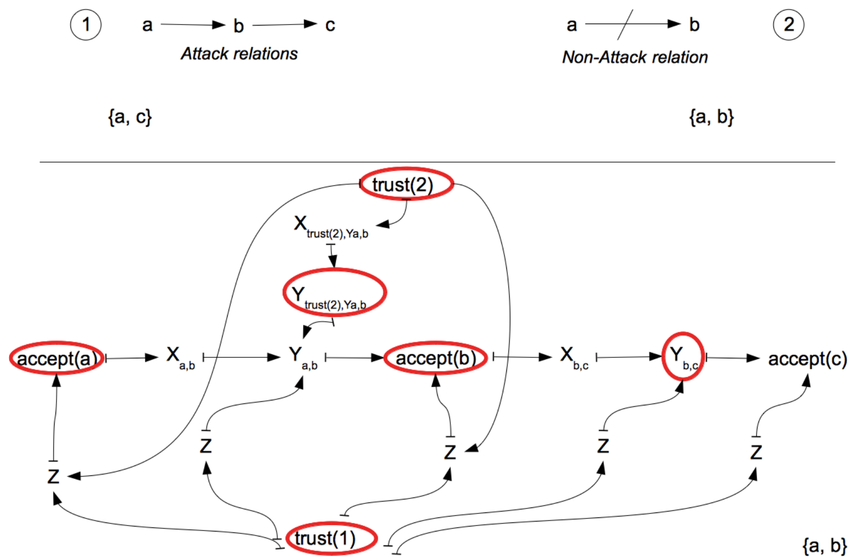


Figure 5.4: An example of merging with technique 3.

supports, by means of the Z meta-argument, or attacks, by means of the usual X and Y meta-arguments, the attack relations or the non-attack relations he believes. These additional arguments have the aim of represent the different degree of trust each agent has regarding the other agents. Let us consider the example presented in Figure 5.4.

Example 7 *Figure 5.4 presents how to model with this technique the previous example. In this case, each non-attack relation is explicitly represented and it consists in an attack relation attacked by the meta-argument representing the agent who has the non-attack relation. For example, agent 1 has two attack relations, $a \rightarrow b$ and $b \rightarrow c$. The merging is done putting in the new framework both these* *attacks and adding agent 1 under form of the meta-argument "agent 1 is trustable". This argument supports the two Y meta-arguments, representing the attack relations he has, through the meta-argument Z. Agent 2, also added in the merged argumentation framework as meta-argument "agent 2 is trustable", attacks meta-argument $Y_{a,b}$ which means that agent 2 has a non-attack relation between a and b. We obtain that the unique extension of the meta-argumentation framework is* $\mathcal{E}(f(EAF)) = \{accept(a), accept(b), trust(1), trust(2), Y_{b,c}, Y_{trust(2),Y_{a,b}}\}$*, thus the values returned by function g are* $\mathcal{E}'(EAF) = g(\{accept(a), accept(b), trust(1), trust(2), Y_{b,c}, Y_{trust(2),Y_{a,b}}\}) = \{a, b\}$.

Merging different argumentation frameworks means to deal with three kinds of situations: an attack relation, a non-attack relation and an ignorance relation. In the following example, we merge three argumentation frameworks from three distinct agents highlighting how to model these situations together using technique 3.

Example 8 *Let us consider the case in which agent 1 has an argumentation framework composed by $a \rightarrow b$, agent 2 has an argumentation framework composed by $a \nrightarrow b$ and agent 3 has an argumentation framwork composed by a and b ignoring what is the relation between the two arguments. These three AFs are depicted in Figure 5.5. Using technique 3, we merge them in the following way: agent 1, represented by the meta-argument $trust(1)$, supports the attack relation $a \rightarrow b$ and the two arguments a and b by means of the Z meta-argument, agent 2 attacks the attack relation between a and b because he knows that this attack relation does not hold but, at the same time, he supports the two arguments, agent 3 has no idea on the relationship between arguments a and b but he knows that these arguments exist thus he supports them by meta-argument Z. The extension of the merged meta-argumentation framework is* $\mathcal{E}(f(EAF)) = \{accept(a), accept(b), trust(1), trust(2), trust(3), Y_{trust(2),Y_{a,b}}\}$ *, thus the* *values returned by function g are* $\mathcal{E}'(EAF) = g(\{accept(a), accept(b), trust(1), trust(2), trust(3), Y_{trust(2),Y_{a,b}}\}) = \{a, b\}$.

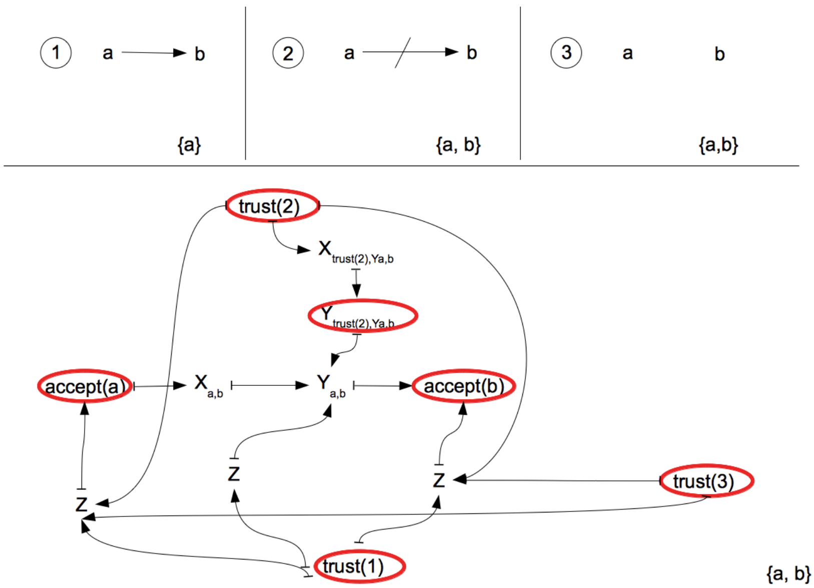


Figure 5.5: An example of merging with technique 3 attack, non-attack and ignorance.

The three techniques we presented in this section define different ways in which the merging of abstract argumentation frameworks could be done. As stated before, we are not interested in analyzing the benefits or the problems of one of these techniques in comparison with the others. In the context of modeling, we aim in providing various ways to merge argumentation frameworks, depending on what you desire to model, e.g., trust, knowledge. Moreover, these merging techniques are useful in order to explain what attacks such as

$X \to Y$ and $Y \to X$ mean. In particular, attacks like $X \to Y$ are used in the context of the non-attack relation, represented by the X meta-argument, attacking the attack meta-argument Y in the merged framework while attacks like $Y \to X$ are used to "confirm" an attack from the individual AF to the merged one. Different contexts of application could necessitate different models for merging. A possible future application and case study for doing some kind of comparison between these models is the coalition formation topic. From an epistemological point of view, merging the EAFs belonging to the members of a coalition could lead to a more stable coalition, decreasing the members' desire of breaking off the coalition. Thus, different ways of modeling the merging could change the degree of stability of the coalition. The ignorance relation is represented in an implicit way and the non-attack relation is represented in an explicit way in the extended argumentation framework. In particular, in the second and third techniques, the non-attack relation is seen as an attack from a meta-argument to the attack relation. The reasons behind this way of representing non-attack and ignorance come from a dialogue-based perspective which can be used to look at merging of AFs and it is discussed in Section 5.4.

5.2 Applications

5.2.1 Technique 1

Technique 1 should be used in order to not "impose" each agent's beliefs, represented by personal argumentation frameworks. The reason is that the merged argumentation framework which is returned thanks to technique 1 only involves attacks which are explicitly (presence of an attack relation) or implicitly (absence of a non-attack relation) accepted by the whole set of agents. Technique 1 seems to be, anyway, much more rigid than the other two techniques and thus less able in modeling different AFs together. It must be clarified that, although we start with different private argumentation frameworks, one for each agent, then, after the merging process we forget these private frameworks, returning only the merged one in which the single ones are no more identifiable.

5.2.2 Technique 2

The argumentation networks can be seen as graphs. An important topic in graph theory as well as in argumentation is the so called cycle analysis. Two kinds of cycles can be highlighted, odd cycles and even cycles. A cycle with an even number of vertices is called an even cycle; a cycle with an odd number of vertices is called an odd cycle. The odd/even cycle dilemma is a well known problem in argumentation theory. Baroni and Giacomin [BG03] observe that preferred semantics gives rise to counterintuitive results in cases related to the presence of odd-length cycles in the attack relation between arguments. To solve these problems, they propose a new semantics which preserves the desirable properties of the preferred semantics, while correctly dealing with odd-length cycles. In [BGG05], the authors introduce the notion of SCC-recursiveness, based on the graph-theoretic notion of strongly connected components. The definition of a SCC-recursive semantics has a straightforward constructive interpretation: it suggests an effective recursive procedure for computing all the extensions of an argumentation framework, according to a given SCC-recursive semantics, once a specific base function is assigned, and an important role in this context is played by the initial strongly connected components. The authors suggest that odd and even cycles should be treated in the same way. Another point of view about the odd/even cycle dilemma is given by [DBC02] where only the absence of odd cycles ensures that the system is coherent.

Cycle analysis is strictly related to technique 2. In particular, technique 2 answers the question: *Under which conditions do we have only even cycles for 2-player games?* The intuition behind this answer is as follows: in the merged argumentation framework obtained following technique 2, each agent finds some of the attacks he supports. But the agent does not support any attack which attacks arguments he does accept. Let us explain this intuition using an example presented in Figure 5.6.

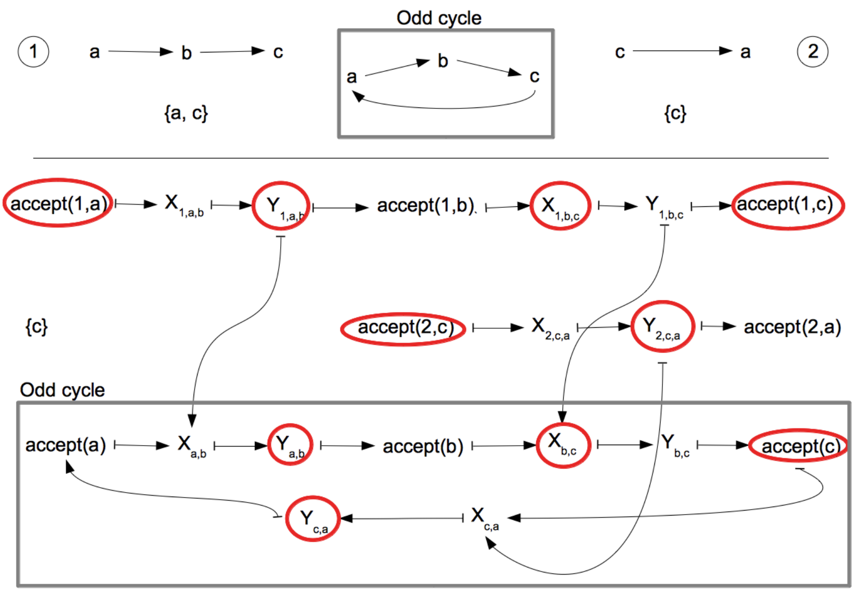


Figure 5.6: Example of cycle analysis using technique 2.

In this example, we have two players 1 and 2 with their argumentation frameworks. If we put the two private argumentation networks together making simply the union between the two argumentation frameworks, the result is a network with an odd cycle. Each attack relation in the merged argumentation framework is supported by the agent who has this attack in his private argumentation network. This additional attack, representing the support on the attacks held by the agent, allows to give a unique extension to the merged argumentation framework, which is an odd cycle, as highlighted by Figure 5.6. Using technique 2, merging two argumentation frameworks a graph with only even cycles is obtained. The following propositions holds for this technique:

Lemma 1 *For any attack $a \rightarrow b$ in the merged argumentation framework there must be at least one agent i who accepts argument a and who has $a_i \rightarrow b_i$.*

Lemma 2 *Arguments which are not accepted by any agent do not attack other arguments in the merged argumentation framework.*

Theorem 3 *For every 2-player game only even cycles are allowed using Technique 2.*

This technique can thus be used in order to solve, to find a unique extension for argumentation networks involving an odd cycle. This can be achieved by dividing the argumentation network into two components, as done in Figure 5.6, and then putting them together in the merged argumentation framework. This argumentation framework is not devoid of odd cycles, as underlined by Figure 5.6, of the starting argumentation network but, in this case, to the odd cycle we can associate a unique extension, thanks to the attacks from the two additional parts of the meta-argumentation framework representing the private argumentation frameworks. Due to the absence of odd cycles, the following postulate holds in the merged argumentation framework: *For every argumentation framework built using technique 2, there exist stable extensions.* This postulate opens to other constraints on the merged argumentation framework, concerning for instance the relationship between preferred and stable extensions, *the preferred and stable semantics coincide for the merged argumentation framework*, and the undecided labels, *there are no undecided arguments*. A deeper analysis of these postulates is necessary in order to establish the real benefits of the proposed constraints. This analysis and the proof is left for future work.

5.2.3 Technique 3

In real life, the notion of merging can be intuitively reduced to sharing personal beliefs and adopting others' beliefs. One of the main constraints that people apply before sharing and adopting their personal beliefs with other people is trust. Trust is a key issue both in societies and now also in multiagent systems. Roughly, trust basically is a mental state, a complex attitude of an agent x towards another agent y about the behaviour or action a relevant for the goal g. The mental ingredients of trust are thus its specific beliefs and goals, with special attention to evaluations and expectations [FC05].

Merging argumentation frameworks can be seen as merging agents' personal beliefs thus trust should be discussed also in this context. In particular, here we mean trust about sources. Intuitively, an agent a who cannot trust another agent b will choose if not to adopt b's arguments and/or b's attacks thus he could attack agent b itself, some or all b's arguments or some or all b's attack relations. Let us apply technique 3 in order to represent trust in merging argumentation frameworks. Technique 3 introduces explicitly the agents into the argumentation framework, more precisely, agents like 1 and 2 in Figure 5.7 can be read as the argument "agent i is a reliable source". An approach related to trust in argumentation is provided by Hunter [Hun08] where the author introduces a logic-based meta-level argumentation framework for evaluating arguments in terms of the appropriateness of their proponents.

In Figure 5.7, a first application of technique 3 is presented. The idea is as follows: if agent 2 does not trust agent 1, it can be represented following different degrees of distrust. First, an agent cannot trust another agent at all. This is represented in Figure 5.7 by the dotted attack from agent 2 to agent 1. This attack leads to the same extension, as shown in the legend of the figure, of the merged argumentation network without this attack since the two arguments a and b are also supported by agent 2 thus the attack on agent 1 does not make they not accepted. Second, agent 2 cannot trust agent 1 only about the attack relation $b \rightarrow c$ thus it attacks only the meta-argument $Y_{b,c}$, supported by agent 1 by means of meta-argument Z. This attack leads to the acceptation also of argument c in the extension of the merged argumentation framework. Third, agent 2 cannot trust agent 1 only about argument c thus it attacks this argument. Note that the attack from $trust(2)$ to $Y_{b,c}$ and $accept(c)$ are abbreviated due to clarity constraints of the figure but they involve the X and Y meta-arguments as follows: $trust(2) \rightarrow X_{trust(2),Y_{b,c}} \rightarrow Y_{trust(2),Y_{b,c}} \rightarrow Y_{b,c}$ and $trust(2) \rightarrow X_{trust(2),accept(c)} \rightarrow Y_{trust(2),accept(c)} \rightarrow accept(c)$, respectively.

Another way to represent the absence of trust regarding a source is presented in Figure 5.8. The depicted situation is as follows: agent 1 has a non-attack relation between arguments a and b but agent 2 does not trust him about this non-attack relation. This lacking of trust concerns only a specific topic used by agent 1 concerning the non-attack $a \nrightarrow b$, for example arguments coming from religion. Thus, the argument representing agent 2 in the merged argumentation network, $trust(2)$, blocks this attack by blocking the meta-argument $Y_{trust(1),Y_{a,b}}$ coming from agent 1 but it does not block in this way all the other attacks coming from agent 1.

The two examples above show how technique three could be used in order to merge different argumentation frameworks. Summariz-

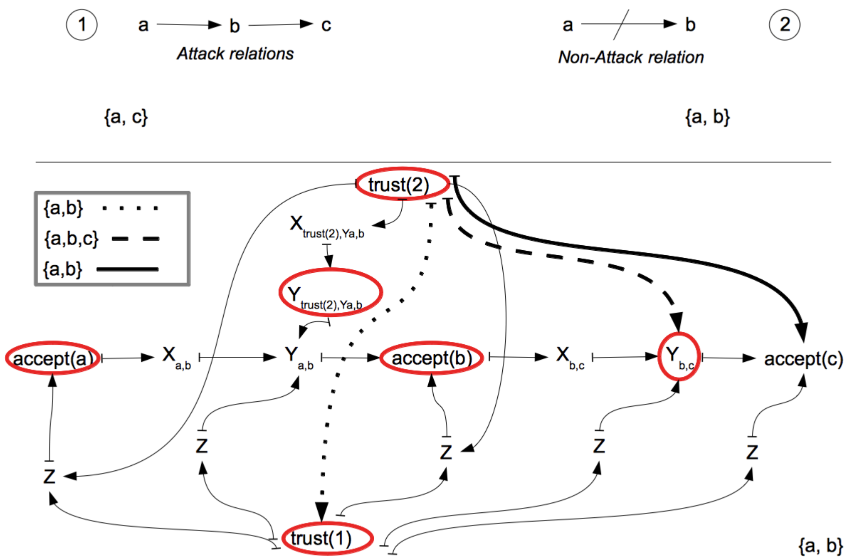


Figure 5.7: Example of merging with the three degrees of distrust.

ing, four kinds of distrust are considered in this section. First, an agent 1 does not trust anymore another agent 2 thus $trust(1)$ attacks $trust(2)$. In this way, all the attack relations, non-attack relations and arguments of agent 2 are attacked by agent 1. Note that, thanks to meta-argument Z which represents the support, if another agent 3 supports one of the attack relations or arguments of agent 2 then they could be accepted in the merged argumentation network. Second, agent 1 does not trust an argument of agent 2 thus $trust(1)$ attacks this argument by means of the X and Y meta-arguments. Third, agent 1 does not trust agent 2 about an attack relation thus $trust(1)$ attacks the Y meta-argument of this attack relation. Fourth, agent 1 does not trust agent 2 about a non-attack relation thus $trust(1)$ at-

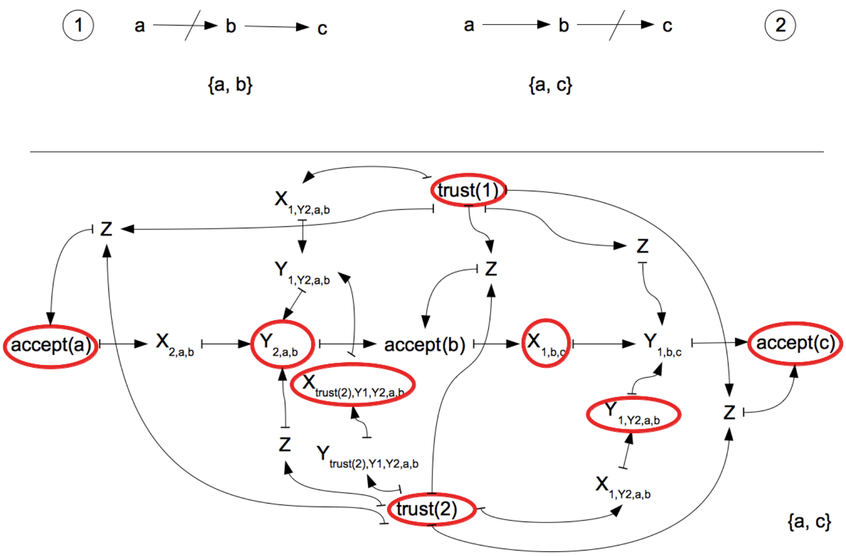


Figure 5.8: Example of merging with an agent attacking a non-attack relation of another agent.

tacks the Y meta-argument which represents the attack from $trust(2)$ to the attack relation it considers not valid.

5.3 Merging 2^{nd} order AFs

Second-order argumentation frameworks have been introduced by Modgil [Mod07] in the context of preference-based argumentation frameworks. Second-order argumentation means that the binary attack relations can be defined both over the set of arguments and the set of attack relations. Roughly, it is possible to have an attack from an argument to another attack relation and attacks between attack relations themselves. These kinds of second order attacks are used to express a preference over the arguments. In this section, we present how merging of second-order argumentation frameworks can be achieved. Let us consider the following example consisting of agents 1 and 2, and a set of attack relations.

Example 9 *Figure 5.9 presents two agents and their argumentation frameworks. Let us consider an example concerning insurances. The default assumption from which we start is that life insurance is a motive for murder. The meaning we give to the arguments is as follows: a is an argument like "the husband has a life insurance", b is an argument like "the wife did not kill her husband", c is an argument like "the wife loves her husband" and d is an argument like "there is another lover". The attacks relations are those presented in the figure. In order to merge the two argumentation frameworks, we put all the attacks and the arguments in the the merged argumentation framework. In order to model second-order attacks, we add three new attacks from the private argumentation frameworks of the agents to the merged one. These attacks are second-order attacks (bold arrows). These attacks express the preference on the attacks or, better, the power of activating the attacks in the merged argumentation framework. For example, agent 2 has the power to activate the attack from a to b, by attacking the attack raised by c. Note that agent 2 has this power even without being aware of these arguments. The extension, if agent 2 activates the attack, is $\{a, d\}$.*

Figure 5.10 presents the translation of the argumentation network of Figure 5.9 into the meta-argumentation model. The possibility to activate the attack is represented by means of second-order attacks from the Y meta-arguments representing the attack in the private argumentation framework to the X meta-arguments in the merged argumentation framework. For example, meta-argument $Y_{d,c,1}$ attacks *meta-argument $X_{d,c}$ in the merged argumentation framework.*

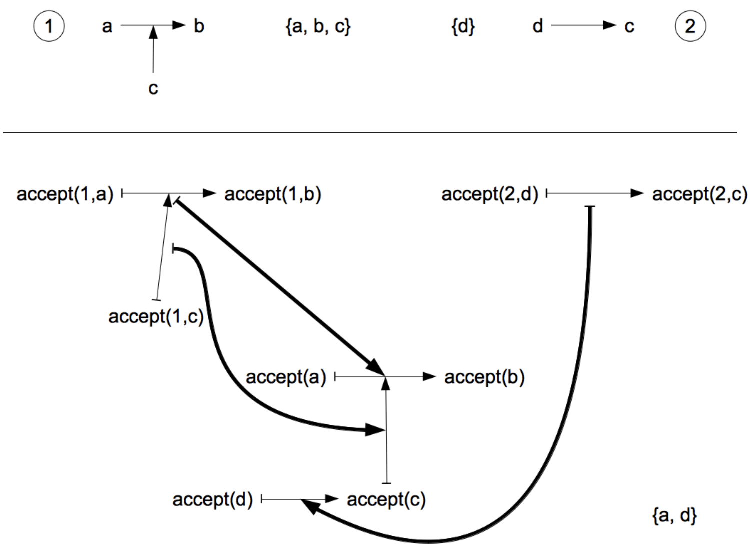


Figure 5.9: An example of merged 2^{nd} order AFs.

The input is a sequence of partial argumentation frameworks $EAF = \langle\langle A_1, \rightarrow_1, \nrightarrow_1\rangle, \ldots\rangle$, which are sets of arguments A_i with a binary attack $\rightarrow_i$ and a non-attack relation $\nrightarrow_i$. Then for each agent, we have an attack from $Y_{a,b,i}$ to $X_{a,b}$: if the agent wants to raise the attack from a to b in the merged argumentation framework then he activates this attack. Moreover, if there is $\nrightarrow (a, b, i)$ then we have an attack from $X_{a,b,i}$ to $Y_{a,b,i}$. Merging second-order argumentation frameworks is defined in the following way:

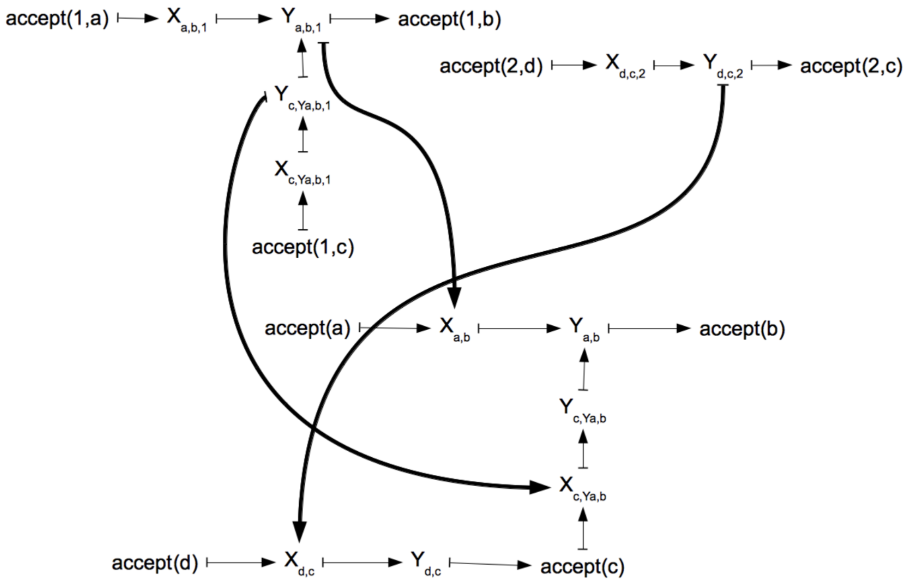


Figure 5.10: Merged 2^{nd} order AFs in meta-argumentation.

Definition 24 *Given a set of arguments and n agents, an extended argumentation framework with second-order attacks is a tuple $EAF = \langle A_1, \rightarrow_1, \nrightarrow_1, \rightarrow_1^2, \nrightarrow_1^2, \ldots, A_n, \rightarrow_n, \nrightarrow_n, \rightarrow_n^2, \nrightarrow_1^2, \rangle$, where A_i, $\rightarrow_i$ and $\nrightarrow_i$ are as in Definition 20, and $\nrightarrow_i, \rightarrow_i^2$ are binary relations on $A_i \cup \rightarrow_i \times \rightarrow_i$.*

The flattening function f is given by $f(EAF) = \langle MA, \longmapsto \rangle$, where the set of meta-arguments MA is

$$\{accept(i,a) \mid 1 \leq i \leq n, a \in A_i\} \cup \{X_{i,a,b}, Y_{i,a,b} \mid 1 \leq i \leq n, a, b \in A_i\} \cup$$

$$\{accept(a) \mid a \in A_1 \cup \ldots \cup A_n\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in A_1 \cup \ldots \cup A_n\}$$

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

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

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

$$Y(i,a,b) \longmapsto X_{a,b}$$

if and only if $a, b \in A_i$ *and* $a \rightarrow_i b$, *and*

$$X(i,a,b) \longmapsto Y_{a,b}$$

if and only if $a, b \in A_i$ *and* $a \nrightarrow b$.

The unflattening function g *and the acceptance function* $\mathcal{E}'$ *are defined as in Definition 20.*

The issue of how to model merging of second-order argumentation frameworks opens the discussion on the following challenges in merging. The notion of expansion does not impose many constraints on the merging, what is important is to preserve the attack and non-attack relations from the initial argumentation framework while extending the set of arguments of each agent. Many policies can be used to give rise to expansions of different kinds, reflecting the various attitudes of agents under the "new" arguments. Coste-Marquis *et al.* [CMDK+07] analyze the policy called consensual expansion. Intuitively, the consensual expansion of an argumentation system is obtained by adding a pair of arguments (a, b) (where at least one of a, b is not in $\mathcal{A}$) into the attack (respectively the non-attack relation) provided that all other agents of the profile who know the two arguments agree on the existence of the attack (respectively the non-attack); otherwise, it is added to the ignorance relation.

Our future aim is in analyzing, in addition, the following cases:

- Agent 1 proposes to agent 2 argument b but he does not claim any relation of b with argument a,

- Agent 1 proposes to agent 2 argument b against argument a, actually saying to him $a, a \rightarrow b$.

In these cases, we do not need the policies of expansion provided by [CMDK+07], but agent 1 proposes to agent 2 also the relation between the new arguments he is adding to agent 2's argumentation framework. As discussed at the beginning of this chapter, this perspective is useful for dialogue more than merging as proposed by Coste-Marquis *et al.* [CMDK+07].

5.4 Merging in dialogues

Another issue which has to be addressed is related to dialogue. Merging of different argumentation frameworks provides an external perspective of the argumentation frameworks but in dialogues the opposite holds since in dialogues an internal perspective of the private argumentation networks of the agents is analyzed. Merging means that we would know in advance what arguments and what attack relations are accepted by a group of agents. Each argumentation framework of the agents is taken, with all its components, and a partial argumentation framework, in the case of Coste-Marquis *et al.* approach [CMDK+07], consists in merging these argumentation frameworks in a single one.

Many considerations have to be made concerning the non-attack relation and the ignorance one. Our model allows us to represent non-attack in an explicit way, differently from what is done in [CMDK+07]. Moreover, we introduce the agents in the merging model by labeling, for example as shown by technique 2, the attack relations and the non-attack ones with the agent which holds these relations. The ignorance relation is represented in an implicit way. In this case, the absence of indexes representing agents means ignorance of these agents concerning the attack relations and the meta-arguments.

Let us consider the example depicted in Figure 5.11.

Example 10 *The example of Figure 5.11 presents three agents as actors of a simple ecological dialogue. The dialogue involves three ar-*

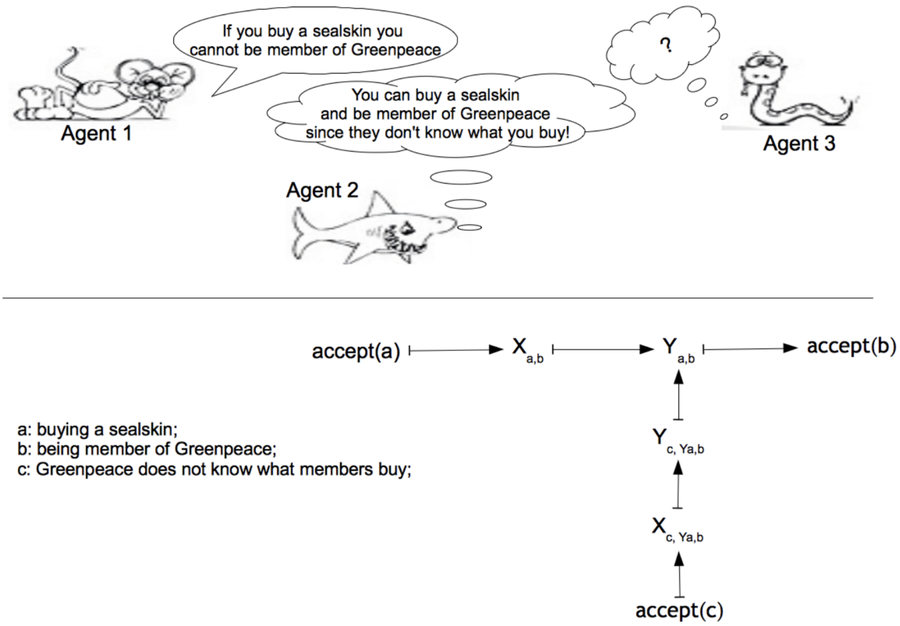


Figure 5.11: An example of introduction of agents in argumentation.

guments which are as follows: a: buying a sealskin, b: being a member of Greenpeace and c: Greenpeace does not know what members buy. Agent 1 has both argument a and argument b and it has also the attack relation $a \to b$. Agent 2 has arguments a, b, c and it has an attack relation between argument c and $a \to b$. Agent 3 has a complete ignorance concerning these issues thus he does not know anything.

However, in the dialogue perspective, if one of the interactants believes an argument a does not attack another argument b, it means that he has some reasons for believing that and these reasons are explicitly defined as other arguments attacking the attack relation between a and b. Let us consider the following example again based on a dialogue of ecological nature:

- A: I bought a fur in the Marnie's shop
- B: Buying a fur means you do not love animals
- C: But furs in the Marnie's shop are made only with animals died a natural death

This dialogue shows three arguments in which argument a attacks argument b but argument c attacks this attack. For instance, we have two agents and the following argumentation framworks:

- $\langle A_1 = \{a, b\}, \rightarrow_1 = \{(a, b)\}\rangle$
- $\langle A_2 = \{a, c\}, \rightarrow_2 = \{(c, (a, b))\}\rangle$

The non-attack relation is explicitly represented as an attack to the meta-argument representing the attack from a to b, such as $Y_{a,b}$, as shown in Figure 5.12. The non attack relation is represented by means of the attack $c \rightarrow (a \rightarrow b)$. Thus, for agent 1 there is an attack relation between arguments a and b while for agent 2 this attack relation does not exist since it knows argument c, attacking the attack $a \rightarrow b$.

This way of modeling the non-attack relation aims at being true to life due to the dialogue nature of argumentation theory. The meta-argumentation model gives us additional possibilities, such as arguing about the attack relations at the object level. Note that since argument b is not involved in the argumentation framework of agent 2 and, by definition, the meta-argument $Y_{a,b}$ attacks b as well. In this case, we have an argumentation framework composed by arguments a and c and by the attack $a \rightarrow b$, without requiring to have the meta-argument "b is accepted". The attack relation in the object level is modeled by the attack arguments $X_{a,b}$ and $Y_{a,b}$ and we can argue about that.

Let us consider another example, represented in Figure 5.13.

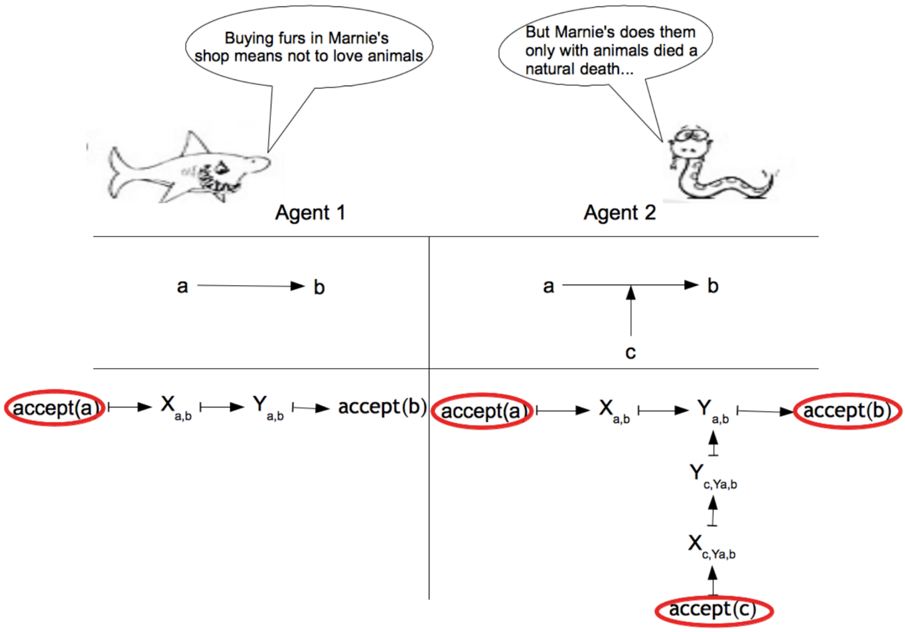


Figure 5.12: Graphical representation of the above dialogue.

Example 11 *Figure 5.13 depicts two agents, 1 and 2, and their argumentation frameworks. We merge the two AFs using technique 2. Agent 1 has arguments a, b, c and $a \rightarrow b$ and $b \rightarrow c$ while agent 2 has arguments a, d and $a \rightarrow b$, $b \nrightarrow c$. The lower part of Figure 5.13 describes the merged argumentation framework resulting from the two argumentation frameworks of agents 1 and 2. We index each argument and meta-argument with the label representing the agent having it. The extension of the EAF_1 is $\{a, c\}$ as for EAF_2. The same holds for the merged AF.*

A second issue which needs to be analyzed in our modelling technique is the one of ignorance. In Coste-Marquis *et al.* [CMDK+07], as stated, this is an explicit relation, $R \cap I = \emptyset$. In our case, we can

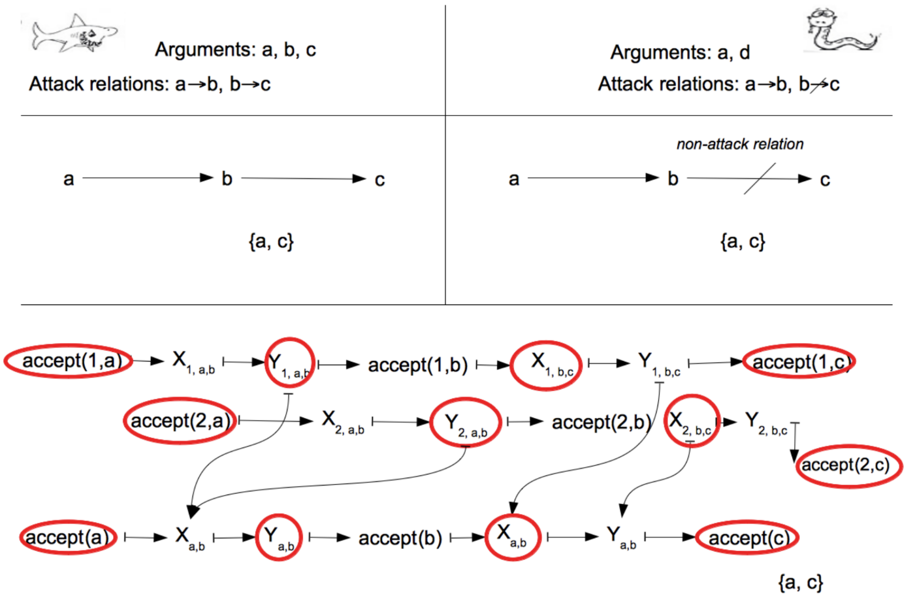


Figure 5.13: An example of merging with a non-attack relation.

model ignorance in an implicit way. This means that an interactant ignores the relation between arguments a and b if there are no meta-arguments of the kind $X_{a,b}$, $Y_{a,b}$ and $X_{b,a}$, $Y_{b,a}$. This for Coste-Marquis *et al.* [CMDK$^+$07] means non-attack.

A particular kind of situation is the following one. Agent 2 has this set of arguments and meta-arguments: a, c and $c \rightarrow (a \rightarrow b)$ but argument c is not acceptable. In this case, what are we representing? We have the following possibilities: we are representing a non-attack relation which means that for agent 2 the attack between a and b does not hold or we are representing an ignorance relation since agent 2 does not know that the attack $(a \rightarrow b)$ does not hold since c is not acceptable. In some way, this could be defined as a kind of *ignorance on a non-attack* in which the agent believes that an attack does not exist due to another argument ignoring that this argument is no more acceptable.

5.5 Discussion on limitations of abstract merging

In this section, a discussion about the limitations of abstract merging is addressed in order to provide a comparison between this approach and merging with structured and instantiated argument. In this kind of approach, arguments are defined as inference trees formed by applying two kinds of inference rules: strict and defeasible rules. This leads to three ways of attacking an argument: attacking a premise, attacking a conclusion or attacking an inference. The attack relation is derived from the arguments themselves. A first kind of criticism addressed to the abstract argumentation theory consists in the fact that with abstract arguments there is no way to derive the attack relations, on the contrary of instantiated arguments. In our view, this is an important open problem regarding abstract argumentation theory.

Another example of merging using abstract arguments is presented by Caminada and Pigozzi [CP09]. The key notion of the paper is that any individual agent has to be able to defend the collective decision and this is guaranteed when the group outcome is compatible with its members views. They use an argumentation approach to judgment aggregation problems. Given an argumentation framework, different individuals may provide different evaluations regarding what should be accepted and rejected. The aggregation of individual evaluations of a given argumentation framework raises the same problems as the aggregation of individual judgments. They show that argument-by-argument majority voting may result in an unacceptable extension, as the proposition-wise majority voting may output an inconsistent collective judgment set. Judgment aggregation is addressed as the problem of combining different individual evaluations of the situation represented by an argumentation framework. The authors motivate their use of abstract argumentation claiming that on the one hand, the existence of different argumentation semantics allows to be flexible when defining which social outcomes are permissible. On the other hand, it allows to bring judgment aggregation from classical logic to nonmonotonic reasoning. A particular counter argument in using structured arguments consists in the difficulty to assess incompatibility of arguments.

Chapter 6

Coalition Formation

A social network is a social structure composed by nodes, which are generally individuals or organizations, that are tied by one or more specific types of interdependency. Wide social networks and small ones share the same structure but different kinds of analysis are needed. The analysis of large social networks [HR05] is usually based on either data-mining or graph-based techniques, such as small world properties, centrality, cliques, similarity, and so on. These analysis tools work well for large networks, such as those composed by the nodes in the world wide web or the members of Facebook, but they work less well for small networks representing the relations among stakeholders in software engineering. Moreover, they do not support iterative design of software in order to interact with the designed system to provide a form of research for informing and evolving a project, as successive versions.

Small social networks are analyzed in software engineering, for example by the TROPOS methodology [BPG+04], developed for agent-oriented design of software systems. The intuition of the TROPOS methodology [BPG+04] is to couple the instruments offered by software engineering and the multiagent paradigm. In this paradigm, the entities composing the system are agents, autonomous by definition, characterized by their own sets of goals, capabilities and beliefs. The multiagent paradigm allows the cooperation among the agents with the aim to obtain common and personal goals. In this way, multiagent systems offer a solution for open, distributed and complex systems and the approach combining software engineering and multiagent systems is defined Agent-Oriented Software Engineering. A typical social dependence network in the TROPOS methodology [BPG$^+$04] contains at most a hundred nodes, in contrast to the hundreds of thousands of nodes used in the web or in Facebook.

In this chapter, we are interested in the analysis based on cooperation which emerges in 'small' social networks in order to achieve a greater number of goals. As a measure of cooperation, we analyze the coalitions [SK98] that emerge in a social network assuming reciprocity, for example measuring the number of coalitions [BvdTV09d], the kinds of coalitions [BvdTV08d], or the stability of the coalitions. This breaks down in the following questions: *How to iteratively design a social network?* and *How to analyze the reciprocity based coalitions that may emerge in social networks at various degrees of abstraction?* and *How to refine the abstract coalition models with social dependencies among agents, powers of sets of agents, and plans or tasks?*.

At the highest level of abstraction, coalitions are purely abstract and we only specify whether the creation of one coalition will block the creation of another coalition. We say that two coalitions are attacking each other if there is a source of incompatibility between them and the second-order argument sets a preference of the first coalition over the second one, and we use abstract argumentation theory [BCD07] to determine acceptable coalitions and second-order arguments. At the second level of abstraction, we detail the composition of a coalition as formed by a set of agents and a set of dependencies between them. Our notion of coalition is based on the concept of reciprocity which constraints each node to contribute something, and to get something out of it. For example, in a virtual organization each node has to be useful for at least another node. At the third level of abstraction, we detail the powers and goals of the individual agents. At the fourth level of abstraction, we also detail beliefs, decisions and goals of the agents. For the analysis we focus on the coalition view and dynamic dependence view, and we leave a detailed analysis of the power and agent views for further research.

We illustrate our approach using a grid scenario. Consider, for example, a virtual organization for e-Science composed by nodes belonging to academic institutions such as universities and research centers. Inside the virtual organization, sub-groups can be formed with the aim to collaborate in order to achieve a greater number of goals, i.e., if node a cannot store a file but it can help node b in doing a computation and b can store a's file, these two nodes form a reciprocity-based coalition in order to achieve both goals. It would be possible that two or more candidate coalitions share the same goals, e.g. two nodes can do the storage for node a and thus it becomes necessary to have a mechanism to decide what coalition can be formed.

Using social dependence networks to represent the multiagent system, as in TROPOS [BPG+04], allows us to model, particularly for the requirements analysis phase of the design process, the domain stakeholders. The analysis of cooperation in this context is relevant since agents can form coalitions with the aim to achieve more goals than what they can achieve alone. As in well known game theoretic approaches to cooperation [SK98], we face with problems of incompatibilities between the possible coalitions which can be formed together. We manage these incompatibilities using an argumentation framework treating each candidate coalition as an argument, the incompatibilities as the attacks between the arguments and, finally, using the extensions to find out the acceptable coalitions and second-order arguments.

6.1 Iterative social network design

In this section, we answer to the first research question of the chapter, presenting the four viewpoints composing our iterative design model and we describe the concepts we use in the model thanks to an ontology. Moreover, we provide a running scenario based on the grid architecture explaining our model of iterative design for small social networks.

6.1.1 Coalitions in a grid-based scenario

Cooperation in grid, in particular virtual organizations, can be seen as coalition formation in social networks. A virtual organization allows the users, their roles and the resources they can access in a collaborative project to be defined [SCD+08]. In particular, we look into small sets of nodes within virtual organizations as coalitions. Reciprocity-based coalitions can be viewed as subsets of a virtual organization, in which there is the constraint that each node has to contribute something, and to get something out of it.

The scenario of virtual organizations based on grid networks represents a case study able to underline the benefits of the presented viewpoints and the argumentation framework to argue about the evolution of coalitions over time. First of all, in the multiagent paradigm agents' autonomy is assumed in all representations, i.e., the grid philosophy imposes the autonomy of the nodes composing it. Second, the presented model depicts the system using dependence networks, structures similar to the grid network itself. Finally, the idea that subsets of nodes composing a virtual organization compose also different coalitions sharing common goals and attacking each others helps in providing the intuition of the addressed problem and the proposed solutions.

Concerning viewpoints, a virtual organization can be represented using our four views in order to highlight different aspects: the agent view presents each node of the grid as an agent with a set of associated skills and goals, the power view presents the nodes which can achieve the goals of the virtual organization and what are the nodes with the conditional power to add new goals to other nodes, the dynamic dependence view describes the virtual organization in terms of dependencies giving it a network structure and, finally, the coalition view presents the virtual organization as sets of nodes representing reciprocity-based groups. In this context, the modeled stakeholders are the nodes of the virtual organization and their concern is to store and run data.

6.1.2 Ontology

In this section, we introduce the ontology used in our model, represented as a UML diagram shown in Figure 6.1. This ontology summarizes the concepts introduced in our four views. Particularly, it introduces the concepts of agent, fact, skill and goal. Each agent has a set of facts in which it believes and a set of goals it has to achieve by means of its skills and these relations are represented by the agent view. Figure 6.1 presents two kinds of dependencies, common dependencies and dynamic dependencies. The first ones explain that an agent (*depender*) depends on another agent (*dependee*) to achieve a goal (*dependum*) while dynamic dependencies enable the addition or removal of dependencies by a third agent (*dyndep creator*). The notion of coalition, with its subclasses, is linked to both the concepts of common and dynamic dependency and agent since we define a coalition as a set of dependencies and agents. The preference of one coalition over the other one is represented by the higher order dependency which is a dynamic dependency. Finally, we introduce in our four views the concept of time grouping the agents, the dependencies and the coalitions present in the system in each time instant.

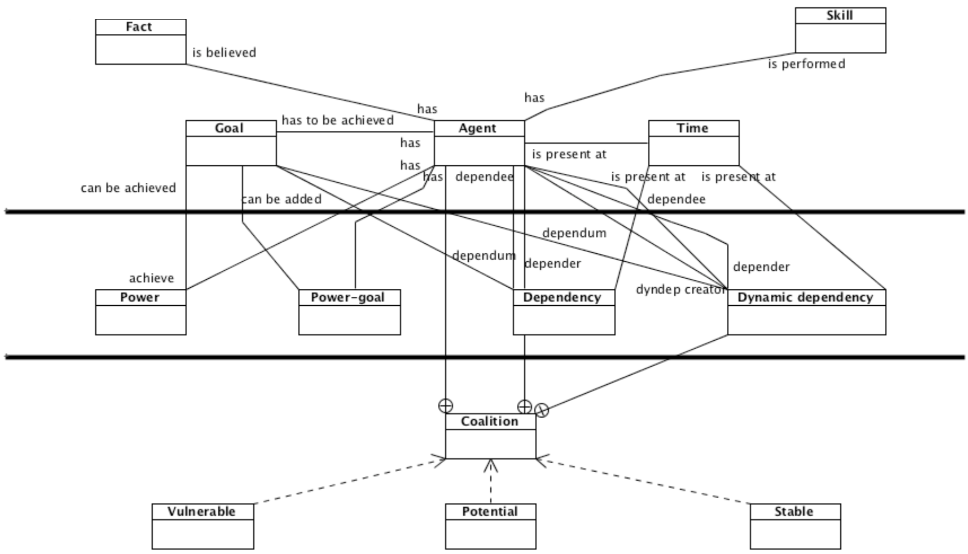


Figure 6.1: UML diagram of the ontology of our model.

6.1.3 Iterative design: refining viewpoints on grid-based coalitions

Figure 6.2 illustrates the iterative design of the grid scenario. It contains our four viewpoints and the refinement relations between them. Each row explains one viewpoint. Going from one row to the one below is a refinement, and going to a row above is an abstraction. The designer starts with the top row, and refines it in each step to the row below it. It can well be that the designer encounters a problem in a more refined view and then has to adapt the more abstract views, leading to the iterative design cycle. However, here we consider only the refinements of the views, not the revisions or updates of them.

In this section, we describe the four viewpoints in detail. For each viewpoint represented by a row, the leftmost column summarizes the

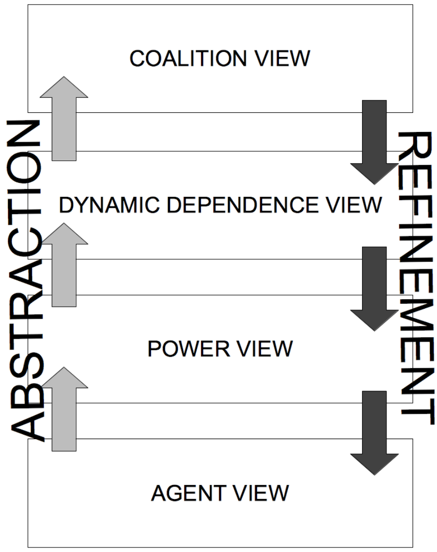


Figure 6.2: Iterative social network design.

part of the ontology used for this viewpoint. The next two columns visualize the first two elements of the temporal sequence within the viewpoint. The rightmost column gives some additional explanation on the grid example. The analysis method is implicitly represented in the example. Cooperation is represented by straight and dashed lines. A straight line represents a candidate coalition, and a dashed line represents that it is not formed.

The coalition view, in Figure 6.3, represents the most abstract viewpoint used to argue on coalitions. Concepts used in this viewpoint are two kinds of nodes, called coalitions and second-order arguments, and one kind of relation, called dominance or attack. The attack relation between candidate coalitions influences which coalition will be formed. In the grid example, we distinguish two candidate coalitions, formed by nodes of a virtual organization, attacking each other, and one second-order argument, preferring the first candidate coalition over the second one. This second-order argument attacks the attack from the candidate coalition C_2 to the candidate coalition C_1 at time frame t_1, and this second-order attack leads to the formation of coalition C_1. The second-order argument can itself be attacked by an higher-order attack, not represented in the figure.

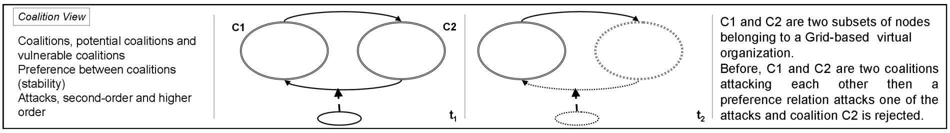


Figure 6.3: Coalition view.

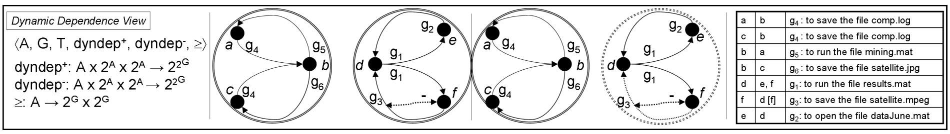


Figure 6.4: Dynamic dependence view.

The dynamic dependence view, in Figure 6.4, represents a refinement of the coalition view, because we introduce the agents and the dependencies that constitute the coalitions. Concepts used in this viewpoint are one kind of node, the agent, and two kinds of relations, representing respectively common dependencies and higher-order dependencies. Goals are represented only as labels of the dependence relations. In the grid example of Figure 6.4, each coalition consists of three nodes. A node can depend on two other nodes for the same goal, as in the case of node d for goal g_1 or two nodes can depend on the same node for a shared goal, as in the case of nodes a and c for goal g_4. The dynamic dependency of the example sees node f able to delete the dependency between itself and node d on goal g_3.

In the power view, in Figure 6.5, we refine the dynamic dependence view. Concepts used in this viewpoint are the same as before, agents and goals, but three new relations, one associating agents with goals (*goals*), one which says which goals a *set of* agents can achieve (*power*), and one which represents which sets of goals can be created or destroyed by an agent (*power-goal*). Likewise there is the possibility to create or destroy powers, not directly represented in the figure. The power relation is depicted as a square including agents and goals and the power-goal relation is depicted as a squared goal linked to the agents that can add or remove it. In the grid example, node f has the power-goal to delete its goal g_3 while node d has the power to see to goal g_3.

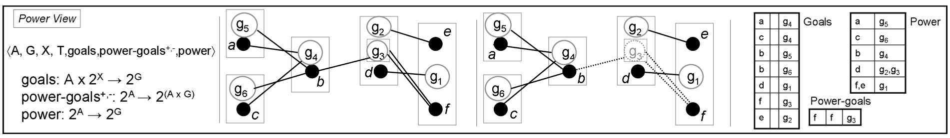


Figure 6.5: Power view.

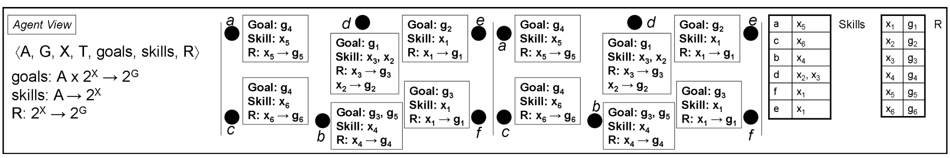


Figure 6.6: Agent view.

In the agent view, in Figure 6.6, we finally refine the power view. The used concepts are skills and rules. Each agent has some skills, whereas in the power view, each *set* of agents has the power to see to other agents' goals. So the power view is more "social" than the agent view. In Figure 6.2, skills are represented for each agent whereas the power is represented for a set of agents, as indicated by the square around them. The agent view is the most detailed view since it considers all the features of the single agents but it looses the notion of "group" present in the power view.

6.2 Arguing on abstract coalitions models

In this section, we answer to the second research question of the Chapter presenting the abstract coalitions models on which we analyze reciprocity-based coalitions that may emerge in social networks at the higher level of abstraction. This can be specified by the following subquestion: How to represent coalition formation and coalitional game theory in meta-argumentation? Dung [Dun95] introduces game theory as one of the three applications of his abstract theory (besides nonmonotonic reasoning and logic programming), and Amgoud [Amg05] shows how to instantiate preference-based argumentation with a task-based coalition formation theory. However, in Amgoud [Amg05], arguments why one coalition would be preferred over another one are not open for debate. In our approach, the preference between arguments is defined in terms of the second-order arguments. These additional arguments set the preference of one argument over the others, attacking the attacks towards the preferred argument. These arguments may be called also *stability arguments* in order to express coalitions' evolution where, on the one hand, coalition's stability is maintained if the coalition is not attacked by the other coalitions, and, on the other hand, the stability is destroyed if the coalition is not preferred over the others and thus it is attacked by some other coalitions.

Let us consider the examples of Figure 6.7. The coalition argument D attacks the coalition argument C, but this attack is itself attacked by the second-order argument B. In other words, we see each candidate coalition as an argument. Candidate coalition D attacks candidate coalition C and the second-order argument B attacks this attack to set a preference between the two candidate coalitions. This is a second-order attack [Mod09].

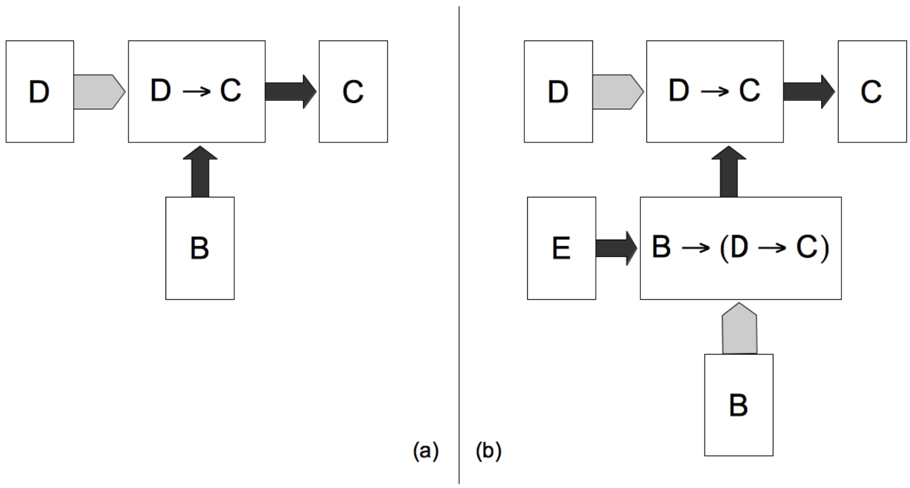


Figure 6.7: (a) Modgil - Bench-Capon scheme, (b) Higher-order argumentation.

In Figure 6.7, we have two kinds of arguments, the atomic arguments and the attack arguments. We represent with the grey arrow the support relation between two arguments, e.g. argument D supports the attack $D \rightarrow C$, and with the black arrow the attack relation between two arguments, e.g. the second-order argument B attacks the attack $D \rightarrow C$. An argument can also support another argument, e.g. when an agent gives an argument which confirms a premise used by an argument provided by another agent.

In coalition formation, as depicted in Figure 6.7, typically coalition D and coalition C conflict, so D not only attacks C, but C also attacks D. This means that the two coalitions cannot or should not be formed together. The second-order argument B represents a preference setting that coalition C is better than coalition D. Also argument E is a second-order argument and it attacks the relevance of the second-order argument B changing the total preference over the coalitions. At this level of abstraction, conflicts are not explicitly defined and distinguished while they are described in details in the refined dynamic dependence view. Figure 6.9 presents the flattened version of the argumentation networks represented in Figure 6.7. The main difference consists in the representation of the arguments by means of auxiliary meta-arguments, e.g., argument a is represented with the two meta-arguments X and Y.

Now we define the flattening procedure for second-order attacks in our meta-argumentation theory for coalition formation. From Definition 25, we define second-order attacks in the following way. Each time a coalition (or argument) raises a second-order attack, in the flattened argumentation network we add a new meta-argument Z representing the preference. Z_a is the preference set by argument a, such as a second-order argument. This meta-argument attacks, always by means of X and Y meta-arguments, the Y meta-argument representing the attack, e.g. from b to c. A particular case is when argument a is attacked by another argument d. Since meta-argument Z_a is strictly linked to argument a of which it represents a preference, it has to be attacked too by argument d. In this general framework, an exception occurs when the argument attacking a is part of a cycle including a itself. In this case, argument d does not attack meta-argument Z_a too since his acceptability is due to this second-order argument.

Definition 25 *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

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

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

if and only if $a \rightarrow b$

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

if and only if $a \rightarrow^2 (b \rightarrow c)$ and if there exists $accept(d) \rightarrow a \rightarrow^2 (b, c)$ then

$$Y_{d,a} \longmapsto accept(Z_a)$$

if and only if there not exists $d \rightarrow a \rightarrow^2 (c \rightarrow d)$ and $b \rightarrow c \rightarrow d \rightarrow a$

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 depicted in Figure 6.8. We have four coalitions, represented by arguments a, b, c, d, and a preference relation, represented by the second-order argument Z_a. Without coalition e, we have that the accepted arguments are $\{accept(b), accept(Z_a), accept(c), accept(a)\}$ because the second-order argument is accepted and thus the attack relation between arguments b and c is destroyed and argument a is accepted too. If there exists another coalition e attacking a without having its acceptability dependent of the second-order argument expressing the preference of a then, argument e attacking a also attacks meta-argument Z_a.

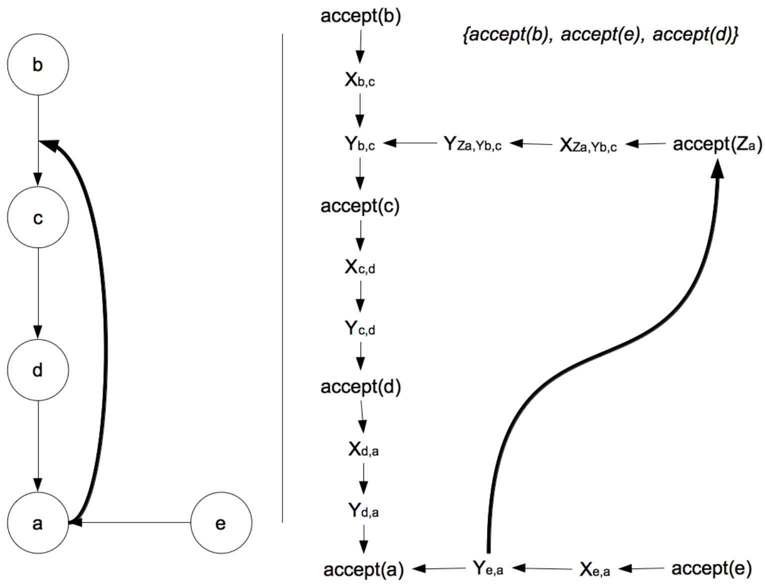


Figure 6.8: Example of second-order arguments.

Our aim with the introduction of the Z meta-argument is that we want to maintain the argument, e.g. a, and its preference or second-order argument, e.g. Z_a, disconnected due to the different levels in which they are positioned in the argumentation framework. Second-order arguments represent in our coalition formation model the real destruction of a coalition by means of the power of one or more of the agents of the first coalition which can add or remove some of its dependencies. What we need is the possibility of having as accepted the meta-argument Z_a even if argument a is not accepted. An example of this kind of situation is provided in the last section of this chapter and it is due to a desire of not to be formed of a coalition which knows that its goal would be achieved anymore without the costs linked to the reciprocity constraints of the coalition contract.

Concerning the acceptability of arguments and second-order arguments, when a coalition is said to be accepted then we go from the coalition view to the dynamic dependence view and, given the second-order arguments which are accepted too, we verify if the accepted coalition is already constrained by a reciprocity-based contract. If the answer is yes, then the coalition will be formed otherwise we do not have the coalition anymore. This kind of approach to argumentation is in some sense related to hierarchical argumentation [Mod06] since we have two kinds of arguments and the acceptability of first-order arguments depends on the acceptability of second-order arguments and attacks on first-order arguments leads to attacks to second-order arguments. Moreover, the acceptability of a second-order argument does not depend on the acceptation of the first-order argument to which it is linked.

Example 12 shows the application of our argumentation framework to compute which coalition is formed in each time instant using, e.g., the preferred semantics.

Example 12 *Let us consider the example depicted in Figure 6.10. Figure 6.10.a represents the case of three candidate coalitions which aim to be formed in the context of a virtual organization in a grid and this leads to the following attacks:* $C_1 \rightarrow C_2$ *and* $C_2 \rightarrow C_3$*. Moreover, there is also the second-order attack:* $C_3 \rightarrow (C_1 \rightarrow C_2)$*. The aim of our arguing model is to decide what coalition would be formed in this case. In Figure 6.10.a, candidate coalition* C_3 *knows that the only way to be formed consists in avoiding the formation of candidate coalition* C_2*.* C_3 *has the possibility to attack* C_1*'s attack due to its powers, specified at the lower level of abstraction, of adding or deleting one or more of the dependencies composing* C_1*.* C_3 *decides to not use its capability of attacking the attack* $C_1 \rightarrow C_2$*. The decision of* C_3 *of avoiding the second-order attack in order to be formed is represented by means*

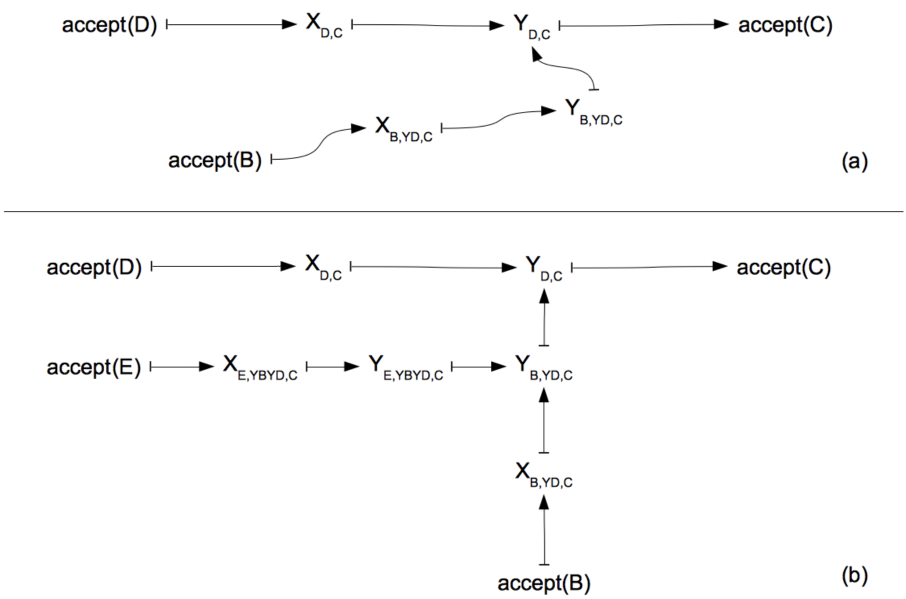


Figure 6.9: Flattened version of the networks of Figure 6.7 with X and Y meta-arguments.

of adding an higher-order attack from C_3 attacking its second-order attack $C_3 \rightarrow (C_1 \rightarrow C_2)$. In the figure, higher-order attacks are depicted as dotted arrows, while second order ones are depicted as dashed arrows on the left side of Figure 6.10. Let $\mathcal{AF} = \langle A, \rightarrow \rangle$ be our argumentation framework with $C_1, C_2, C_3 \subseteq A$, then the extensions of the argumentation framework are as follows: if an higher-order attack attacking the second-order attack is added to the argumentation framework, $\{accept(C_1), accept(C_3)\}$, while without the higher-order attack $\{accept(C_1), accept(C_2), accept(Z_{C3})\}$. Thus, C_3 should add the higher-order attack to inhibit the second-order one, otherwise, C_3 will not be formed. Recall that while higher order attacks can be added by the attacking coalition itself to the argumentation framework, first-

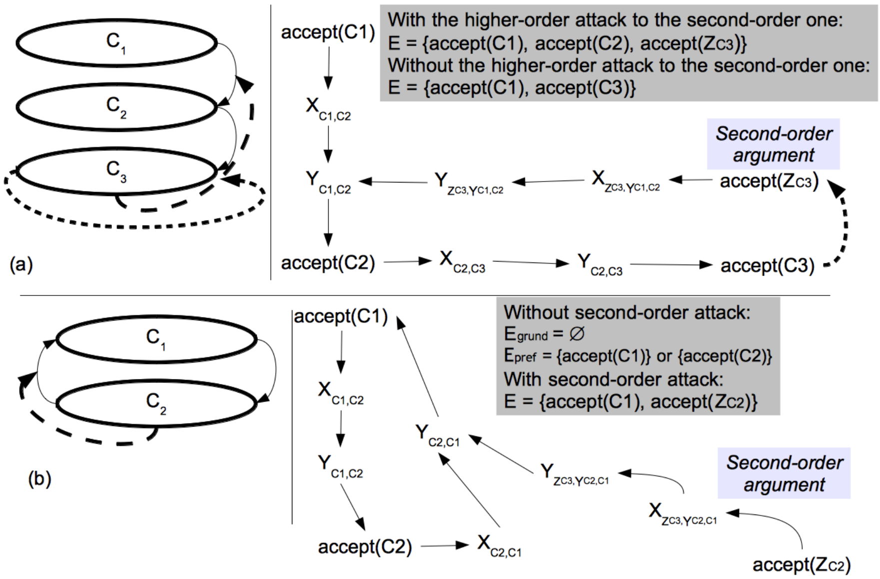


Figure 6.10: Candidate coalitions attacking each other from Example 12.

and second-order attacks are determined only by the lower levels of abstraction. Thus coalitions cannot add or delete them at their will, but they can only attack them via higher-order attacks.

Figure 6.10.b visualizes two candidate coalitions belonging to the same grid-based virtual organization attacking each other. In this case, candidate coalition C_2 does not want to be formed since, for example, it can achieve its goal without any effort if coalition C_1 is formed. Thus $C_2 \rightarrow (C_2 \rightarrow C_1)$. Let $\mathcal{AF} = \langle A, \rightarrow \rangle$ our argumentation framework with $C_1, C_2 \subseteq A$ then the extensions are, without the second-order attack, $\{accept(C_1)\}$ or $\{accept(C_2)\}$. This situation can be seen as a sort of deadlock. Otherwise, if there is the presence of the second order

attack due to the possibility for candidate coalition C_2 of adding or removing one or more of the dependencies of the concurrent coalition C_1, then the extension is $\{accept(C_1), accept(Z_{C2})\}$ and the only formed coalition is C_1, as desired by coalition C_2. Figure 6.10.b depicts a second order attack where a second-order argument Z_{C2} sets a preference of coalition C_1 over coalition C_2. There can be another second-order argument setting the preference of coalition C_2 over coalition C_1, attacking by means of an incompatibility attack the first second-order argument. This would be the case in which also coalition C_1 does not want to be formed for the same reasons of coalition C_2.

6.3 Analyzing reciprocity based coalitions

In this section, we answer to the third research question. First, we present the dynamic dependence view and the refined notion of coalition for this view. Second, we show how to argue on the attacks between coalitions in this refined level of abstraction.

6.3.1 Dependence Networks

Conte and Sichman [SC02] introduce dependence networks, a kind of social networks representing how each agent depends on other agents to achieve the goals he cannot achieve himself. Dependence networks are based on Castelfranchi [Cas03]'s basic notion of social power. They are used to specify early requirements in the TROPOS methodology [BPG+04], and to model and reason about agents' interactions in multiagent systems by Conte and Sichman [SC02].

The theory of social power and dependence is an attempt to transfer theories developed initially in the field of sociology to the field of multiagent systems and to refine them. This theory models the potential interactions among the agents which lead to the achievement of a shared goal, i.e. cooperation, or the reciprocal satisfaction of their own goals, i.e. social exchange. This involves the development of a social reasoning mechanism that analyzes the possibility to profit from mutual-dependencies, e.g., the case in which two agents depend on each other for the satisfaction of a shared goal, or reciprocal-dependencies, e.g., the case in which two agents depend on each other for the satisfaction of two different goals.

In a multiagent system, since an agent is put into a system that involves also other agents, he can be supported by the others to achieve his own goals if he is not able to do them alone. This leads to the concept of power representing the capability of a group of agents (possibly composed only by one agent) to achieve some goals (theirs or of other agents) performing some actions without the possibility to be obstructed. The power of a group of agents is defined as follows:

Definition 26 (Agents' power) $\langle A, G, power : 2^A \rightarrow 2^{2^G} \rangle$ *where* A *is a set of agents and* G *is a set of goals. The function power relates with each set* $S \subseteq A$ *of agents the sets of goals* $G_S^1, \dots, G_S^m$ *they can achieve.*

Example 13 *In the Grid scenario, the simplest example of power consists in the power of the local or global administrator to give to common users the possibility to access a resource. Particularly, if we consider a role based access control policy, the Grid administrator has the power to give to the common users, under request, a new role which makes them able to access a resource. Other kinds of power are, for example, the power to perform a heavy computation or to store a great amount of data.*

The notion of power brings to the definition of a structure with the aim to show the dependencies among agents. In order to define these relations in terms of goals and powers, we adopt, as said, the methodology of dependence networks developed by Conte and Sichman [SC02] and extended with the notion of time by Caire and van der Torre [CvdT09]. In this model, an agent is described by a set of prioritized goals, and there is a global dependence relation that explicates how an agent depends on other agents for fulfilling its goals. For example, $dep(\{a, b\}, \{c, d\}) = \{\{g_1, g_2\}, \{g_3\}\}$ expresses that the set of agents $\{a, b\}$ depends on the set of agents $\{c, d\}$ to see to their goals $\{g_1, g_2\}$ or $\{g_3\}$. For each agent we add a priority order on its goals, and we say that agent a gives higher priority to goal g_1 than to goal g_2, written as $\{g_1\} \succ(a) \{g_2\}$, if the agent tries to achieve goal g_1 before it tries to achieve g_2. In other words, it gives more attention to g_1 than to g_2. A dependence network is defined as follows:

Definition 27 (Dependence Networks (DN)) *A dependence network is a tuple $\langle A, G, dep, \geq \rangle$ where:*

- *A is a set of agents;*
- *G is a set of goals;*
- *$dep : 2^A \times 2^A \rightarrow 2^{2^G}$ is a function that relates with each pair of sets of agents all the sets of goals on which the first depends on the second.*
- *$\geq: A \rightarrow 2^G \times 2^G$ is for each agent a total pre-order on goals which occur in his dependencies: $G_1 \geq (a)G_2$ implies that $\exists B, C \subseteq A$ such that $a \in B$ and $G_1, G_2 \in depend(B, C)$.*

In the early requirements analysis, we model the dependencies among the agents and the roles associated to the agents of the organization. In this way, we represent the domain stakeholders and we model them using the multiagent paradigm. These dependencies are based both on goals and institutional goals. In the late requirements analysis, the same kind of approach is followed but the agents involved in the dependence networks are those of the future system. A graphical representation of the model of the *dependency modeling* is built following the legend of Figure 6.11 which describes the agents (depicted as white circles), the roles (depicted as black circles), the agents assigned to roles (depicted as grey circles), the agents'/roles' goals (depicted as white rectangles) and the dependency among agents (one arrowed line connecting two agents with the addition of a label which represents the goal on which there is the dependency). For simplicity, the legend considers the dependency only among agents but these dependencies can be also among roles or agents assigned to roles.

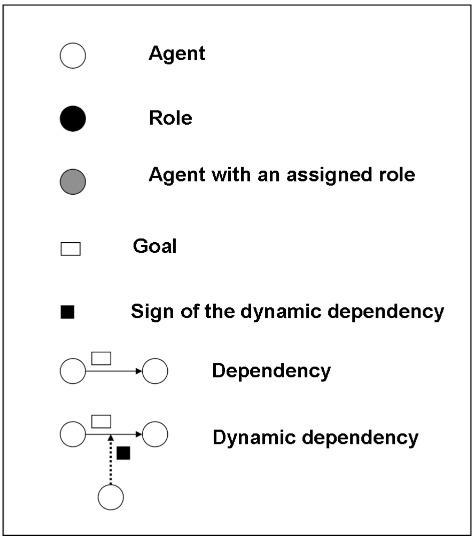


Figure 6.11: The legend of the graphical representation of the modeling activities of *dependency* and *dynamic dependency*.

We present a first example of modeling a virtual organization based on a grid network containing only the notions of the agent view.

Example 14 *Considering a grid composed by the nodes of Figure 6.12, we can imagine to view each node as an agent and we can form the following dependence network* $DN = \langle A, G, dep, \geq \rangle$*:*

1. *Agents* $A = \{n_1, n_2, n_3, n_4, n_5, n_6\}$*;*

2. *Goals* $G = \{g_1, g_2, g_3, g_4, g_5, g_6\}$*;*

3. $dep(\{n_1\}, \{n_2\}) = \{\{g_1\}\}$*: agent* n_1 *depends on agent* n_2 *to achieve the goal* $\{g_1\}$*: to store the file* comp.log*;*

   $dep(\{n_2\}, \{n_3\}) = \{\{g_2\}\}$*: agent* n_2 *depends on agent* n_3 *to achieve the goal* $\{g_2\}$*: to run the file* mining.mat*;*

   $dep(\{n_3\}, \{n_1\}) = \{\{g_5\}\}$*: agent* n_3 *depends on agent* n_1 *to achieve the goal* $\{g_5\}$*: to store the file* satellite.jpg*;*

   $dep(\{n_4\}, \{n_6\}) = \{\{g_3\}\}$*: agent* n_4 *depends on agent* n_6 *to achieve the goal* $\{g_3\}$*: to run the file* results.mat*;*

   $dep(\{n_6\}, \{n_5\}) = \{\{g_4\}\}$*: agent* n_6 *depends on agent* n_5 *to achieve the goal* $\{g_4\}$*: to store the file* satellite.mpeg*;*

   $dep(\{n_5\}, \{n_3\}) = \{\{g_6\}\}$*: agent* n_5 *depends on agent* n_3 *to achieve the goal* $\{g_6\}$*: to have the authorization to open the file* dataJune.mat*;*

Example 14 shows the dependence network based on a simple grid example composed by six agents. The kind of dependencies are all related to the agent view and they always refer to material goals and not to the institutional ones, except for goal g_6. Using dependence networks as methodology to model a system advantage us from different points of view. First, they are abstract, so they can be used for example for conceptual modeling, simulation, design and formal analysis. Second, they are used in high level design languages, like TROPOS [BPG+04], so they can be used also in software implementation.

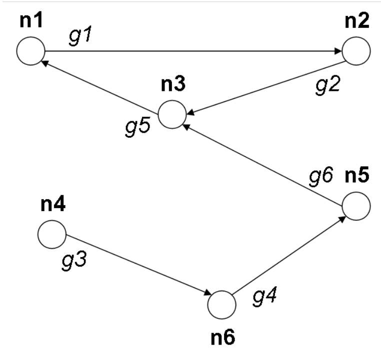


Figure 6.12: Dependence Network of Example 14.

6.3.2 Refining coalitions with dynamic dependencies

Dynamic dependence networks have been introduced by Caire *et al.* [CVBvdT08]. In this work, a dependency between agents can depend on the interaction of other agents. Here, as done by Boella *et al.* [BvdTV08b], we distinguish "negative" dynamic dependencies where a dependency exists unless it is removed by a set of agents, due to removal of a goal or ability of an agent, and "positive" dynamic dependencies where a dependency may be added due to the power of a third set of agents. As explained in the following section, these two dynamic dependencies can be used to reason about the evolution of candidate coalitions at the dynamic dependence view level of abstraction.

Definition 28 (Dynamic Dependence View) *A dynamic dependence network is a tuple* $\langle A, G, T, dyndep^-, dyndep^+, \geq \rangle$ *where:*

- A *is a set of agents,* G *is a set of goals and* T *is a set of time frames.*

- *dyndep*$^-$ $: A \times 2^A \times 2^A \rightarrow 2^{2^G}$ *is a function that relates with each triple of an agent and two sets of agents all the sets of goals in which the first depends on the second, unless the third deletes the dependency.*

- *dyndep*$^+$ $: A \times 2^A \times 2^A \rightarrow 2^{2^G}$ *is a function that relates with each triple of an agent and two sets of agents all the sets of goals on which the first depends on the second, if the third creates the dependency.*

- $\geq: A \rightarrow 2^G \times 2^G$ *is a total pre-order on goals which occur in each agent's dependencies:* $G_1 \geq (a)G_2$ *implies that* $\exists B, C \subseteq A$ *such that* $a \in B$ *and* $G_1, G_2 \in$ *dyndep*$^-(a, B, C)$ *or* $G_1, G_2 \in$ *dyndep*$^+(a, B, C)$.

The static dependencies are defined by dep$(a, B) =$ *dyndep*$^-(a, B, \emptyset)$.

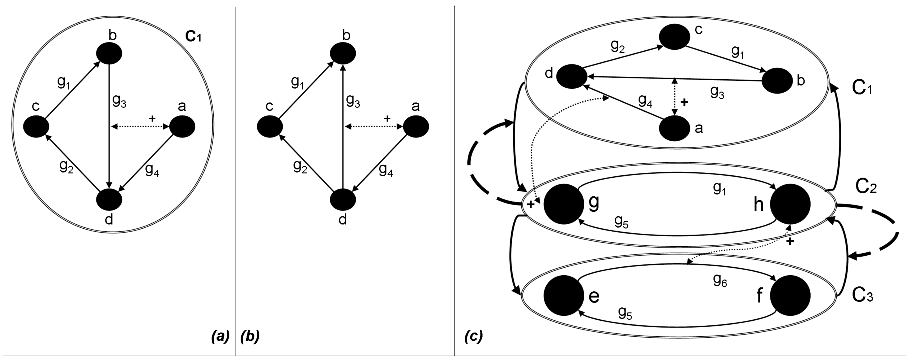


Figure 6.13: Agents forming a coalition (a) or not (b); the coalition view and the dynamic dependence view merged together (c)

Example 15 *Let us consider the example depicted in Figure 6.13.a where we have four nodes belonging to a grid-based virtual organization. Node b depends on node d for goal g_3, if node a creates this dependency: $dep(a, \{d\}) = \{\{g_4\}\}$, $dep(d, \{c\}) = \{\{g_2\}\}$, $dep(c, \{b\}) = \{\{g_1\}\}$, $dyndep^+(b, \{d\}, \{a\}) = \{\{g_3\}\}$.*

A coalition can be defined in dependence networks, based on the idea that to be part of a coalition, every agent has to contribute something, and has to get something out of it. Roughly, a coalition can be formed when there is a cycle of dependencies (the definition of coalition is more complicated due to the fact that an agent can depend on a set of agents, see below). We show how dependence networks can be used for coalition evolution, by assuming that goals are maintenance goals rather than achievement goals, which give us automatically a longer term and more dynamic perspective. Agents' goals in a dynamic environment are often more than just achieving a desired state, as after the agent has successfully acted to achieve a goal the environment may change that state. In such a case, a common goal of an agent is to "maintain", as for contracts, rather than just "achieve", as for coalitions, certain conditions.

We define reciprocity-based coalitions for dynamic dependence networks, firstly introduced by Boella *et al.* [BvdTV08d], as a refinement of the coalition view. We represent the coalition not only as a set of agents, like in game theoretical approaches, but as a set of agents together with a partial dynamic dependence relation. Intuitively, the dynamic dependence relation represents the "contract" of the coalition: if $H \in dyndep^+(a, B, D)$, then the set of agents D is committed to create the dependency, and the set of agents B is committed to see to the goals H of agent a. The rationality constraint on such reciprocity-based coalitions is that each agent contributes something, and receives something back. Our notion of coalition presents the agents composing it not only as utility maximizers as in coalitional game theoretical approaches but as complex entities with their sets of beliefs and goals which have to be satisfied. In our approach, coalitions have a complex structure, composed by existing dependencies and potential ones which represent a kind of dynamic contract.

Definition 29 (Reciprocity-based Coalition) *Given a dynamic dependence network $\langle A, G, T, dyndep^-, dyndep^+, \geq \rangle$, a reciprocity based coalition is represented by coalition $C \subseteq A$ together with dynamic dependencies $dyndep^{+'} \subseteq dyndep^+$, such that:*

- *if $\exists b, B, D, H$ with $H \in dyndep^{+'}(a, B, D)$ then $a \in C$, $B \subseteq C$ and $D \subseteq C$ (the domain of $dyndep^{+'}$ contains only agents in coalition C), and*

- *for each agent $a \in C$ we have $\exists b, B, D, H$ with $H \in dyndep^{+'}(b, B, D)$ such that $a \in B \cup D$ (agent a contributes something, either creating a dependency or fulfilling a goal), and*

- *for each agent $a \in C$ $\exists B, D, H$ with $H \in dyndep^+(a, B, D)$ (agent a receives something from the coalition).*

The following example illustrates that dependencies will be created by agents only if these new dependencies work out in their advantage.

Example 16 (Continued) *Each agent of $C_1 = \{a, b, c, d\}$ creates a dependency or fulfills a goal. Figure 6.13.a represents a set of agents composing a coalition in accordance with Definition 29 while Figure 6.13.b represents the same set of agents not forming a coalition. The difference among the two figures consists in the direction of the arrow joining agents b and d.*

6.3.3 Maintaining or destroying coalitions

The basic attack relations between coalitions are due to the fact that coalitions are based on the same goals, differently from the conflicts between coalitions in Amgoud's coalition theory [Amg05] where two coalitions are based on the same tasks. In the coalition view, we distinguish between two kinds of attacks: first-order ones, between meta-arguments of the kind $accept(a)$, and higher-order attacks, between X and Y meta-arguments. In the dynamic dependence view, we details these two kinds of attacks. Attack relations between coalitions sharing the same goals are the refined version of first order attacks presented in the coalition view. In this view, we present first-order attacks as the reciprocal attacks between coalitions without coming into details of the reasons behind these attacks. In this refined view, this reason is characterized by the sharing of a goal between the two (or more) coalitions. In this case, the two candidate coalitions cannot be formed together since an agent cannot be part of two coalitions at the same time, particularly if the two candidate coalitions are based on the same goal since each goal cannot be achieved concurrently by more than one agent.

Definition 30 (First-order attack) *Coalition $\langle C_1, dyndep_1\rangle$ attacks coalition $\langle C_2, dyndep_2\rangle$ if and only if there exist $a_1, a_2, B_1, B_2, D_1, D_2, G_1, G_2$ such that $G_1 \in dyndep_1(a_1, B_1, D_1)$, $G_2 \in dyndep_2(a_2, B_2, G_2)$ and $G_1 \cap G_2 \neq \emptyset$.*

Figure 6.14 aims at representing in the refined version the two cases in which a coalition wants or not to be formed. In Figure 6.14 two candidate coalitions composed by three nodes of the grid are depicted. On the one hand, in the first case we have that both the two candidate coalitions want to be formed. This is a sort of deadlock situation but it would be solved thanks to the presence of eventual dynamic dependencies. Two reasons are behind the decision of an agent to "use" one

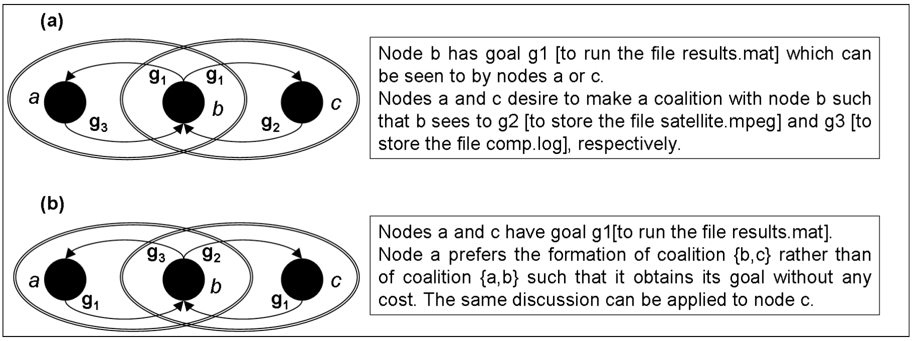


Figure 6.14: Candidate coalitions sharing goals.

or more of the dynamic dependencies under its control: first, the agent uses the dynamic dependency in order to destroy a coalition of which it is not a member, second, the agent uses the dynamic dependency in order to destroy the coalition of which it is a member. Although the reasons behind the first case are clear and received a lot of attention also in coalitional game theory, the reasons behind the second choice are less evident. The thing is that if the agent, knowing that each goal can be achieved atomically, "supports" the formation of another coalition which will achieve the goal he aims at, without the necessity to help other agents in achieving their goals. These two candidate coalitions are attacking each other as the first two coalitions of Figure 6.10.a. On the other hand, in the second case we have that both nodes a and c depend on node b to run the file *results.mat* and both of them know that if the other coalition is formed goal g_1 will be achieved without any effort. These two candidate coalitions are attacking each other but if, for example, coalition C_2 has the possibility to delete one of its dependencies then this higher-order attack would decide the formation of coalition C_1. In this way, coalition C_2 obtains its aim and goal g_1 will be achieved by agent a.

Definition 31 presents three different classes in which we divide the set of candidate coalitions due to their features and the sign, positive or negative, of the dynamic dependencies involving them.

Definition 31 *Let A be a set of agents and G be a set of goals. A coalition function is a partial function $C : A \times 2^A \times 2^G$ such that $\{a \mid C(a,B,G)\} = \{b \mid b \in B, C(a,B,G)\}$, the set of agents profiting from the coalition is the set of agents contributing to it. Let $\langle A, G, T, dyndep^-, dyndep^+, \geq \rangle$ be a dynamic dependence network, and dep the associated static dependencies.*

1. *A coalition function C is a coalition if $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in dep(a,B)$. These coalitions which cannot be destroyed by addition or deletion of dependencies by agents in other coalitions.*

2. *A coalition function C is a vulnerable coalition if it is not a coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in \cup_D dyndep^-(a,B,D)$. Coalitions which do not need new goals or abilities, but whose existence can be destroyed by removing dependencies.*

3. *A coalition function C is a potential coalition if it is not a coalition or a vulnerable coalition and $\exists a \in A, B \subseteq A, G' \subseteq G$ such that $C(a,B) \rightarrow G'$ implies $G' \in \cup_D(dyndep^-(a,B,D) \cup G' \in dyndep^+(a,B,D))$. Coalitions which could be created or which could evolve if new abilities or goals would be created by agents of other coalitions on which they dynamically depend.*

There are various further refinements of the notion of coalition. For example, Boella *et al.* [BSvdT06] look for minimal coalitions. In this thesis we do not consider these further refinements.

Second-order attacks are detailed in the dynamic dependence view by removing or adding one of the dependencies of the attacked coalition. This kind of attack is the refined version of second-order attacks of the coalition view and is represented by means of the second-order arguments. This kind of attack relation means a real destruction of the attacked coalition since one or more of its dependencies are deleted or added and the coalition does not exist any more. The second-order argument establishes the preference and the preferred coalition is preserved by these additions and removals and thus it maintains its stability. Higher-order attacks represent the decision of the coalition of not to raise a second-order attack and they are modeled as a new attack from the coalition to its second-order attack. Two points have to be highlighted concerning these two kinds of attacks:

- Each second-order attack is originated by a dynamic dependency thus the second-order attack attacks each coalition in which the dynamic dependency adds or deletes a common dependency.
- Each higher-order attack means the decision of not adding or deleting a dependency given a dynamic dependency. This means that when an higher-order attack is fired then it attacks each second-order attack which has been originated by the dynamic dependency the coalition decides not to raise.

Definition 32 (Second-order attack) *$\forall C_1, C_2$ such that $C_1 \rightarrow C_2$, coalition $\langle C, dyndep \rangle$ attacks the attack from coalition $\langle C_1, dyndep_1 \rangle$ on coalition $\langle C_2, dyndep_2 \rangle$ if and only if there exists a set of agents $D \subseteq \{a \mid \exists E, H, C(a, E, H)\}$ such that $\exists a, B, G', C_1(a, B, G')$ and $G \in dyndep(a, B, D)$.*

Second-order attacks, presented in Definition 32, can arise if the coalition C which has to attack the attack $C_1 \rightarrow C_2$ is composed by a set of agents D such that they can add or delete at least one dynamic dependency.

Example 17 *Assume we have eight agents, $a, ..., h$ and the dependencies of Example 15, depicted in Figure 6.13.c:* $dep(a, \{d\}) = \{\{g_4\}\}$, $dep(d, \{c\}) = \{\{g_2\}\}$, $dep(c, \{b\}) = \{\{g_1\}\}$, $dyndep^+(b, \{d\}, \{a\}) = \{\{g_3\}\}$, *plus the following ones:*
$dep(e, \{f\}) = \{\{g_6\}\}$, $dep(f, \{e\}) = \{\{g_5\}\}$, $dep(g, \{h\}) = \{\{g_1\}\}$, $dep(h, \{g\}) = \{\{g_5\}\}$, $dep(c, \{h\}) = \{\{g_1\}\}$, $dep(g, \{b\}) = \{\{g_1\}\}$, $dep(h, \{e\}) = \{\{g_5\}\}$, $dep(f, \{g\}) = \{\{g_5\}\}$.

The possible coalitions are C_1, C_2 and C_3 where:

$$C_1 = \{dep(a, \{d\}) = \{\{g_4\}\}, dep(d, \{c\}) = \{\{g_2\}\}, dep(c, \{b\}) = \{\{g_1\}\}, dyndep^+(b, \{d\}, \{a\}) = \{\{g_3\}\}\},$$
$$C_2 = \{dep(e, \{f\}) = \{\{g_6\}\}, dep(f, \{e\}) = \{\{g_5\}\}\},$$
$$C_3 = \{dep(g, \{h\}) = \{\{g_1\}\}, dep(h, \{g\}) = \{\{g_5\}\}\}.$$

Some of the dependencies remain outside all coalitions (e.g., $dep(c, \{h\}) = \{\{g_1\}\}$, $dep(g, \{b\}) = \{\{g_1\}\}$, $dep(h, \{e\}) = \{\{g_5\}\}$, $dep(f, \{g\}) = \{\{g_5\}\}$, *not reported in Figure 6.13.c). Thus,* $C_1 \to C_2$, $C_2 \to C_1$, $C_2 \to C_3$ *and* $C_3 \to C_2$ *due to the fact that they share goals* g_1 *and* g_5 *respectively. Note that these attacks are reciprocal. The coalitions attack each other since agents b and h on which respectively c and g depend for g_1 would not make their part hoping that the other one will do that, so to have a free ride and get respectively goal g_3 achieved by agent d and goal g_5 by agent g.*

Figure 6.15 illustrates a new example of conflict among vulnerable coalitions.

Example 18 *Using the grid-based scenario, we can model the example depicted in Figure 6.15. Assume, in the first time instant t_1, we have a portion of a virtual organization composed by three nodes, a, b, c represented as agents in our model. There are three goals g_1: to run the file* results.mat, *g_2: to save the file* satellite.mpeg, *g_3: to save the file* comp.log.

These goals, associated to the power of the agents to achieve them, form the following dependencies among the agents (we write $C(a, b, g_1)$

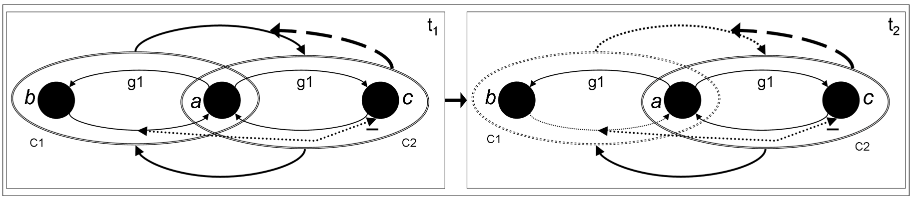


Figure 6.15: Two vulnerable coalitions attacking each other.

for $C(a, \{b\}, \{g_1\})$ *and* $dep(a, b, g_1)$ *for* $dep(a, \{b\}) = \{g_1\}$*):* $dep(a, b, g_1)$, $dep(a, c, g_1)$, $dep(b, a, g_2)$, $dep(c, a, g_3)$. *The situation is that node a depends on both nodes b and c to run the file* results.mat *and thus to obtain the results of his job, node b depends on node a for the storage of file* satellite.mpeg *and, finally, node c depends on node a for the storage of file* comp.log. *Thus, there are two candidate coalitions:* $C_1 = \{(a, b, g_1), (b, a, g_2)\}, C_2 = \{(a, c, g_1), (c, a, g_3)\}$.

They will not create both since one is enough for node a to have someone look after his goal g_1*:* $C_1 \rightarrow C_2$ *and* $C_2 \rightarrow C_1$. *Now, we assume that node c removes the necessity of node b to store the file* satellite.mpeg*, destroying the dependency* $dep(b, a, g_2)$*, i.e., we substitute it with* $dyndep^{-}(b, a, c, g_2)$*, e.g., by removing the power of node a to see to goal* g_2*, or by removing the goal* g_2 *of node b. This deletion, shown in time instant* t_2 *of Figure 6.15, allows node c to ensure himself the dependency on himself of node a to perform his job, goal* g_1. *In this way, node c ensures himself the help of node a to store file* comp.log. *This deletion sets a preference relation of the candidate coalition* C_2*, represented here with the attack of coalition* C_2 *to the attack relation of coalition* C_1 *to coalition* C_2. *In this case, coalition* $C_2 = \{(a, c, g_1), (c, a, g_3)\}$ *will become the only possible extension, since* $C_2 \rightarrow (C_1 \rightarrow C_2)$ *by Definition 32.*

Chapter 7

Related Work

In this thesis, we introduce the methodology of meta-argumentation to model argumentation itself. Bondarenko *et al.* [BDKT97] and Verheij [Ver03] may be seen as predecessors of the meta-argumentation approach.

In some way, Dung and colleagues [BDKT97] 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.* [BDKT97] 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 [Ver03] 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, and it is expressed by $j \leadsto \times y$.

Meta-argumentation has been treated in an explicit way in the following works. Jakobovitz and Vermeir [JV99] 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, providing a simple procedure to compute robust sets. They define 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.

A work which discusses another way of doing flattening of argumentation frameworks is presented by Gabbay [Gab09b, Gab09a]. 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 a 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 [Cam06] labelling thinking in terms of labels as functions and giving values to the nodes in a 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 meta-argumentation methodology are both concerned with the notions of abstraction and instantiation. In Gabbay [Gab09b, Gab09a], 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 can also have the same arguments at distinct abstraction levels. The applied methods are different. While Gabbay [Gab09b, Gab09a] 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., particularly 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 by Wooldridge *et al.* [WMP05]. The starting point of this work is the same of our one and consists in the point of 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 clear 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 language.

Modgil and Bench-Capon [MBC08] 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's argumentation frameworks. In particular, they use 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 $a \rightarrow b$ 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.* [BCGG09] investigate the generalization of 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 achieved by a recursive definition of the attack relation leading to the introduction and investigation of a formalism called argumentation framework with recursive attacks.

7.1 Subsumption relation and Toulmin scheme

In Chapter 4, we introduce a new relation among arguments, called the subsumption relation. It is inspired by the counts-as relation coming from constitutive norms, where "X counts as Y in the context of C" is a standard representation to represent legal ontologies: a piece of paper counts as money, a procedure in an institution counts as getting married, and so on. Counts as relations may hold between brute and institutional facts, but also between actions or processes and propositions, and so on. Constitutive norms, introduced by Searle [Sea95] [Sea69], define that something *counts-as* something else for a given institution. Searle claims about these rules: "The activity of playing chess is constituted by action in accordance with these rules. The institutions of marriage, money, and promising are like the institutions of baseball and chess in that they are systems of such constitutive rules or conventions". There are various kinds of norms. Constitutive norms describing the legal ontology are usually represented as count as conditionals 'C counts as D in context B', and regulative norms are represented by 'in institution B, if C, then D is obligatory / prohibited / permitted'. They are used to detach so-called institutional facts (for constitutive norms) or deontic facts (for regulative norms). Norms generate a set of institutional and deontic facts [MvdT00], or multiple of those sets when the output is constrained by contrary-to-duty reasoning. The way we model counts-as is based on the classificatory view of counts-as statements: if counts-as statements yield classifications, this means that they are as conceptual subsumption relations, that is, counts-as statements assert just that a concept X is a sub-concept of a concept Y [GMD05].

Subsumption relations and argumentation are often related to the field of legal ontologies. As far as we know, there are no approaches about extending an argumentation framework by adding the subsumption relation. Subsumption relations have been introduced in inheritance networks and then in ontologies [WG01] and, in the last years, they are used in order to express subsumption between laws in the legal ontologies field.

Concerning the application of argumentation frameworks to normative reasoning, in Oren *et al.* [OLN08] and Oren *et al.* [OLMN08], the authors propose to use argument schemes in order to represent reasoning about rules. They present a number of argument schemes that can be used to reason about normative concepts. By representing its knowledge using these argument schemes, and using results from argumentation theory, an agent is able to infer, from the interactions between argument schemes, how to act on the basis of its norms, and whether any of its norms should be ignored. This approach has a non-monotonic nature and the main contribution of this work concerns the framework's ability to aid an agent in resolving the normative conflict. In Atkinson and Bench-Capon [ABC05], the authors provide a reconstruction of the reasoning of the majority and dissenting opinions for a case from property law. This approach uses instantiations of an argumentation scheme to provide presumptive justifications for actions, and critical questions to identify arguments which attack these justifications. These arguments and attacks are organised into argumentation frameworks to identify the status of individual arguments. In Atkinson *et al.* [ABCM06], the authors present the PARMENIDES system guiding the user through a justification of an action giving opportunities to disagree. Each of these disagreements represents an attack.

Concerning the argument schema proposed in [Tou58], in Bench-Capon [BC98] the author 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 introduces the notion of presupposition, which is supposed to represent propositions assumed to be true in the context, and so which do not need to be discussed but which can be made explicit if required. With this schema, the author argues to have some flexibility in assigning particular roles to premises in an argument.

7.2 Merging views

The problem of merging multiple sources of information is a central topic in many information processing areas such as databases integrating problems, multiple criteria decision making and multiagent systems. Different approaches have been proposed in this direction.

Coste-Marquis *et al.* [CMDK+07] present a new approach to the problem of merging argumentation frameworks belonging to the different agents, without using neither voting nor union. Their purpose is to characterize the set of arguments acceptable by a group of agents, when the information furnished by each agent consists solely of an abstract Dung's argumentation system. The authors proposes a three-step process in which, first, each argumentation system is expanded into a partial system over the set of all arguments considered by the group of agents in order to reflect that some agents may easily ignore arguments pointed out by other agents, as well as how all the arguments interact with its own ones. The second step is the real merging one and it is used on the expanded systems as a way to solve the possible conflicts between them. The last step consists in selecting the acceptable arguments at the group levels from the set of argumentation systems. The paper introduces the notion of partial argumentation system, which extends abstract Dung's argumentation system in order to represent ignorance concerning the attack relation. The argumentation system provided by each agent is first expanded into a partial argumentation system, and all such partial systems are built over the same set of arguments, those pointed out by at least one agent. Since there exist many ways to incorporate a new argument, the authors focus on one of them, called the consensual expansion. When incorporating a new argument into its system, an agent is ready to conclude that this argument attacks (or is attacked by) another argument whenever all the other agents who are aware of both arguments agree with this attack; otherwise, it concludes that it ignores whether an attack takes place or not.

A particular case treated by Coste-Marquis *et al.* [CMDK+07] is what can an agent i do on the attack relation in order to add a new argument b if he only has a. There are different strategies of expansion:

- always reject b (e.g., adding (b, b) to its attack relation R_i),
- always accept b (adding $(a, b), (b, a)$ and (b, b) to its non-attack relation N_i),

- express its ignorance about b (adding $(a, b), (b, a)$ and (b, b) to its ignorance relation I_i).

It could be noted that Coste-Marquis *et al.* [CMDK+07] do not consider the case in which a is attacked by b. This could be reasonable in the merging perspective, but, in the dialogue one, it is important also the role of the interactant the new argument comes from. For example, in a cooperative dialogue, one of the interactants puts a new argument to reinstate another argument they aim to prove, or a new argument for accrual with the other interactants' arguments against someone else. In other kinds of dialogue, the interactant puts new arguments against the others' ones, but maybe sometime he could make a concession, thus he puts forward an argument which does not attack the others. Another point which should be noted is that at point 3 of the above list, concerning $(a, b), (b, a)$ we have that the ignorance relation is symmetrical. Using our meta argumentation model, we make it non symmetrical, being in this way more related to the dialogue perspective in which this symmetry is not obligatory.

An approach to merging is provided also by Condotta *et al.* [CKMS09] where a merging procedure for qualitative constraints networks (QCN) is presented. Starting from a set of QCNs defined on the same set of variables $V = \{v_1, ..., v_n\}$ representing the spatial or temporal entities and a function C associating each pair of variables (v_i, v_j) and element R, where R is the set of all possible basic relations between v_i and v_j. The merged QCNs are defined also on the same qualitative algebra. Instead of merging directly the QCNs, the authors propose first to translate each QCN into a propositional formula and then merging these formulas using the classical merging operations, often based on a pseudo-distance d. This kind of merging is a three step process: first each QCN is encoded into a propositional formula, second an integrity constraint operator for merging is applied on the resulting set of propositional formulas and third the set of interpretations obtained by this merging is the subset of consistent scenarios resulting from the merging of the set of QCNs. Starting form the presupposition that our merging techniques provide an instrument to represent merging without taking into account integrity constraints of any kind, in some way, this approach is similar to our one in the translation from a formalism to another one. This is the same as what we do, we first take the original $EAFs$ and then we translate them in our meta-argumentation modelling language, in order to simplify the merging but always maintaining the correspondence between the EAF and the MAF.

Amgoud and Kaci [AK07]'s approach concerns the merging problem applied to conflicting knowledge bases. They propose an argumentation framework for solving conflicts arising between agents in a multiagent system. Supposing that each agent is represented by a knowledge base and that different agents are conflicting, the authors show that the argumentation framework retrieves the results of the merging approach. The aim of [AK07] is to establish a relationship between argumentation theory and propositional knowledge bases merging with priorities. The authors present a preference-based argumentation framework for reasoning with conflicting knowledge bases where each KB is associated to a separate agent. Each argument is seen as composed by a support H and a conclusion h, where H is a subset of the propositional formulas of the knowledge base and h is a propositional formula. Merging of AFs is done by means of merging operators used in the propositional logic framework. Roughly, a merging operator in propositional logic is a mapping which associates a propositional formula to a profile K and a propositional formula representing the integrity constraints. This approach builds arguments from separate KBs, evaluates them and computes a set of acceptable arguments from which conclusions are drawn. The argumentation framework captures the result of the merging operator without merging the different KBs. This approach differs from our one in the representation of arguments, here composed by premises and conclusions.

7.3 Coalition formation and dependence networks

Although there were many approaches defining coalition formation, two represents different perspectives: the model of Shehory and Kraus [SK98] and the one of Sichman [Sic98]. Shehory and Kraus [SK98] present algorithms that enable the agents to form groups and assign a task to each group, calling these groups coalitions. Sichman [Sic98] presents coalition formation using a dependence-based approach founded on the notion of social dependence, introduced by Castelfranchi [Cas03]. Boella *et al.* [BSvdT06] show how to use dependence networks to discriminate among different potential coalitions during the coalition formation process. In this work, the authors assume that a coalition is effectively formed only when all its members agree on it and they cannot deviate from what established in the agreement, once they decide to enter it. They develop a criterion of admissibility called *do-ut-des* property describing a condition of reciprocity: an agent gives a goal only if this fact enables it to obtain, directly or indirectly, the satisfaction of one of its own goals. Moreover, they define another criterion, called the *indecomposable do-ut-des* property, establishing which coalitions cannot be formed under the assumption that the agents are self-interested. In the indecomposable *do-ut-des* property, differently from the *do-ut-des* property, the decomposability of a coalition in independent sub-coalitions is considered as a discriminant for the admissibility of the coalition itself. These two criteria have only a qualitative connotation and thus, they cannot be directly applied to the solutions developed in game theory. In this approach goals are not structured and they do not represent explicitly the costs of the actions. See Sauro [Sau05] for a further discussion.

In Sauro *et al.* [SvdTV09], the authors propose a new step to make the computation of the core easier by means of the dependence networks associated to the cooperative boolean game introduced by Dunne *et al.* [DvdHKW08]. First, they present a number of abstractions that allow to reduce the search space by means of a set of criteria principally based on graphs' visit algorithms which are computationally tractable; second, they underline a number of hidden properties in the notion of core showing how, in certain cases, this notion is too much strict and, thus, it can lead to counterintuitive results. The authors define two kinds of dependence networks, representing two different levels of abstraction of a cooperative boolean game. Abstract dependence networks, already used by Bonzon *et al.* [BLSL07] to show that the notion of stability is complete with respect to the pure Nash equilibrium for non costly actions, are used to show that the notion of stability is complete also with respect of the solution concept of the core in the case of cooperative boolean games with costly actions. Refined dependence networks essentially provide a graph representation of a cooperative boolean game where the numerical information about costs is abstracted and actions are simply partitioned in free and costly actions. Sauro *et al.* [SvdTV09] present a reduction, called Δ-reduction, to pass from a cooperative boolean game G to a CBG G', simpler to be solved because less actions can be executed.

Once represented the internal structure of coalitions, one could study which kind of relations there are among candidate coalitions at an higher level of detail disregarding which are the causes for incompatibility.

The application of argumentation frameworks to coalition formation has been discussed by Amgoud [Amg05] and by Bulling *et al.* [BDC08]. The latter combines argumentation theory and ATL presenting a generalization of Dung's argumentation framework, extended with a preference relation. Alternating-time temporal logic is a temporal logic that is used for reasoning about the behavior and abilities of agents under various rationality assumptions. The key construct in ATL expresses that a coalition of agents can enforce a given formula [AHK02]. Amgoud [Amg05], instead, proposes to use an extension of Dung's argumentation theory with preferences and associated dialogue theories as a formal framework for coalition formation. As preferred extensions exist for every argumentation framework, we can introduce the preferred solutions to coalitional games by defining them as the preferred extensions of the corresponding argumentation system. Amgoud illustrates this idea by formalizing a task based theory of coalition formation, where the conflict relation represents that two coalitions contain the same task. However, a drawback of this abstract approach is that it is less clear where the preferences among coalitions come from. In contrast with our approach, a coalition is viewed as an abstract entity whose structure is not known. Unlike Amgoud's work, we do not provide a proof theory since it is derivable from the argumentation theory's literature. Another formal approach to reason about coalitions is coalition logic [Pau02] and ATL [AHK02], describing how a group of agents can achieve a set of goals, but without considering the internal structure of the group of agents [vdHJW05].

Second- and higher-order argumentation frameworks have been discussed in a modeling approach to argumentation by Boella *et al.* [BvdTV09a]. 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 Boella *et al.* [BvdTV09a], 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).

Chapter 8

Future Work

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 thesis, 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 of meta-arguments. For the X and Y meta-arguments discussed in this thesis, 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?

8.1 Subsumption relation

The formalization of the notion of subsumption in argumentation can be developed following different directions. First, a new kind of configuration has to be considered in order to provide an effective method to represent the subsumption relation. Thus far, we present and discuss various kinds of attack from and to the arguments involved in the subsumption relation. Finally, we present also the attack raised against the subsumption relation itself. The lacking step is a representation of an attack from the subsumption relation to another argument.

The case in which the subsumption relation attack another argument seems reasonable. Let us consider the example provided in Chapter 4 in which we have argument "icing and baking powder are necessary for making birthday cakes" which is subsumed by argument "baking powder is necessary for making cakes". This subsumption relation attacks argument "icing and baking powder are necessary for making birthday cakes but baking powder is not necessary for making cakes". The way we represent this attack is not immediate because the attack starts from an argument which is not directly represented in the argumentation framework. Moreover, another problem concerns the relation between the arguments involved in the subsumption relation and the argument attacked by the subsumption relations. We should investigate if these arguments too attack the argument attacked by the subsumption relation or not. Moreover, a further development about the introduction of the subsumption relation in meta-argumentation is the passage from this notion to the one of support, highlighting the constraints which allow to identify and distinguish the two kinds of relationship between arguments given an argumentation framework. Finally, the proposed approach to subsumption seems promising to be applied to the field of ontology in order to provide a way to reason about the notions of an ontology in an automatic way. This research direction should be deeply analyzed.

8.2 Merging views

In Chapter 5, we propose three techniques for modeling merging highlighting a number of possible applications of them. A topic which should be further investigate is related to cycle analysis. We restrict cycle analysis as application of only the second merging technique but it seems reasonable to do the same assumption also for technique three. Moreover, the postulates due to the existence of the stable extensions have to be proved and the consequences of these results deeply analyzed. Concerning technique three, further research should focus on the notion of trust we introduce in argumentation and improve what can be called the *trust network* relating it to a more dialogue based perspective, incorporating also the notion of coherence. This notion of coherence would come out from the arguments used by an agent which can be judged trustable due to the degree of coherence of his dialogue, of his network of arguments.

From a more general point of view, the application of merging in an epistemic context should be addressed to the coalition formation issue. The idea behind this application is that in order to form a coalition together, agents should have common ideas or at least a common list of aims. Merging the argumentation frameworks of the agents forming a coalition would lead to the formation of stable coalitions. In this context, a comparison between the three techniques would return what kinds of merging technique are better in order to keep a coalition stable.

A further development in the context of merging argumentation framework is the analysis and formalization of a modelling technique for dynamic merging. In a multiagent environment, agents may change their mind about arguments and their relationships and these changes have to be reflected by the merged argumentation framework in an incremental step that means that the merging process should not start every time frame from the beginning.

8.3 Coalition formation

We refine the abstract coalition models presented in Chapter 6 with powers of sets of agents and the conditional goals of the agents. We present two more refined viewpoints, the power view and the agent view, but the analysis of reciprocity based coalitions at these refined levels is left for future research. In classical planners, goals are unconditional. Therefore, many models of goal based reasoners, including the model of Boella *et al.* [BSvdT04], define the goals of a set of agents A by a function $goals : A \rightarrow 2^G$, where G is the complete set of goals. However, in many agent programming languages and architectures, goals are conditional and can be generated. The power to trigger a goal is distinguished from the power to fulfill a goal.

Definition 33 (Power view) *The Power view is represented by the tuple $\langle A, G, X, T, goals, power\text{-}goals, power\rangle$, where A, G, X and T are sets of agents, goals, decision variables, and time frames, $goals : A \rightarrow 2^G$, and $power\text{-}goals : 2^A \rightarrow 2^{(A\times G)}$ is a function associating with each set of agents the goals they can create for agents, and $power : 2^A \rightarrow 2^G$ is a function associating with agents the goals they can achieve.*

The function power represents what goals each agent or group of agents can achieve without being supported by other agents. For example, $power(\{a_1\}) = \{g_1\}$ means that agent a_1 is able to achieve g_1. Note that it is not given that g_1 is a goal of agent a_1. We therefore extend the agent view with conditional goals.

Definition 34 (Agent view) *The Agent view is represented by the tuple $\langle A, G, X, T, goals, skills, R\rangle$, where A, G, X, T are disjoint sets of agents, goals, decision variables, and time frames, goals is as before, $skills : A \rightarrow 2^X$ is a function associating with an agent its possible decisions, and $R : 2^X \rightarrow 2^G$ is a function associating with decisions the goals they achieve.*

The power view can be defined as an abstraction of the agent view. A set of agents B has the power to see to the goal g of agent a, written as $(a, g) \in \mathit{power\text{-}goals}(B)$, if and only if there is a set of decisions of B such that g becomes a goal of a. A set of agents B has the power to see to goal g if and only if there is a set of decisions of B such that g is a consequence of it.

Definition 35 *$\langle A, G, T, goals, power\text{-}goals, power\rangle$ is an abstraction from $\langle A, G, X, T, goals, skills, R\rangle$ if and only if: $(a, g) \in power\text{-}goals(B)$ if and only if $\exists Y \subseteq skills(B)$ with $skills(B) = \cup\{skills(b) \mid b \in B\}$ such that $g \in goals(a, Y)$, and $g \in power(B)$ if and only if $\exists Y \subseteq skills(B)$ such that $g \in R(Y)$.*

Abstracting the power view to a dynamic dependence network can be done as follows. Note that in this abstraction, the creation of a dynamic dependency is based only on the power to create goals. In other models, creating a dependency can also be due to the creation of new skills of agents.

Definition 36 *$\langle A, G, T, dyndep^-, dyndep^+, \geq\rangle$ is an abstraction of $\langle A, G, T, power\text{-}goals, power\rangle$, if we have $H \in dyndep^+(a, B, C)$ if and only if $\forall g \in H : (a, g) \in power\text{-}goals(C)$, and $H \subseteq power(B)$.*

Combining these two abstractions, abstracting the agent view to a dynamic dependence view can be done as follows.

Proposition 1 *$\langle A, G, T, dyndep^-, dyndep^+, \geq\rangle$ is an abstraction of $\langle A, G, X, T, goals, skills, R\rangle$, if we have $H \in dyndep^+(a, B, C)$ if and only if $\exists Y \subseteq skills(C)$ such that $H \subseteq goals(a, Y)$, and $\exists Y \subseteq skills(B)$ such that $H \subseteq R(Y)$.*

Arguing with the meta-argumentation methodology at these refined levels of abstraction is our main aim for future work concerning the topic of coalition formation. The approach we plan to apply will follow the examples provided in Boella *et al.* [BHvdT05a] and Amgoud and Prade [AP09], particularly concerning the agent view in which we describe an agent by means of features such as goals, beliefs and so on. The main difference concerns the power view which is not considered in these approaches and which has to be represented taking into account also the implicit notion of group present in it.

Subjects of further research concern also the use of our new theory for coalition formation. For example, when two agents can make the other depend on itself and thus create a potential coalition, when will they do so? Moreover, in this thesis we concentrate our attention on single coalitions. We aim at extending this model by considering more than one formed coalition which cooperates with other coalitions in order to achieve an increased outcome. From this point of view, the model represent each coalition as a node of an argumentation network in which coalitions have to manage attack decisions and coalitions can aggregate to each other due to their decisions and the achievable outcome represented for instance by a game.

8.4 Dependence networks

In standard argumentation networks, all the attacks are actual ones and the decisions of an argument not to attack another argument even if it has the possibility to do so have not been analyzed. We are working on a proposal of a new abstract model of argumentation network with voluntary attacks. We apply argumentation networks with voluntary attacks to the coalition formation problem in the context of coalitions represented with dependence networks, similarly of what is presented in Chapter 6. We present a mapping between dependence networks and voluntary argumentation networks and we see each agent and each goal as an argument which voluntarily attacks another argument, depending on a number of constraints. Due to these voluntary attacks, agents negotiate with each other to form coalitions in order to achieve a greater amount of goals. The negotiation process is under the form of a game using for instance the Nash equilibrium. It returns, after the computation of the labelings, depending on the set of activated voluntary attacks, the set of agents that will achieve their goals and the formed coalitions.

Moreover, we propose to use two kinds of attacks, disjunctive attacks and joint attacks introduced by Gabbay [Gab09b, Gab09a], in order to cover the possible kinds of dependencies composing the dependence network. Specifically, we represent by means of disjunctive attacks the multiple dependency of one agent on a group of agents for a unique goal while we represent as joint attacks the dependency of one agent on different agents for the same goal. The nodes of our argumentation network are of two types: the agent arguments and the goal arguments. Each agent is an unattacked argument and it can attack the goal arguments of the other agents. We propose to build a complete mapping between the dependence networks model and the argumentation networks with voluntary attacks. The passage from the dependence network to the argumentation network can be summarized as follows:

- For each agent in the dependence network build an agent argument g_{agent} and associate to this agent all the goals on which it depends on the other agents for their achievement.

- For each dependency, add a voluntary attack from the dependee to the goal of the depender which it can achieve.

The complete mapping between dependence networks and argumentation ones is provided in Figure 8.1, presenting all the possible configurations of dependencies and the resulting argumentation network with voluntary attacks.

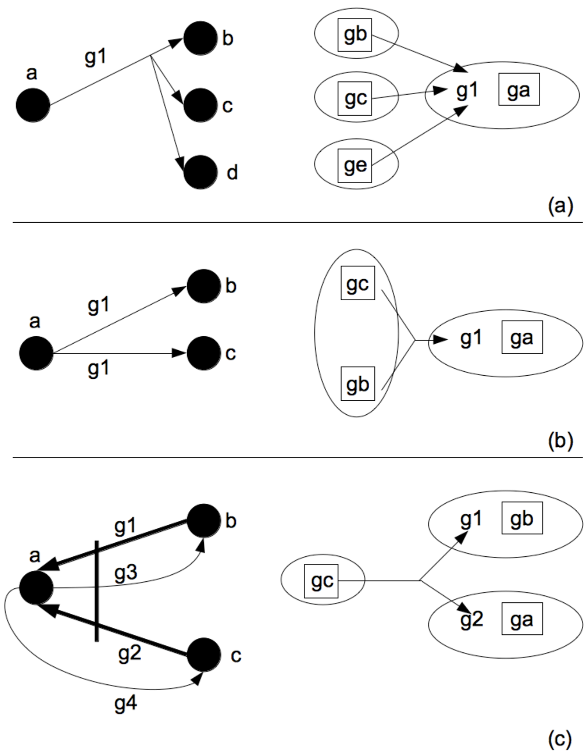


Figure 8.1: The mapping between dependence networks and voluntary argumentation networks.

In Figure 8.1-a, a dependence network with three agents is described. Agent a depends on agents b, c, d for goal g_1 which means that a needs both b and c and d for achieving the goal since this achievement depends, for example, on three actions which have to be done together and these actions can be performed by agents b, c, d. This is translated into three attacks from agents b, c, d to goal g_1 of agent a. If one of these agents attacks goal g_1, it is *out* since the goal is achieved only if every agent involved in the dependency relation does its action.

Figure 8.1-b presents three agents where agent a depends on agents b and c for goal g_1 since both the agents can achieve g_1 for agent a. In this case, the translation from the dependence network to the argumentation network consists in representing these two dependencies by means of a joint attack from the agent arguments g_b and g_c to the goal argument g_1. In this case, if one of the attacks is not raised then the joint attack is *out* thus the attacked argument is *in*.

Finally, in Figure 8.1-c are represented disjunctive attacks between one attacker and two attacked arguments. From the dependence networks point of view, the disjunctive attack is represented by two agents, b and c, which depend on agent a for goals g_1, g_2, respectively. The disjunctive issue is achieved by adding a straight line between these two dependencies (in bold in the figure) which means that agent a cannot achieve both of them. Agent a has to choose what goal he wants to attack (he will not achieve this goal) and what goal will survive. An intuitive example of this kind of application consists in home electricity where each home cannot have at the same time more than two or three household electrical appliances otherwise the electricity will go away.

This is a complete mapping between the two different kinds of network and it allows to pass from a representation to another one in order to highlight different aspects of the represented multiagent system. If we are interested in an analysis of the inter-relationships between the agents we analyze the dependence network while if we are more interested in a coalition formation process applied to a multiagent system, the argumentation network gives us a more appropriate representation. In every case, if you have one of the two kinds of network you can always achieve the representation of the multiagent system in the other kind of network.

Chapter 9

Conclusions

In this thesis, 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 conflict 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 [Dun95]. 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 what introduced by Dung, for which we use the Baroni and Giacomin framework [BG07] 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. The meta-argumentation methodology and techniques come from a research line addressed at the beginning of 2009 and they have been published in [BGvdTV09a, BvdTV09e].

We illustrate the methodology and techniques of meta-argumentation on three challenges in formal argumentation.

First, we show how meta-argumentation can be used for the representation of subsumption in argument ontologies. We analyze the consequences of attacks from and to a subsumption relation. Given that argument a is subsumed by argument b, we highlight how to model the fact that another agent c attacks b or b attacks c and so on. These kinds of attacks have numerous consequences on the arguments which are subsumed by argument b since new attack relations are inferred due to the existence of the subsumption relation. Finally, we consider also how to model an attack on the subsumption relation itself. Modeling subsumption, which is compared with the modeling of support relations between arguments, gives an example of extended argumentation framework used for instantiating Dung's abstract argumentation framework. Moreover, we show how meta-argumentation can be used for the representation of the well known Toulmin scheme [Tou58] when representing and combining micro arguments. We propose to represent the data D which supports the claim C with the warrant W by D is subsumed by C, where the absence of a warrant is equal to an attack on the subsumption relation. Rebuttals R are modeled as standard attacks on the claim and this is translated as an attack from meta-argument R to meta-argument C. Using the subsumption relation in order to express the support relation, we have that D is subsumed by C and if there is an attack from R to C, this is translated in an attack also on D, returning the extension $\{R\}$.

Second, we show how meta-argumentation can be used to model the merging of argumentation frameworks in multi-agent argumentation. We present three modeling techniques which allow to model merging of argumentation frameworks coming from different agents. Technique 1, the simplest one, merges the argumentation frameworks introducing an attack in the partial argumentation framework only of there is this attack relation in one of the starting argumentation frameworks and there is not the same non-attack relation. The second technique merges the argumentation frameworks using X and Y meta-arguments and it introduces arguments such as "argument a is acceptable" and "argument a is known", distinguishing them and providing constraints about the coexistence of the two arguments. A possible application of this technique is that of cycle analysis, where the odd/even cycle dilemma is a well known problem [BGG05, DBC02]. Technique 2 provides the conditions in which it is possible to have only even cycles for two-players games, such that stable extensions always exist. Technique 3, instead, introduces in the argumentation framework the agents under the form of arguments like "agent i is trustable". A possible application for this kind of technique concerns the introduction in multiagent argumentation of trust sources. We allow to distinguish different degrees of trustworthiness, such as distrust about an agent, distrust about an argument or distrust about an attack relation. Finally, a technique for merging second-order argumentation frameworks is presented. The application of the meta-argumentation methodology to the introduction of the subsumption relation and to the merging of argumentation systems has been developed during 2009 and it has been submitted recently to an international journal, after being the topic of some talks.

Third, we analyze reciprocity-based coalitions that emerge in social networks at various degrees of abstraction. We present an approach to iteratively design social networks by introducing four viewpoints, the refinement relations between them, and the methods to analyze cooperation based on emerging coalitions. At the most abstract viewpoint, coalitions are abstract entities and we adapt existing coalition argumentation theory to reason about these coalitions seen as arguments. We introduce a new meta-argument, called second-order argument, representing second-order attacks, preferring a coalition over the others. In this context, meta-argumentation allows to model the attacks among candidate coalitions and to decide whether a coalition could be formed. This analysis is refined in the dynamic dependence view providing the composition of each coalition and the motivations behind attack and preference relations. We refine abstract coalition models with social dynamic dependencies among agents, powers of sets of agents, and plans by making the dependence relation conditional to the agents that have the power to create or delete it. These dynamic dependencies are higher-order dependencies reflecting the behaviours of the more abstract higher-order attacks of the coalition view. A further refinement leads to the definition the power view and the agent view. The agent view is the most detailed view considering all the features of the single agents as facts, goals and skills but it looses the notion of "group" which is present, instead, in the power view, associating a *set of* agents to the goals they can achieve. The use of a meta-argumentation approach to coalition formation has been published in [BvdTV08a, BvdTV08d, BvdTV09a]. The social network approach to coalition formation has been the research topic of the second year of my PhD and it has been published in [BvdTV08c, BvdTV09d, BvdTV09b] while the research about dependence networks is published in [Vil08b, Vil08a, Vil09, BvdTV09c, SvdTV09, BBvdTV09, BGvdTV09b, Vil10].

The use of the X and Y meta-arguments leads to two challenges. First, if we like to model something, 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, how can or should we read the attacks among these meta-arguments? Both of these questions are addressed the definition of some concepts in terms of X and Y arguments in the thesis. Merging provides an answer to these challenges. For instance, in technique 2 for modeling merging, an attack like $X \rightarrow Y$ means that the AF of the individual agent has a non-attack relation, characterized by the X meta-argument, and this non-attack relation attacks the meta-argument Y, representing the same attack in the merged framework. In subsumption, X and Y meta-arguments are used in order to fire the inferred attacks between the arguments part of the subsumption relation and they are used also to represent the subsumption relation itself in the flattened argumentation network. In coalition formation, these two meta-arguments represents the attacks of first- and higher-order between candidate coalitions.

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. For example, if we introduce audiences [BC02] in our meta-argumentation theory, we can do objective acceptance but the question is does it make things easier or more complicated? On the other hand, there are other extensions which we do not discuss in this thesis, but which seem straightforward to model. For example, accrual of arguments by instantiating meta-arguments with sets of arguments, or even complete extensions.

Bibliography

[AB09] 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.

[ABC05] Katie Atkinson and Trevor J. M. Bench-Capon. Legal case-based reasoning as practical reasoning. *Artif. Intell. Law*, 13(1):93–131, 2005.

[ABCM06] Katie Atkinson, Trevor J. M. Bench-Capon, and Peter McBurney. Parmenides: Facilitating deliberation in democracies. *Artif. Intell. Law*, 14(4):261–275, 2006.

[AC02] 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.

[ACLSL08] 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.

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[AK07] Leila Amgoud and Souhila Kaci. An argumentation framework for merging conflicting knowledge bases. *Int. J. Approx. Reasoning*, 45(2):321–340, 2007.

[Amg05] Leila Amgoud. An argumentation-based model for reasoning about coalition structures. In Parsons et al. [PMMR06], pages 217–228.

[AP09] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artif. Intell.*, 173(3-4):413–436, 2009.

[BBvdTV09] Guido Boella, Jan Broersen, Leendert van der Torre, and Serena Villata. Representing excuses in social dependence networks. In Roberto Serra and Rita Cucchiara, editors, *Emergent Perspectives in Artificial Intelligence, Eleventh International Conference of the Italian Association for Artificial Intelligence, AI*IA*, volume 5883 of *Lecture Notes in Computer Science*, pages 365–374. Springer, 2009.

[BC98] Trevor J.M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *Legal Knowledge and Information Systems, The Eleventh Annual Conference, JURIX*, pages 5–20, 1998.

[BC02] Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In Salem Benferhat and Enrico Giunchiglia, editors, *Ninth International Workshop* on *Non-Monotonic Reasoning, NMR*, pages 443–454, 2002.

[BC03] T.J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Logic and Computation*, 13(3):429–448, 2003.

[BCD07] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[BCGG09] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Encompassing attacks to attacks in abstract argumentation frameworks. In Sossai and Chemello [SC09], pages 83–94.

[BCPT06] Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors. *ECAI 2006, 17th European Conference on Artificial Intelligence, August 29 - September 1, 2006, Riva del Garda, Italy, Including Prestigious Applications of Intelligent Systems (PAIS 2006), Proceedings*, volume 141 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2006.

[BDC08] Nils Bulling, Jürgen Dix, and Carlos Iván Chesñevar. Modelling coalitions: Atl + argumentation. In Padgham et al. [PPMP08], pages 681–688.

[BDKT97] 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.

[BE09] Gerhard Brewka and Thomas Eiter. Argumentation context systems: A framework for abstract group argumentation. In Esra Erdem, Fangzhen Lin, and Torsten Schaub, editors, *LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2009.

[BG03] Pietro Baroni and Massimiliano Giacomin. Solving semantic problems with odd-length cycles in argumentation. In Thomas D. Nielsen and Nevin Lianwen Zhang, editors, *ECSQARU*, volume 2711 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 2003.

[BG07] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

[BGG05] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artif. Intell.*, 168(1-2):162–210, 2005.

[BGvdTV09a] Guido Boella, Dov M. Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling i: Methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[BGvdTV09b] Guido Boella, Valerio Genovese, Leendert van der Torre, and Serena Villata. Checking consistency in role oriented dependence networks. In *Twenty-first Belgian-Netherlands Conference on Artificial Intelligence, BNAIC*, pages 27–34, 2009.

[BGW05] Howard Barringer, Dov M. Gabbay, and John Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. *Mechanizing Mathematical Reasoning*, pages 59–98, 2005.

[BH09a] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. The MIT Press, 2009.

[BH09b] Felix Brandt and Paul Harrenstein. Characterization of dominance relations in finite coalitional games. *Theory and Decision*, (to appear), 2009.

[BHH08] The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors. *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents, PRIMA 2008, Hanoi, Vietnam, December 15-16, 2008. Proceedings*, volume 5357 of *Lecture Notes in Computer Science*. Springer, 2008.

[BHvdT05a] Guido Boella, Joris Hulstijn, and Leendert W. N. van der Torre. Argumentation for access control. In *Advances in Artificial Intelligence, Ninth Congress of the Italian Association for Artificial Intelligence, AI*IA*, pages 86–97, 2005.

[BHvdT05b] Guido Boella, Joris Hulstijn, and Leendert W. N. van der Torre. A logic of abstract argumentation. In Parsons et al. [PMMR06], pages 29–41.

[BKvdT09a] 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 [SC09], pages 107–118.

[BKvdT09b] Guido Boella, Souhila Kaci, and Leendert van der Torre. Dynamics in argumentation with single extensions: attack refinement and the grounded extension. In Sierra et al. [SCDS09], pages 1213–1214.

[BLSL07] Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang. Dependencies between players in boolean games. In Mellouli [Mel07], pages 743–754.

[Boc03] Alexander Bochman. Collective argumentation and disjunctive logic programming. *J. Log. Comput.*, 13(3):405–428, 2003.

[Boc05] A. Bochman. *Explanatory Nonmonotonic Reasoning.* World Scientific Publishing, 2005.

[BPG+04] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[BSvdT04] Guido Boella, Luigi Sauro, and Leendert W. N. van der Torre. Social viewpoints on multiagent systems. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1358–1359. IEEE Computer Society, 2004.

[BSvdT06] Guido Boella, Luigi Sauro, and Leendert W. N. van der Torre. Strengthening admissible coalitions. In Brewka et al. [BCPT06], pages 195–199.

[BvdTV08a] Guido Boella, Leendert van der Torre, and Serena Villata. Attack relations among dynamic coalitions. In

Twenty Belgian-Netherlands Conference on Artificial Intelligence, BNAIC, pages 25–32, 2008.

[BvdTV08b] Guido Boella, Leendert van der Torre, and Serena Villata. Changing institutional goals and beliefs of autonomous agents. In Bui et al. [BHH08], pages 78–85.

[BvdTV08c] Guido Boella, Leendert van der Torre, and Serena Villata. Self adaptive coalitions in multiagent systems. In Sven A. Brueckner, Paul Robertson, and Umesh Bellur, editors, *Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO*, pages 461–462. IEEE Computer Society, 2008.

[BvdTV08d] Guido Boella, Leendert van der Torre, and Serena Villata. Social viewpoints for arguing about coalitions. In Bui et al. [BHH08], pages 66–77.

[BvdTV09a] Guido Boella, Leendert van der Torre, and Serena Villata. Analyzing cooperation in iterative social network design. *Journal of Universal Computer Science*, (to appear), 2009.

[BvdTV09b] Guido Boella, Leendert van der Torre, and Serena Villata. Coalitions dynamics in requirements engineering: a social network approach. In *Second Multi-Agent Logics, Languages, and Organizations Federated Workshops, MALLOW*, volume 494, 2009.

[BvdTV09c] Guido Boella, Leendert van der Torre, and Serena Villata. Conditional dependence networks in requirements engineering. In *Coordination, Organizations, Institutions and Norms in Agent Systems, COIN*. LNCS, Springer (to appear), 2009.

[BvdTV09d] Guido Boella, Leendert van der Torre, and Serena Villata. Four measures for the dynamics of coalitions in social networks. In Ciro Cattuto, Giancarlo Ruffo, and Filippo Menczer, editors, *Hypertext*, pages 361–362. ACM, 2009.

[BvdTV09e] Guido Boella, Leendert van der Torre, and Serena Villata. On the acceptability of meta-arguments. In *The 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT*, pages 259–262. IEEE, 2009.

[CA07] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[Cam06] 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.

[Cas03] C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.

[CKMS09] Jean-François Condotta, Souhila Kaci, Pierre Marquis, and Nicolas Schwind. Merging qualitative constraints networks using propositional logic. In Sossai and Chemello [SC09], pages 347–358.

[CLS05] 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.

[CMDK+07] Sylvie Coste-Marquis, Caroline Devred, Sébastien Konieczny, Marie-Christine Lagasquie-Schiex, and Pierre Marquis. On the merging of dung's argumentation systems. *Artif. Intell.*, 171(10-15):730–753, 2007.

[CP09] M.W.A. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, (to appear), 2009.

[CVBvdT08] Patrice Caire, Serena Villata, Guido Boella, and Leendert van der Torre. Conviviality masks in multiagent systems. In Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors, *Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS (3)*, pages 1265–1268. IFAAMAS, 2008.

[CvdT09] Patrice Caire and Leendert van der Torre. Temporal dependence networks for the design of convivial multiagent systems. In Sierra et al. [SCDS09], pages 1317–1318.

[DBC02] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artif. Intell.*, 141(1/2):187–203, 2002.

[DDK+05] Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors. *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS*

2005), July 25-29, 2005, Utrecht, The Netherlands. ACM, 2005.

[DMT07] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[Dun95] 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.

[Dun07] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[DvdHKW08] Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus, and Michael Wooldridge. Cooperative boolean games. In Padgham et al. [PPMP08], pages 1015–1022.

[FC05] Rino Falcone and Cristiano Castelfranchi. Socio-cognitive model of trust. In Mehdi Khosrow-Pour, editor, *Encyclopedia of Information Science and Technology (V)*, pages 2534–2538. Idea Group, 2005.

[Gab09a] Dov Gabbay. Semantics for higher level attacks in extended argumentation frames. part 1: overview. *Studia Logica*, 93(2-3):357–381, 2009.

[Gab09b] Dov M. Gabbay. Fibring argumentation frames. *Studia Logica*, 93(2-3):231–295, 2009.

[GJOW02] Gabbay, Johnson, Ohlbach, and Woods, editors. *Handbook of the logic of argument and inference: the turn towards the practical.* Elsevier Science, 2002.

[GMD05] Davide Grossi, John-Jules Ch. Meyer, and Frank Dignum. Modal logic investigations in the modal logic investigations in the semantics of counts semantics of counts-as as. In *Tenth International Conference on Artificial Intelligence and Law, ICAIL*, pages 1–9. ACM, 2005.

[Han01] Sven Ove Hansson. Preference logic. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, pages 319–387. Kluwer Academic Publishers, 2001.

[HR05] R. Hanneman and M. Riddle. *Introduction to social network methods.* University of California, Riverside, 2005.

[Hun08] Anthony Hunter. Reasoning about the appropriateness of proponents for arguments. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 89–94. AAAI Press, 2008.

[HV07a] David Hitchcock and Bart Verheij, editors. *Arguing On The Toulmin Model: New Essays In Argument Analysis And Evaluation.* Springer, 2007.

[HV07b] David Hitchcock and Bart Verheij. Introduction. In *Arguing On The Toulmin Model: New Essays In Argument Analysis And Evaluation*, pages 1–23. 2007.

[JV99] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9(2):215–261, 1999.

[KM03] Antonis C. Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS*, pages 883–890. ACM, 2003.

[KvdT08] Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *Int. J. Approx. Reasoning*, 48(3):730–751, 2008.

[KvdTW06] Souhila Kaci, Leendert W. N. van der Torre, and Emil Weydert. Acyclic argumentation: Attack = conflict + preference. In Brewka et al. [BCPT06], pages 725–726.

[KvdTW07] Souhila Kaci, Leendert W. N. van der Torre, and Emil Weydert. On the acceptability of incompatible arguments. In Mellouli [Mel07], pages 247–258.

[LEW00] 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.

[Lou07] Ron Loui. A citation based reflection on Toulmin and argument. In David Hitchcock and Bart Verheij, editors, *Arguing On The Toulmin Model: New Essays In Argument Analysis And Evaluation*, pages 31–38. 2007.

[MBC08] Sanjay Modgil and Trevor J. M. Bench-Capon. Integrating object and meta-level value based argumentation. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *COMMA*, volume 172 of *Frontiers in*

Artificial Intelligence and Applications, pages 240–251. IOS Press, 2008.

[Mel07] 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.

[Mod06] Sanjay Modgil. Value based argumentation in hierarchical argumentation frameworks. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *COMMA*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 297–308. IOS Press, 2006.

[Mod07] Sanjay Modgil. An abstract theory of argumentation that accommodates defeasible reasoning about preferences. In Mellouli [Mel07], pages 648–659.

[Mod09] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[MvdT00] D. Makinson and L. van der Torre. Input-output logics. *J. Philosophical Logic*, 29(4):383–408, 2000.

[OLMN08] Nir Oren, Michael Luck, Simon Miles, and Timothy J. Norman. An argumentation inspired heuristic for resolving normative conflict. In *Coordination, Organizations, Institutions and Norms in Agent Systems, COIN*, pages 41–56, 2008.

[OLN08] Nir Oren, Michael Luck, and Timothy J. Norman. Argumentation for normative reasoning. In *Proceedings of the Symposium on Behaviour Regulation in Multi-Agent Systems, BRMAS*, pages 55–60, 2008.

[Pau02] M. Pauly. A modal logic for coalitional power in games. *J. of Logic and Computation*, 12:146–166, 2002.

[PMMR06] 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.

[PPMP08] Lin Padgham, David C. Parkes, Jörg Müller, and Simon Parsons, editors. *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*. IFAAMAS, 2008.

[Pra09] 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.

[PS99] Henry Prakken and Giovanni Sartor. A system for defeasible argumentation, with defeasible priorities. In *Artificial Intelligence Today: Recent Trends and Developments. Lecture Notes in Computer Science*, volume 1600, pages 365–379. Springer, 1999.

[PTG08] Duy Hoang Pham, Subhasis Thakur, and Guido Governatori. Settling on the group's goals: An n-person argumentation game approach. In Bui et al. [BHH08], pages 328–339.

[PV02] H. Prakken and G. Vreeswijk. *Logics for defeasible argumentation.* Handbook of Philosophical Logic, Kluwer Academic Publishers, 2002.

[Sau05] Luigi Sauro. *Formalizing admissibility criteria in coalition formation among goal directed agents.* PhD thesis, University of Turin, 2005.

[Sav54] L.J. Savage. *The Foundations of Statistics.* Dover Publications, New York, 1954.

[SC02] Jaime Simão Sichman and Rosaria Conte. Multi-agent dependence by dependence graphs. In *AAMAS*, pages 483–490. ACM, 2002.

[SC09] 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.

[SCD+08] Richard O. Sinnott, David W. Chadwick, Thomas Doherty, David Martin, Anthony Stell, Gordon Stewart, Linying Su, and J. P. Watt. Advanced security for virtual organizations: The pros and cons of centralized vs decentralized security models. In *Eighth IEEE International Symposium on Cluster Computing and the Grid, CCGRID*, pages 106–113. IEEE Computer Society, 2008.

[SCDS09] Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman, editors. *8th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 2*. IFAAMAS, 2009.

[Sea69] J. Searle. *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[Sea95] J. Searle. *The Construction of Social Reality*. New York: The Free Press, 1995.

[Sic98] Jaime Simão Sichman. Depint: Dependence-based coalition formation in an open multi-agent scenario. *J. Artificial Societies and Social Simulation*, 1(2), 1998.

[SK98] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artif. Intell.*, 101(1-2):165–200, 1998.

[SvdTV09] Luigi Sauro, Leendert van der Torre, and Serena Villata. Dependency in cooperative boolean games. In Anne Håkansson, Ngoc Thanh Nguyen, Ronald L. Hartung, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES-AMSTA*, volume 5559 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2009.

[Tou58] Stephen Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[vdHJW05] Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. A logic for strategic reasoning. In Dignum et al. [DDK+05], pages 157–164.

[Ver03] Bart Verheij. Artificial argument assistants for defeasible argumentation. *Artif. Intell.*, 150(1-2):291–324, 2003.

[Vil08a] Serena Villata. Institutional social dynamic dependence networks. In Guido Boella, Gabriella Pigozzi, Munindar P. Singh, and Harko Verhagen, editors, *NORMAS*, pages 201–215, 2008.

[Vil08b] Serena Villata. Towards social dynamic dependence networks for institutions. In Amedeo Cesta and Nikos Fakotakis, editors, *STAIRS*, volume 179 of *Frontiers in Artificial Intelligence and Applications*, pages 187–198. IOS Press, 2008.

[Vil09] Serena Villata. Normas-re: a normative multiagent approach to requirements engineering. In Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, editors, *Normative Multi-Agent Systems*, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[Vil10] Serena Villata. A normative multiagent approach to requirements engineering. *Logic Journal of the IGPL*, (to appear), 2010.

[WC09] Yining Wu and Martin Caminada. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. In *Twenty-first Belgian-Netherlands Conference on Artificial Intelligence, BNAIC*, pages 51–57, 2009.

[WG01] Christopher A. Welty and Nicola Guarino. Supporting ontological analysis of taxonomic relationships. *Data Knowl. Eng.*, 39(1):51–74, 2001.

[WMP05] Michael Wooldridge, Peter McBurney, and Simon Parsons. On the meta-logic of arguments. In Dignum et al. [DDK+05], pages 560–567.