ArgMAS 2011

Eighth International Workshop

on

Argumentation in Multi-Agent Systems

Taipei, Taiwan, May 2011

In conjunction with AAMAS 2011

Workshop Proceedings

Editors:

Peter McBurney, Simon Parsons and Iyad Rahwan

ArgMAS 2011 PROGRAM COMMITTEE

Leila Amgoud, IRIT, Toulouse, France
Katie Atkinson, University of Liverpool, UK
Jamal Bentahar, Concordia University, Canada
Elizabeth Black, Oxford University, UK
Guido Boella, Università di Torino, Italy
Carlos Chesnevar, Universidad Nacional del Sur, Argentina
Yannis Dimopoulos, University of Cyprus, Cyprus
Sylvie Doutre, University of Toulouse 1, France
Paul Dunne, University of Liverpool, UK
Rogier van Eijk, Utrecht University, The Netherlands
Frank Guerin, University of Aberdeen, UK
Anthony Hunter, University College London, UK
Antonis Kakas, University of Cyprus, Cyprus
Nikos Karacapilidis, University of Patras, Greece
Nicolas Maudet, Universite Paris Dauphine, France
Peter McBurney, King's College London, UK
Jarred McGinnis, London, UK
Sanjay Modgil, King's College London, UK
Pavlos Moraitis, Paris Descartes University, France
Nir Oren, University of Abderdeen, UK
Fabio Paglieri, ISTC-CNR, Roma IT
Simon Parsons, Brooklyn College, City University of New York, USA
Enric Plaza, Spanish Scientific Research Council, Spain
Henry Prakken, Utrecht University, & University of Groningen, The Netherlands
Iyad Rahwan, Masdar Institute, UAE, & MIT, MA, USA
Michael Rovatsos, University of Edinburgh, UK
Chris Reed, University of Dundee, UK
Hajime Sawamura, Niigata University, Japan
Guillermo Simari, Universidad Nacional del Sur, Argentina
Francesca Toni, Imperial College, London, UK
Leon van der Torre, University of Luxembourg, Luxembourg
Paolo Torroni, Università di Bologna, Italy
Bart Verheij, University of Groningen, The Netherlands
Gerard Vreeswijk, Utrecht University, The Netherlands
Douglas Walton, University of Winnipeg, Canada
Simon Wells, University of Dundee, UK
Michael Wooldridge, University of Liverpool, UK.

ArgMAS STEERING COMMITTEE

Antonis Kakas, University of Cyprus, Cyprus
Nicolas Maudet, Universite Paris Dauphine, France
Peter McBurney, King's College London, UK
Pavlos Moraitis, Paris Descartes University, France
Simon Parsons, Brooklyn College, City University of New York, USA
Iyad Rahwan, Masdar Institute, UAE, and MIT, USA
Chris Reed, University of Dundee, UK

Contents

Paper Presentations

Martin Caminada, Gabriella Pigozzi and Mikolaj Podlaszewski:
Manipulation in group argument evaluation 1

Cristian Gratie and Adina Maqda Florea:
Fully labeling for argumentation frameworks 18

Antonis Kakas, Leila Amgoud, Gabriele Kern-Isberner, Nicolas Maudet and Pavlos Moraitis:
ABA: Argumentation-based agents 26

Ioan Alfred Letia and Adrian Groza:
Arguing with justifications between collaborating agents 44

Yoshifumi Maruyama, Taichi Hasegawa, Hajime Sawamura and Takeshi Hagiwara:
Syncretic argumentation for multi-agents by lattice homomorphism and fusion 58

Simon Parsons, Elizabeth Sklar and Peter McBurney:
Using argumentation to reason with and about trust 76

Samy Sa:
Position Statement: Arguments cooperatively built by groups of agents 95

Elizabeth Sklar and M. Q. Azhar:
Toward the application of argumentation to interactive learning systems 97

Kazuko Takahashi and Yu Nambu:
Dynamic semantics for dynamic argumentation framework 115

Serena Villata, Guido Boella and Leon van der Torre:
Argumentation patterns 133

Tom van der Weide and Frank Dignum:
Reasoning about and discussing preferences between arguments 151

Preface

Welcome to the eighth edition of the *International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2011)*, being held in Taipei, Taiwan, in association with the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2011). Previous ArgMAS workshops have been held in New York City (2004), Utrecht (2005), Hakodate (2006), Honolulu (2007), Estoril (2008), Budapest (2009), and Toronto (2010), and the event has now established itself on the international calendar among researchers in computational argument and dialectics.

This document contains the proceedings of the workshop, comprising 11 papers and position statements selected following a rigorous peer-review process. We thank all authors who made submissions to ArgMAS 2011, and we thank the members of the Programme Committee for their efforts in reviewing the papers submitted. We also thank the two anonymous reviewers selected by Iyad Rahwan to review the submission of two of the editors in a process of indirection. The papers presented at the workshop are assembled in this document in alphabetical order of the surname of the first author.

Following successful experiences in the previous two years, we hope again this year to have selected official respondents offer short critiques to several of the papers presented at ArgMAS 2011. We have adopted this innovation from conferences in Philosophy, where it is standard, and we found it worked will in stimulating discussion. We thank the official respondents for their willingness to undertake this task.

We hope that you enjoy the workshop, the conference overall, and your time in Taiwan.

Peter McBurney, Simon Parsons and Iyad Rahwan

Programme Co-Chairs

May 2011

Manipulation in group argument evaluation

Martin Caminada[1], Gabriella Pigozzi[2], and Mikołaj Podlaszewski[1]

[1] Individual and Collective Reasoning, Computer Science and Communication, University of Luxembourg, 6, Rue Richard Coudenhove Kalergi, L-1359 – Luxembourg
[2] LAMSADE, Université Paris-Dauphine, Place du Marchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
{martin.caminada}@uni.lu
{gabriella.pigozzi}@lamsade.dauphine.fr
{mikolaj.podlaszewski}@gmail.com

Abstract. Given an argumentation framework and a group of agents, the individuals may have divergent opinions on the status of the arguments. If the group needs to reach a common position on the argumentation framework, the question is how the individual evaluations can be mapped into a collective one. This problem has been recently investigated by Caminada and Pigozzi, who introduced and studied three aggregation operators that guarantee a collective outcome compatible with the individuals' positions. In this paper, we investigate the behaviour of two of such operators from a social choice-theoretic point of view. In particular, we study under which conditions these operators are Pareto optimal and whether they are manipulable. Our findings cast light on a virtuous type of manipulation, where - by lying - an agent increases not only its personal utility but also promotes the social welfare.

Keywords: Group decisions, Argumentation, Collective argument evaluation, Pareto optimality, Manipulation

1 Introduction

Individuals may draw different conclusions from the same information. For example, members of a jury may disagree on the verdict even though each member possesses the same information on the case under discussion. This happens because individuals can hold different reasonable positions on the information they share. Hence, the question is how the group can reach a common stance starting from the positions of each member.

In this paper we are interested in group decision-making where members share the same information. One of the principles of argumentation theory is that an argumentation framework can have several extensions/labellings. If the information the group shares is represented by an argumentation framework, and each agent's reasonable position is an extension/labelling of that argumentation framework, the question is how to aggregate the individual positions into a collective one. Caminada and Pigozzi [3] have studied this issue. They formalised the problem of the aggregation of individual labellings using the judgment aggregation framework [16, 9, 17], a research line shared also by Rahwan and Tohmé [24]. This approach is justified by the similarity between the problem of aggregating individual judgments on a given set of propositions, and that of aggregating individual labellings on a given argumentation framework.

Formal models of judgment aggregation combine logic with an axiomatic approach in the social choice tradition [1, 25]. As the famous impossibility theorem of Arrow showed that there exists no aggregation function that assigns a collective preference ordering to a set of individual preference orderings, and that meets some minimal conditions, so impossibility results in judgment aggregation showed that there exists no judgment aggregation function satisfying similar minimal conditions [15, 19, 9]. Correspondingly, Rahwan and Tohmé provided a counter-part of such impossibility result for abstract argumentation theory [24].

Following the tradition of social choice theory, the relaxation of one of the conditions imposed on the aggregation function constitutes a possible escape route to the impossibility results. While Rahwan and Tohmé explored the restriction of the space of possible individual judgments, Caminada and Pigozzi ensured collective rationality by dropping the condition of independence of irrelevant alternatives. Intuitively, this condition ensures that the collective position on each argument depends only on the individual positions on that argument. In preference and in judgment aggregation settings, the independence condition is defended by appealing to non-manipulability [10]. However, [3] did not study the consequences of dropping independence. Aim of this paper is precisely to fill in the gaps and to examine the impact of dropping the independence condition.

In order to investigate the consequences of relaxing the independence condition, we study here the behaviour of the aggregation operators introduced in [3] from a social choice-theoretic point of view. The key property of the three aggregation operators in [3] is that the collective outcome is ‘compatible’ with each individual position. That is, an agent who has to defend the collective position in public will never have to argue directly against his own private position.

We start here by formalising and examining the intuition that, although every social outcome that is compatible with one’s own labelling is acceptable, some outcomes are more acceptable than others. That is, a collective outcome is more acceptable than another if it is compatible and more similar to one’s own position than the other. In order to capture how much the various possible positions differ from each other, we use the notion of distance among labellings. Distance-based approaches have already been used to tackle aggregation problems, like in social choice theory [18, 13, 2], belief merging [14] and its application to judgment aggregation [20]. Thus, we say that a collective outcome is more acceptable than another if it is compatible, but the distance to one’s own labelling is smaller than the other.

The observations above give rise to two new research questions, to be addressed in the current paper:

(i) Are the social outcomes of the aggregation operators in [3] Pareto optimal if preferences between different (compatible) outcomes are also taken into account?

(ii) Do agents have an incentive to misrepresent their own opinion in order to obtain a more favourable outcome? And if so, what are the effects of this from the perspective of social welfare (that is, on the utility of the outcome for the other agents)?

We focus on the behaviour of two of the three aggregation operators defined in [3]. The object of study of welfare economics is the well-being of a society. Pareto optimality is a key principle of welfare economics which intuitively stipulates that a social state cannot be further improved. When comparing two possible social outcomes, an outcome is called Pareto optimal if it is not possible to make one individual better off without making at least one other person worse off. In our approach, an outcome is a possible collective position. Thus, the first contribution of the paper is to study whether the compatible social outcomes selected by our aggregation operators are Pareto optimal. In order to investigate Pareto optimality, we consider the submitted labelling as the individual's most preferred option. By using a notion of distance, we derive the individual preference ordering over the other permissible labellings. We show that the two aggregation operators are Pareto optimal, when a certain distance is used.

The second contribution is on the manipulability of the aggregation operators. Manipulability is usually considered to be an undesirable property of social choice decision rules. If an aggregation rule is manipulable, an individual may, upon learning the preferences of the other agents, misrepresent his input to ensure a social outcome that is better for him than it would have been had he voted sincerely. Our findings show that, while the two operators are manipulable, the sceptical aggregation operator guarantees that an agent who lies does not only ensure a preferable outcome for himself, but even promotes social welfare, what we call a *benevolent lie*.

The paper is structured as follows: Section 2 is devoted to outlining the abstract argumentation framework. In Section 3 we define preferences over the individual evalutaions. Pareto optimality and the manipulability issues are addressed in Section 4 and 5 respectively. Section 6 discusses the related work, and Section 7 concludes the paper.

The current paper should be seen as a full version of an extended abstract presented at the AAMAS 2011 main conference [4].

2 Preliminaries

2.1 Argumentation preliminaries

Definition 1 (Argumentation framework). *Let U be the universe of all possible arguments. An* argumentation framework *is a pair (Ar, def) where Ar is a finite subset of U and $def \subseteq Ar \times Ar$.*

We say that an argument A *defeats* (or *attacks*) an argument B iff $(A, B) \in def$. For example, in Fig. 1, we have that A attacks B and that B attacks C.

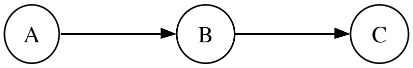


Fig. 1. An argumentation framework.

Following [3], we use the argument labellings approach of [5][3] rather than Dung's original extension approach [11]. The idea of a labelling is to associate with each argument exactly one label, which can either be `in`, `out` or `undec`. The label `in` indicates that the argument is explicitly accepted, the label `out` indicates that the argument is explicitly rejected, and the label `undec` indicates that one abstains from an explicit position on the argument.

Definition 2 (Labelling). *Let* (Ar, def) *be an argumentation framework. A* labelling *is a total function* $\mathcal{L} : Ar \longrightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$.

We write $\mathtt{in}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{in}\}$, $\mathtt{out}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{out}\}$ and $\mathtt{undec}(\mathcal{L})$ for $\{A \mid \mathcal{L}(A) = \mathtt{undec}\}$. Sometimes, we write a labelling $\mathcal{L}$ as a triple $(\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3)$ where $\mathcal{A}rgs_1 = \mathtt{in}(\mathcal{L})$, $\mathcal{A}rgs_2 = \mathtt{out}(\mathcal{L})$ and $\mathcal{A}rgs_3 = \mathtt{undec}(\mathcal{L})$. In some cases, it only matters whether an agent has a clear position (`in` or `out`) on an argument or whether he abstains. We then write $\mathtt{dec}(\mathcal{L})$ for $\mathtt{in}(\mathcal{L}) \cup \mathtt{out}(\mathcal{L})$.

Typically, given an argumentation framework, there exists more than one possible labelling, but not all labellings are reasonable. Several semantics have been defined in the literature, but we will consider only the following ones.[4]

Definition 3 (Illegal arguments). *Let* $\mathcal{L}$ *be a labelling of argumentation framework* (Ar, def) *and let* $A \in Ar$. *We say that:*

1. *A is* illegally `in` *(in* $\mathcal{L}$*) iff* $\mathcal{L}(A) = \mathtt{in}$ *and* $\exists B \in Ar : (B def A \wedge \mathcal{L}(B) \neq \mathtt{out})$
2. *A is* illegally `out` *(in* $\mathcal{L}$*) iff* $\mathcal{L}(A) = \mathtt{out}$ *and* $\neg\exists B \in Ar : (B def A \wedge \mathcal{L}(B) = \mathtt{in})$
3. *A is* illegally `undec` *(in* $\mathcal{L}$*) iff* $\mathcal{L}(A) = \mathtt{undec}$ *and* $\exists B \in Ar : (B def A \wedge \mathcal{L}(B) = \mathtt{in}) \vee \forall B \in Ar : (B def A \supset \mathcal{L}(B) = \mathtt{out})$.

For example, argument A of Fig. 1 is `in` as it has no defeaters. Argument B must be `out` since it has a defeater (argument A) and its defeater is `in`. Finally, argument C is `in` since its only defeater (argument B) is `out`.

Definition 4 (Admissible labelling). *An* admissible labelling *is a labelling without arguments that are illegally* `in` *and without arguments that are illegally* `out`.

Definition 5 (Complete labelling). *A* complete labelling *is a labelling without arguments that are illegally* `in`*, without arguments that are illegally* `out` *and without arguments that are illegally* `undec`.

Intuitively, an admissible labelling ensures that an agent has reasons for each `in` and `out` positions on an argument, and finally, a complete labellings adds the condition that an agent cannot remain undecided if he has a reason to accept or reject an argument.

Four labellings of an argumentation framework are depicted in Fig. 2. Labellings $\mathcal{L}_1$ and $\mathcal{L}_2$ are admissible and complete. Labelling $\mathcal{L}_3$ is admissible but not complete because argument C is illegally `undec` as its attacker, argument B is `in`. Labelling $\mathcal{L}_4$ is not admissible and not complete because argument C is illegally `out`.

[3] Other labellings approaches are [21, 27, 28].

[4] For an explanation of why this is not a too restrictive assumption, the reader is referred to [3].

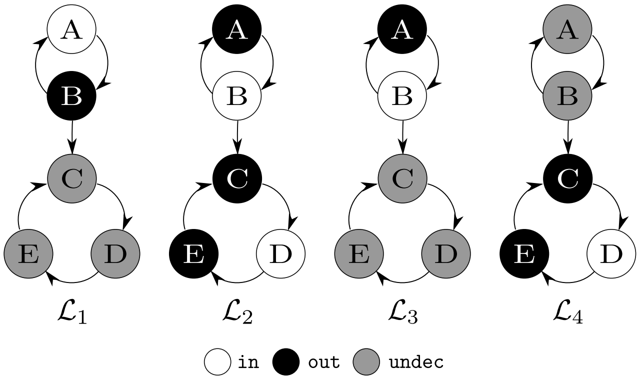


Fig. 2. The four labellings of a simple argumentation framework.

The basic difference between an extension [11] and a labelling [5, 6] is that an extension only represents the arguments that are accepted, whereas a labelling also represents the arguments that are rejected or left undecided. Well-known results are that the set of in-labelled arguments of an admissible labelling (complete labelling) is an admissible set (complete extension) in the sense of [11]. Moreover, for each admissible set (complete extension) there exists an admissible labelling (complete labelling) of which set of in-labelled arguments is precisely the admissible set (complete extension). Finally, for admissible labellings/sets the relationship is many-to-one, whereas for complete labellings/extensions the relationship is one-to-one. For more details on how labellings relate to extensions, see [5, 6].

So the idea is that labellings provide a slightly more expressive though equivalent way to express Dung's theory of argumentation. Whereas Dung's original theory focusses only on the sets of accepted arguments, with labellings one can also refer explicitly to the sets of rejected arguments and the sets of arguments where one does not have an explicit opinion about.

In essence, a labelling based semantics can be seen as a function that, given an argumentation framework, yields zero or more labellings, each of which can be seen as a reasonable position that one can take in the presence of the argumentation framework.

Definition 6 (Labelling based semantics). *Let $\mathcal{AF}$ be the set of all possible argumentation frameworks using a universe U. Let $\mathcal{L}abellings$ be $\{\mathcal{L} \mid$ there exists an argumentation framework $AF \in \mathcal{AF}$ such that $\mathcal{L}$ is a labelling of $AF\}$. A* labelling based semantics *is a function $\mathcal{T} : \mathcal{AF} \to 2^{\mathcal{L}abellings}$.*

2.2 Aggregation problem

We are now ready to formally summarize the problem of aggregation of individual labellings into a collective position on a given argumentation framework. The definitions and results in this section are from [3].

Given a set of individuals $N = \{1, \ldots, n\}$, we need to define a general labellings aggregation operator O_{AF} that assigns a collective labelling $\mathcal{L}_{Coll}$ to each profile $P = \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$ of individual labellings.

Definition 7 (Labelling aggregation operator O_{AF}). *Let $\mathcal{L}abellings$ be the set of all possible labellings of argumentation framework $AF = (Ar, def)$. A general labellings aggregation operator is a function $O_{AF} : 2^{\mathcal{L}abellings} - \{\emptyset\} \rightarrow \mathcal{L}abellings$ such that $O_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\}) = \mathcal{L}_{Coll}$.*

We are interested in aggregation operators that produce a collective outcome *compatible* with the individual opinions. The idea is to ensure that each member can publicly defend the common decision without having to directly go against his own position. Two notions of compatibility have been introduced in [3].

Definition 8 (Less or equally committed $\sqsubseteq$). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two labellings of argumentation framework $AF = (Ar, def)$. We say that $\mathcal{L}_1$ is less or equally committed as $\mathcal{L}_2$ (denoted by $\mathcal{L}_1 \sqsubseteq \mathcal{L}_2$) iff $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$.*

Definition 9 (Compatible labellings $\approx$). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two labellings of argumentation framework (Ar, def). We say that $\mathcal{L}_1$ is* compatible *with $\mathcal{L}_2$ (denoted as $\mathcal{L}_1 \approx \mathcal{L}_2$) iff $\texttt{in}(\mathcal{L}_1) \cap \texttt{out}(\mathcal{L}_2) = \emptyset$ and $\texttt{out}(\mathcal{L}_1) \cap \texttt{in}(\mathcal{L}_2) = \emptyset$.*

The intuition is that, in order to be compatible, two labellings cannot have $\texttt{in} - \texttt{out}$ conflicts. It can be noted that $\sqsubseteq$ is a partial order on labellings, whereas $\approx$ is not transitive [3]. It holds that if $\mathcal{L}_1 \sqsubseteq \mathcal{L}_2$, then $\mathcal{L}_1 \approx \mathcal{L}_2$.

We are now ready to state the *sceptical* and *credulous* aggregation operators.

Definition 10 (Sceptical initial aggregation operator sio_{AF}). *Let $\mathcal{L}abellings$ be the set of all possible labellings of argumentation framework $AF = (Ar, def)$. The* sceptical initial aggregation operator *is a function $sio_{AF} : 2^{\mathcal{L}abellings} - \{\emptyset\} \rightarrow \mathcal{L}abellings$ such that $sio_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\}) =$*
$\{(A, \texttt{in}) \mid \forall i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{in}\} \cup$
$\{(A, \texttt{out}) \mid \forall i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{out}\} \cup$
$\{(A, \texttt{undec}) \mid \exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) \neq \texttt{in} \land \exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) \neq \texttt{out}\}$.

The idea is that the group initially labels an argument `in` (respectively `out`) if all individual participants agree that the argument is `in` (respectively `out`). Otherwise it is `undec`. This procedure does not preserve admissibility. This means that it may return a labelling with illegally `in` or illegally `out` arguments. This is why, after the initial aggregation, a second iterative phase follows, where all the illegally `in` or `out` arguments are re-labelled to `undec`. Formally, this is defined as follows:

Definition 11 (Down-admissible labelling). *Let $\mathcal{L}$ be a labelling of argumentation framework $AF = (Ar, def)$. The* down-admissible *labelling of $\mathcal{L}$ is the biggest element of the set of all admissible labellings that is less or equally committed than $\mathcal{L}$.*

The down-admissible labelling is defined according to the partial order given by $\sqsubseteq$. It has been shown in [3] that such element always exists and is unique. We can now define the sceptical operator that ensures admissible outcomes.

Definition 12 (Sceptical aggregation operator so_{AF}). *Let $\mathcal{L}abellings$ be the set of all labellings of argumentation framework $AF = (Ar, def)$. The* sceptical aggregation operator *is a function $so_{AF} : 2^{\mathcal{L}abellings} - \{\emptyset\} \rightarrow \mathcal{L}abellings$ such that $so_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$ is the down-admissible labelling of $sio_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$.*

The aggregation operator above produces social outcomes that are less or equally committed to all the individual labellings. The following theorem, taken from [3], also ensures that this result is maximal.

Theorem 1. *Let $\mathcal{L}_1, \ldots, \mathcal{L}_n$ ($n \geq 1$) be labellings of argumentation framework $AF = (Ar, def)$. Let $\mathcal{L}_{so}$ be $so_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$. It holds that $\mathcal{L}_{so}$ is the biggest admissible labelling such that for every $i \in \{1, \ldots, n\}$: $\mathcal{L}_{so} \sqsubseteq \mathcal{L}_i$.*

The second aggregation operator that we consider here is the *credulous* one.

Definition 13 (Credulous initial aggregation operator cio_{AF}). *Let $\mathcal{L}abellings$ be the set of all possible labellings of argumentation framework $AF = (Ar, def)$. The* credulous initial aggregation operator *is a function $cio_{AF} : 2^{\mathcal{L}abellings} - \{\emptyset\} \rightarrow \mathcal{L}abellings$ such that $cio_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\}) =$*
$\{(A, \texttt{in}) \mid \exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{in} \wedge \neg\exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{out}\} \cup$
$\{(A, \texttt{out}) \mid \exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{out} \wedge \neg\exists i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{in}\} \cup$
$\{(A, \texttt{undec}) \mid \forall i \in \{1, \ldots, n\} : \mathcal{L}_i(A) = \texttt{undec} \vee (\exists i \in \{1, \ldots, n\} : \mathcal{L}_i = \texttt{in} \wedge \exists i \in \{1, \ldots, n\} : \mathcal{L}_i = \texttt{out})\}$.

The idea is that the group initially labels an argument A `in` (respectively `out`) if there is someone who believes A is `in` (respectively `out`) and nobody thinks A is `out` (respectively `in`). A is labelled `undec` in all other cases. The admissibility problem reappears here and is solved again by an iterative second phase where all illegally `in` and `out` arguments are relabelled `undec`.

Definition 14 (Credulous aggregation operator co_{AF}). *Let $\mathcal{A}dm\mathcal{L}abellings$ be the set of all admissible labellings of argumentation framework $AF = (Ar, def)$. The credulous aggregation operator is a function $co_{AF} : 2^{\mathcal{A}dm\mathcal{L}abellings} - \{\emptyset\} \rightarrow \mathcal{A}dm\mathcal{L}abellings$ such that $co_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$ is the down-admissible labelling of $cio_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$.*

It holds that the credulous outcome labelling ($\mathcal{L}_{co} = co_{AF}(\{\mathcal{L}_1, \ldots, \mathcal{L}_n\})$) is compatible with all the individual labellings, i.e. $\mathcal{L}_{co} \approx \mathcal{L}_i$ (for each $i \in \{1, \ldots, n\}$).

3 Preferences

In order to investigate Pareto optimality and strategy-proofness we need to assume that agents have preferences over the possible collective outcomes. For this purpose, we now define Hamming sets and Hamming distance (also called Dalal distance) among labellings, using similar approach as in [8].

Definition 15 (Hamming set $\ominus$). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two labellings of argumentation framework (Ar, def). We define the* Hamming set *between these labellings as $\mathcal{L}_1 \ominus \mathcal{L}_2 = \{A \mid \mathcal{L}_1(A) \neq \mathcal{L}_2(A)\}$.*

Definition 16 (Hamming distance $|\ominus|$). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two labellings of argumentation framework (Ar, def). We define the* Hamming distance *between these labellings as $\mathcal{L}_1 \, |\ominus| \, \mathcal{L}_2 = |\mathcal{L}_1 \ominus \mathcal{L}_2|$.*

In short, the Hamming set is the set of arguments on which two labellings differ, whereas the Hamming distance is the number of arguments on which two labellings differ. Since the labellings have only three values, we can use the following lemma.

Lemma 1. *Let (Ar, def) be an argumentation framework and $\mathcal{L}_1$ and $\mathcal{L}_2$ two labellings:*

a) $\mathcal{L}_1 \ominus \mathcal{L}_2 = \texttt{in}(\mathcal{L}_1) \cap \texttt{out}(\mathcal{L}_2) \cup \texttt{in}(\mathcal{L}_1) \cap \texttt{undec}(\mathcal{L}_2) \cup \texttt{out}(\mathcal{L}_1) \cap \texttt{in}(\mathcal{L}_2) \cup \texttt{out}(\mathcal{L}_1) \cap \texttt{undec}(\mathcal{L}_2) \cup \texttt{undec}(\mathcal{L}_1) \cap \texttt{in}(\mathcal{L}_2) \cup \texttt{undec}(\mathcal{L}_1) \cap \texttt{out}(\mathcal{L}_2)$

b) if $\mathcal{L}_1 \sqsubseteq \mathcal{L}_2$ then $\mathcal{L}_1 \ominus \mathcal{L}_2 = \texttt{undec}(\mathcal{L}_1) \cap \texttt{dec}(\mathcal{L}_2)$

c) if $\mathcal{L}_1 \approx \mathcal{L}_2$ then $\mathcal{L}_1 \ominus \mathcal{L}_2 = \texttt{dec}(\mathcal{L}_1) \cap \texttt{undec}(\mathcal{L}_2) \cup \texttt{undec}(\mathcal{L}_1) \cap \texttt{dec}(\mathcal{L}_2)$

Proof.

a) Follows from the fact that $\texttt{in}(\mathcal{L})$, $\texttt{out}(\mathcal{L})$ and $\texttt{undec}(\mathcal{L})$ partition the domain of any labelling $\mathcal{L}$.

b) and c) are obtained by eliminating the empty sets in a) and replacing $\texttt{in}(\mathcal{L}) \cup \texttt{out}(\mathcal{L})$ by $\texttt{dec}(\mathcal{L})$.

□

We are now ready to define an agent's preference given by the Hamming set and the Hamming distance as follows.

We write $\mathcal{L} \geq_i \mathcal{L}'$ to denote that agent *i prefers* labelling $\mathcal{L}$ to $\mathcal{L}'$. We write $\mathcal{L} \sim_i \mathcal{L}'$, and say that *i is indifferent* between $\mathcal{L}$ and $\mathcal{L}'$, iff $\mathcal{L} \geq_i \mathcal{L}'$ and $\mathcal{L}' \geq_i \mathcal{L}$. Finally, we write $\mathcal{L} >_i \mathcal{L}'$ (agent *i strictly prefers* $\mathcal{L}$ to $\mathcal{L}'$) iff $\mathcal{L} \geq_i \mathcal{L}'$ and not $\mathcal{L} \sim_i \mathcal{L}'$.

We assume that the labelling submitted by each agent is his most preferred one and, hence, the one he would like to see adopted by the whole group. The order over the other possible labellings is generated according to the distance from the most preferred one.

Definition 17 (Hamming set based preference $\geq_{i,\ominus}$). *Let (Ar, def) be an argumentation framework, $\mathcal{L}abellings$ the set of all its labellings and $\geq_i$ the preference of agent i. We say that agent i's preference is* Hamming set based *(written as $\geq_{i,\ominus}$) iff $\forall \mathcal{L}, \mathcal{L}' \in \mathcal{L}abellings, \mathcal{L} \geq_i \mathcal{L}' \Leftrightarrow \mathcal{L} \ominus \mathcal{L}_i \subseteq \mathcal{L}' \ominus \mathcal{L}_i$ where $\mathcal{L}_i$ is the agent's most preferred labelling.*

Definition 18 (Hamming distance based preference $\geq_{i,|\ominus|}$). *Let (Ar, def) be an argumentation framework, $\mathcal{L}abellings$ the set of all its labellings and $\geq_i$ the preference of agent i. We say that agent i's preference is* Hamming distance based *(written as $\geq_{i,|\ominus|}$) iff $\forall \mathcal{L}, \mathcal{L}' \in \mathcal{L}abellings, \mathcal{L} \geq_i \mathcal{L}' \Leftrightarrow \mathcal{L} \, |\ominus| \, \mathcal{L}_i \leq \mathcal{L}' \, |\ominus| \, \mathcal{L}_i$ where $\mathcal{L}_i$ is the agent's most preferred labelling.*

The Hamming set based preference yields an partial order, whereas the Hamming distance based preference yields a total preorder.

We now prove two lemmas establishing the relations between less or equally committed labellings and Hamming set/distance based preferences over labellings.

Lemma 2. *Let $\mathcal{L}$, $\mathcal{L}'$ and $\mathcal{L}_i$ be three labellings such that $\mathcal{L} \sqsubseteq \mathcal{L}' \sqsubseteq \mathcal{L}_i$. If $\mathcal{L}_i$ is the most preferred labelling of agent i and his preference is Hamming set or Hamming distance based, then $\mathcal{L}' \geq_{i,\ominus} \mathcal{L}$ and $\mathcal{L}' \geq_{i,|\ominus|} \mathcal{L}$ respectively.*

Proof. From $\mathcal{L} \sqsubseteq \mathcal{L}'$, we have that $\mathtt{dec}(\mathcal{L}) \subseteq \mathtt{dec}(\mathcal{L}')$, which is equivalent to $\mathtt{undec}(\mathcal{L}') \subseteq \mathtt{undec}(\mathcal{L})$ because $\mathtt{undec}$ is the complement of $\mathtt{dec}$. From this it follows that $\mathtt{undec}(\mathcal{L}') \cap \mathtt{dec}(\mathcal{L}_i) \subseteq \mathtt{undec}(\mathcal{L}) \cap \mathtt{dec}(\mathcal{L}_i)$. Since $\mathcal{L} \sqsubseteq \mathcal{L}_i$ and $\mathcal{L}' \sqsubseteq \mathcal{L}_i$ (by assumption and transitivity of $\sqsubseteq$), we can use Lemma 1b to obtain $\mathcal{L}' \ominus \mathcal{L}_i \subseteq \mathcal{L} \ominus \mathcal{L}_i$. By definition we have that $\mathcal{L}' \geq_{i,\ominus} \mathcal{L}$ and $\mathcal{L}' \geq_{i,|\ominus|} \mathcal{L}$.

□

Lemma 3. *Let $\mathcal{L}$, $\mathcal{L}'$ and $\mathcal{L}_i$ be three labellings and let $\mathcal{L} \sqsubseteq \mathcal{L}_i$. If $\mathcal{L}_i$ is the most preferred labelling of agent i, his preference is Hamming set based and $\mathcal{L}' \geq_{i,\ominus} \mathcal{L}$, then $\mathcal{L} \sqsubseteq \mathcal{L}'$.*

Proof. $\mathcal{L}' \geq_{i,\ominus} \mathcal{L}$ implies $\mathcal{L}' \ominus \mathcal{L}_i \subseteq \mathcal{L} \ominus \mathcal{L}_i$ which implies $\mathcal{L}(A) = \mathcal{L}_i(A) \Rightarrow \mathcal{L}'(A) = \mathcal{L}_i(A)$ for any argument A (i). $\mathcal{L} \sqsubseteq \mathcal{L}_i$ implies $\mathcal{L}(A) = \mathcal{L}_i(A)$ for any $A \in \mathtt{dec}(\mathcal{L})$ (ii). From (i) and (ii) it follows that $\mathcal{L}(A) = \mathcal{L}'(A)$ for any $A \in \mathtt{dec}(\mathcal{L})$. Hence $\mathcal{L} \sqsubseteq \mathcal{L}'$.

□

We now have the machinery to represent individual preferences over the collective outcomes. We can now turn to the first research question of the paper, i.e., whether the sceptical and credulous aggregation operators are Pareto optimal.

4 Pareto optimality

Pareto optimality is a fundamental social welfare principle that guarantees that it is not possible to improve a social outcome, i.e. it is not possible to make one individual better off without making at least one other person worse off. In order to address the question of whether the sceptical and the credulous aggregation operators are Pareto optimal, we first need to define when a labelling Pareto dominates another labelling.

Definition 19 (Pareto dominance). *Let $N = \{1, \ldots, n\}$ be a set of individuals with preferences $\geq_i, i \in N$. Labelling $\mathcal{L}$* Pareto dominates *$\mathcal{L}'$ if $\forall i \in N$, $\mathcal{L} \geq_i \mathcal{L}'$ and $\exists j \in N, \mathcal{L} >_j \mathcal{L}'$.*

A labelling is Pareto optimal if it is not dominated by any other labelling.

Definition 20 (Pareto optimality). *Labelling $\mathcal{L}$ is* Pareto optimal *if there is no $\mathcal{L}' \neq \mathcal{L}$ such that $\forall i \in N$, $\mathcal{L}' \geq_i \mathcal{L}$ and $\exists j \in N, \mathcal{L}' >_j \mathcal{L}$.*

We say that an aggregation operator is Pareto optimal if all its outcomes are Pareto optimal. In particular, candidates for dominance are admissible and less or equally committed labellings in the case of the sceptical operator, and compatible labellings in the case of the credulous operator.

Theorem 2. *If individual preferences are Hamming set based or Hamming distance based, then the sceptical aggregation operator is Pareto optimal when choosing from the admissible labellings that are smaller or equal (w.r.t $\sqsubseteq$) to each of the participants' individual labellings.*

Proof. Let P be a profile of admissible labellings, $\mathcal{L}_{SO} = so_{AF}(P)$ and $\mathcal{L}_X$ some admissible labelling with the property $\forall i \in N, \mathcal{L}_X \sqsubseteq \mathcal{L}_i$. From Theorem 1 we know that $\mathcal{L}_{SO}$ is the biggest admissible labelling with such property, so $\mathcal{L}_X \sqsubseteq \mathcal{L}_{SO}$. So we have $\forall i \in N, \mathcal{L}_X \sqsubseteq \mathcal{L}_{SO} \sqsubseteq \mathcal{L}_i$. From Lemma 2 we have $\mathcal{L}_{SO} \geq_{i,\ominus} \mathcal{L}_X$ and $\mathcal{L}_{SO} \geq_{i,|\ominus|} \mathcal{L}_X$ for any i. So no agent strictly prefers $\mathcal{L}_X$ and hence there is no labelling that dominates $\mathcal{L}_{SO}$.

□

Theorem 3. *If individual preferences are Hamming set based, then the credulous aggregation operator is Pareto optimal when choosing from the admissible labellings that are compatible ($\approx$) to each of the participants' individual labellings.*

Proof. Let P be a profile of admissible labellings, $\mathcal{L}_{CO} = co_{AF}(P)$, $\mathcal{L}_{CIO} = cio_{AF}(P)$. Assume by contradiction that there exists some admissible labelling $\mathcal{L}_X$ with the property $\forall i \in N, \mathcal{L}_X \approx \mathcal{L}_i$ that dominates $\mathcal{L}_{CO}$.

First notice that compatibility ensures that there are no `in`/`out` conflicts between $\mathcal{L}_X$ and $\mathcal{L}_{CO}$. If there is a conflict between agents' labellings on some argument, then both $\mathcal{L}_X$ and $\mathcal{L}_{CO}$ need to label it `undec`. If there exists an agent whose labelling decides on some argument and other agents' labellings agree or retain from decision, $\mathcal{L}_{CO}$ and $\mathcal{L}_X$ also agree or retain from decision. If all agents retain from decision on some argument, $\mathcal{L}_{CO}$ by definition also retains, and $\mathcal{L}_X$ may label freely.

Let us take $A \in \texttt{dec}(\mathcal{L}_X)$. Then, there needs to be an agent with a labelling that agrees on A. Otherwise all agents' labellings would be undecided on such argument and, according to definition, $\mathcal{L}_{CO}$ would not decide either. But then all agents' labellings will agree on such argument with $\mathcal{L}_{CO}$ and disagree with $\mathcal{L}_X$, so no agent will strongly prefer $\mathcal{L}_X$, which contradicts with domination. So there exists at least one agent whose labelling agrees with $\mathcal{L}_X$ on A. Other agents' labellings also need to agree on A or label it `undec` because of the compatibility of $\mathcal{L}_X$. Then by definition $\mathcal{L}_{CIO}(A) = \mathcal{L}_X(A)$. This holds for any argument $A \in dec(\mathcal{L}_X)$, so we have $\mathcal{L}_X \sqsubseteq \mathcal{L}_{CIO}$. But $\mathcal{L}_X$ is admissible and, by Theorem 1, $\mathcal{L}_{CO}$ is the biggest admissible labelling less or equally committed as $\mathcal{L}_{CIO}$. So we have $\mathcal{L}_X \sqsubseteq \mathcal{L}_{CO} \sqsubseteq \mathcal{L}_{CIO}$.

$\mathcal{L}_X$ must be different from $\mathcal{L}_{CO}$ to dominate it. Let A be an argument on which these labellings differ. From the previous it follows that $A \in \texttt{undec}(\mathcal{L}_X)$ and $A \in \texttt{dec}(\mathcal{L}_{CO})$. $\mathcal{L}_{CO}$ decides on an argument only if there exists an agent that decides on such argument. But then this argument will agree on A with $\mathcal{L}_{CO}$ and disagree with $\mathcal{L}_X$, so it will not prefer $\mathcal{L}_X$. This is in contradiction with dominance. Hence, such dominating labelling cannot exist.

□

Observation 1 *The credulous aggregation operator is not Pareto optimal when the preferences are Hamming distance based. An example is given in Fig. 3. Both labellings $\mathcal{L}_{CO}$ and $\mathcal{L}_X$ are compatible with both $\mathcal{L}_1$ and $\mathcal{L}_2$, but $\mathcal{L}_X$ is closer when applying Hamming distance. $\mathcal{L}_1 \ominus \mathcal{L}_{CO} = \mathcal{L}_2 \ominus \mathcal{L}_{CO} = \{A, B, E, F, G\}$, so Hamming distance is 5, whereas $\mathcal{L}_1 \ominus \mathcal{L}_X = \mathcal{L}_2 \ominus \mathcal{L}_X = \{A, B, C, D\}$, so Hamming distance is 4.*

We summarise our results in Table 1 below.

We are now ready to address the second research question of the paper, that is, whether the credulous and sceptical aggregation operators are manipulable.

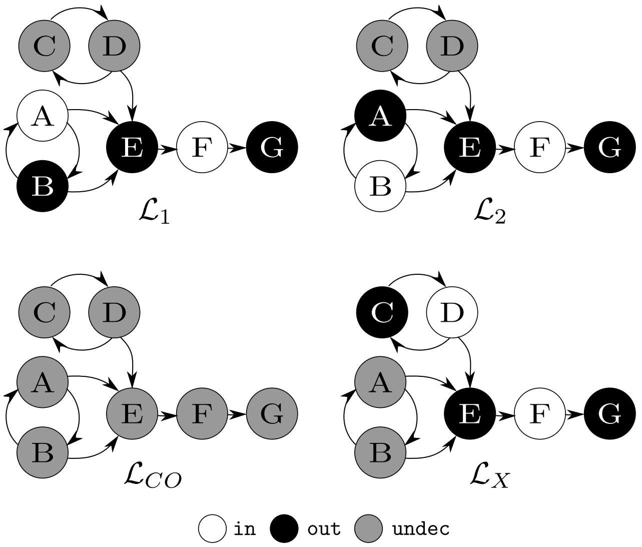


Fig. 3. The credulous aggregation operator is not Pareto optimal under Hamming distance based preferences.

	Sceptical Operator	Credulous Operator
Hamming set	Yes (Theorem 2)	Yes (Theorem 3)
Hamming distance	Yes (Theorem 2)	No (Observation 1)

Table 1. Pareto optimality of the aggregation operators depending on the type of preference.

5 Strategic manipulation

When an agent knows the positions of the other agents, he may discover that he has an incentive to submit an insincere position. If an aggregation rule is manipulable, an agent may obtain a social outcome that is closer to his actual preferences by submitting an insincere input. Hence, an important question to address when dealing with aggregation procedures is to study whether they are strategy-proof (i.e. non-manipulable).

In order to talk about manipulability, we first need to denote a profile in which a labelling has been changed. We recall that by *profile* we refer to a set of individual labellings $\{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$. Profile $P_{\mathcal{L}_k/\mathcal{L}'_k}$ is profile P where agent k's labelling $\mathcal{L}_k$ has been changed to $\mathcal{L}'_k$.

Definition 21 (Strategic lie). *Let P be a profile and $\mathcal{L}_k \in P$ the most preferred labelling of an agent with preference $\geq_k$. Let O be any aggregation operator. A labelling $\mathcal{L}'_k$ such that $O(P_{\mathcal{L}_k/\mathcal{L}'_k}) >_i O(P)$ is called a* strategic lie .

Definition 22 (Strategy-proof operator). *We say that an aggregation operator O is* strategy-proof *if strategic lies are not possible.*

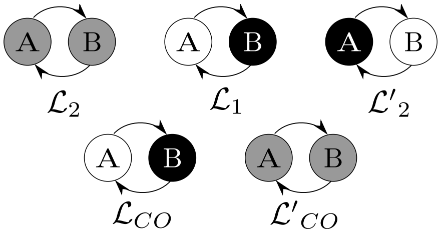


Fig. 4. The credulous aggregation operator is not strategy-proof.

Observation 2 *The credulous aggregation operator is not strategy-proof. See the example in Fig. 4. The agent with labelling $\mathcal{L}_2$ can insincerely report $\mathcal{L}'_2$ to obtain his preferred labelling. This makes agent with labelling $\mathcal{L}_1$ worse off. The example is valid for both Hamming set and Hamming distance based preferences.*

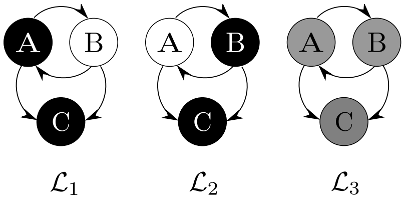


Fig. 5. The sceptical aggregation operator is not strategy-proof.

Observation 3 *The sceptical aggregation operator is not strategy-proof. Consider the three labellings in Fig. 5. Labelling $\mathcal{L}_1$ of agent 1 when aggregated with $\mathcal{L}_2$ gives labelling $\mathcal{L}_3$, which differs on all three arguments* [5]. *But, when the agent strategically lies*

[5] To see why this is reasonable outcome, consider the following example, taken from [22].

A: John says the suspect stabbed the victim.
B: Bob says the suspect shot the victim.
C: The suspect is innocent.

This essentially yields an argumentation framework like in Fig. 5 where A and B attack each other as well as C. John may subscribe to labelling $\mathcal{L}_2$, Bob to labelling $\mathcal{L}_1$. However, given *and reports labelling $\mathcal{L}_2$ instead, the result of the aggregation is the same labelling $\mathcal{L}_2$, which differs only on two arguments $\{A, B\}$. The example is valid for both Hamming set and Hamming distance based preferences.*

Surprisingly, however, this lie does not harm the other agent. Quite the opposite, it improves the social outcome for both the agents. In order to study this kind of situation, we now introduce the distinction between malicious and benevolent lies.

Definition 23 (Malicious lie). *Let O be some aggregation operator and P a profile. We say that a strategic lie $\mathcal{L}'_k$* is malicious *iff, for some agent $j \neq k$, $O(P) >_j O(P_{\mathcal{L}_k/\mathcal{L}'_k})$.*

Definition 24 (Benevolent lie). *Let O be some aggregation operator and P a profile. We say that a strategic lie $\mathcal{L}'_k$* is benevolent *iff, for any agent i $O(P_{\mathcal{L}_k/\mathcal{L}'_k}) \geq_i O(P)$ and there exists an agent $j \neq k$, $O(P_{\mathcal{L}_k/\mathcal{L}'_k}) >_j O(P)$.*

Theorem 4. *Consider the sceptical aggregation operator and Hamming set based preferences. For any agent, his strategic lies are benevolent.*

Proof. Let P be a profile, and $\mathcal{L}'_k$ a strategic lie of agent k. Denote $\mathcal{L}_{SO} = so_{AF}(P)$ and $\mathcal{L}'_{SO} = so_{AF}(P_{\mathcal{L}_k/\mathcal{L}'_k})$. Agent k's preference is $\mathcal{L}'_{SO} >_k \mathcal{L}_{SO}$ (i). We will show that for any agent $i \neq k$, we have $\mathcal{L}'_{SO} >_i \mathcal{L}_{SO}$.

Since the sceptical aggregation operator produces social outcomes that are less or equally committed to all the individual labellings, we have that $\mathcal{L}'_{SO} \sqsubseteq \mathcal{L}_i$ for all $i \neq k$ (ii). Similarly, we have $\mathcal{L}_{SO} \sqsubseteq \mathcal{L}_k$ (iii). From (i) and (iii), by Lemma 3, we have that $\mathcal{L}_{SO} \sqsubseteq \mathcal{L}'_{SO}$ (iv). From (iv) and (ii) we have $\mathcal{L}_{SO} \sqsubseteq \mathcal{L}'_{SO} \sqsubseteq \mathcal{L}_i$ for all $i \neq k$. Finally, we can apply Lemma 2 to obtain $\mathcal{L}'_{SO} \geq_i \mathcal{L}_{SO}$ for all $i \neq k$ (v). We showed that lie is not malicious, now we show that it is benevolent.

(iii) implies $\mathtt{undec}(\mathcal{L}_k) \subseteq \mathtt{undec}(\mathcal{L}_{SO})$ (vi). (i) and (vi) implies $\exists A \in \mathtt{dec}(\mathcal{L}_k) : A \in \mathtt{undec}(\mathcal{L}_{SO}) \wedge A \in \mathtt{dec}(\mathcal{L}'_{SO})$ (vii). From (vii), (ii) and (v) $\mathcal{L}'_{SO} >_i \mathcal{L}_{SO}$ for $i \neq k$. □

Theorem 5. *Consider the sceptical aggregation operator and Hamming distance based preferences. For any agent, his strategic lies are benevolent.*

Proof. Let P be a profile, and $\mathcal{L}'_k$ a strategic lie of agent k whose most preferred labelling is $\mathcal{L}_k$. Denote $\mathcal{L}_{SO} = so_{AF}(P)$ and $\mathcal{L}'_{SO} = so_{AF}(P_{\mathcal{L}_k/\mathcal{L}'_k})$. We will show that, if $\mathcal{L}'_{SO}$ is strictly preferred to $\mathcal{L}_{SO}$ by agent k, then it is also strictly preferred by any other agent. Without loss of generality we can take agent j, $j \neq k$,whose most preferred labelling is $\mathcal{L}_j$.

Let us partition the arguments into the following disjoint groups: $A = \mathtt{dec}(\mathcal{L}_{SO}) \setminus \mathtt{dec}(\mathcal{L}'_{SO})$ (decided arguments that became undecided), $B = \mathtt{dec}(\mathcal{L}'_{SO}) \setminus \mathtt{dec}(\mathcal{L}_{SO})$ (undecided arguments that became decided), $C = \mathtt{dec}(\mathcal{L}'_{SO}) \cap \mathtt{dec}(\mathcal{L}_{SO})$ (arguments decided in both labellings), $D = \mathtt{undec}(\mathcal{L}'_{SO}) \cap \mathtt{undec}(\mathcal{L}_{SO})$ (arguments undecided in both labellings).

Labellings $\mathcal{L}_{SO}$ and $\mathcal{L}'_{SO}$ agree on the arguments in D (which are labeled $\mathtt{undec}$) and C, whose arguments are labeled $\mathtt{in}$ or $\mathtt{out}$. On the arguments in C there are no $\mathtt{in}-\mathtt{out}$

the conflicting testimonies, a judge may decide that there is simply insufficient evidence that is beyond reasonable doubt to reject the presumption of innocence.

conflicts between $\mathcal{L}_{SO}$ and $\mathcal{L}'_{SO}$ as the sceptical aggregation operator guarantees social outcomes less or equally committed than $\mathcal{L}_j$. Therefore, only arguments from A and B have an impact on Hamming distance.

Both labellings $\mathcal{L}_k$ and $\mathcal{L}_j$ agree with $\mathcal{L}_{SO}$ on the arguments in A because $\mathcal{L}_{SO}$ decides on those arguments and is less or equally committed than both labellings. On the other side, $\mathcal{L}'_{SO}$ remains undecided on the arguments in A so both labellings $\mathcal{L}_k$ and $\mathcal{L}_j$ disagrees with $\mathcal{L}'_{SO}$ on A.

$\mathcal{L}'_{SO}$ is less or equally committed than $\mathcal{L}_j$ so, as above, we obtain that on the arguments in B, $\mathcal{L}_j$ agrees with $\mathcal{L}'_{SO}$ and disagrees with $\mathcal{L}_{SO}$. On the contrary, $\mathcal{L}'_{SO}$ does not have to be less or equally committed than $\mathcal{L}_k$ and so, for agent k, some of the arguments from B increase the distance and some of them decrease. If agent k prefers $\mathcal{L}'_{SO}$ to $\mathcal{L}_{SO}$, then the number of the arguments decreasing the distance must be greater than the number of those increasing by more than $|A|$. But for agent j all the arguments from B are decreasing the distance, as $\mathcal{L}_j$ agrees with $\mathcal{L}'_{SO}$ on the whole B. So, if agent k gains by switching to labelling $\mathcal{L}'_{SO}$, agent j needs to gain at least the same.

□

Note that this is not in contradiction with the result that the sceptical aggregation operator is Pareto optimal (Theorem 2). There, the outcome needed to be less or equally committed than all the opinions in the profile. Here, the outcome is less or equally committed than the insincere submitted labelling $\mathcal{L}'$, rather than the most preferred labelling $\mathcal{L}_k$. If we restrict the lies to the ones that aggregated are less or equally committed, then the sceptical aggregation operator is strategy-proof.

We summarise our results in Table 2.

	Sceptical Operator	Credulous Operator
Hamming set	No (Observation 3) but it is benevolent (Theorem 4)	No and not benevolent (Observation 2)
Hamming distance	No (Observation 3) but it is benevolent (Theorem 5)	No and not benevolent (Observation 2)

Table 2. Strategy-proofness of operators depending on the type of preference.

6 Related Work

The study of aggregation problems in abstract aggregation is recent. The aggregation of individual defeat relations into a social one has been investigated by Coste-Marquis *et al.* [7] and Tohmé *et al.* [26]. In [7], an approach to merge Dung's argumentation frameworks is presented. Unlike our approach, the argumentation frameworks to be merged may be different, that is agents may ignore arguments put forward by other agents. Conflicts between argumentation frameworks are solved using merging techniques [14], in particular a distance-based merging operator. The intuition is to minimize the distance between the profile and the collective outcome. Typically, more than one argumentation system minimizes the chosen distance. Hence, the final step consists in asking the individuals to vote on the selected extensions to obtain the final group argumentation framework. Their approach is shown to preserve at the collective level all the evaluations on which the individuals do not disagree.

The main conceptual difference between the approach of Coste-Marquis *et al.* [7] and the approach of Caminada and Pigozzi [3] on which the current paper is based can be explained as follows. When one interprets an argumentation framework as the often conflicting information that is available in a particular case, one can distinguish between situations where the participants all have the same information available and situations when they do not. In the latter situations, one has to merge the available information. How to do this is a question studied in the work of Coste-Marquis *et al.* [7]. However, in the former situations, where all participants do share the same information, it is still possible to have different opinions about how to interpret this information, because it is perfectly possible for several reasonable interpretations to exist. An example of this would be a jury in a legal trial, whose members are all presented precisely the same facts but can still disagree on how to interpret these facts when it comes to reaching a verdict. It is these kinds of situations, where the participants share the same information (argumentation framework) but still disagree on how to interpret it (resulting in different labellings) that are studied in [3] as well as in the current paper.

In Tohmé *et al.* [26], the aggregation of individual attack relations is linked to the aggregation of individual preferences in social-choice. They show that, by assuming argumentation frameworks in which the attack relations are acyclic, it is possible to define an aggregation operator that satisfies Arrow's theorem conditions.

The work of Rahwan and Tohmé [24] is closer to the approach of [3] and of the present paper. Given an argumentation framework, Rahwan and Tohmé address the question of how to aggregate individual labellings into a collective position. By drawing on a general impossibility theorem from judgment aggregation, they prove an impossibility result and provide some escape solutions. Moreover, they investigate the manipulability issue of the plurality aggregation rule.

Another work by Rahwan and Larson [23] explores welfare properties of collective argument evaluation. They consider agents with a preference relation between the various complete labellings. An example of such a preference relation would be to try to maximize the `in`-labelling of a set of arguments an individual agent cares about. In particular, they show that different types of preference orderings result in different types of labellings becoming Pareto optimal.

In this paper, the analysis of manipulability conducted to the formulation of a benevolent type of manipulation. A similar idea was introduced and studied in [12], where manipulation was seen as a coordinated action of the whole group. There, the authors consider individuals that are willing to deliberately fallback into declaring a less preferred input to ensure that a collective decision can be made. Hence, they analyze how the individuals can achieve two goals simultaneously: to ensure a rational group decision and, at the same time, to do so by diverging the slightest possible from their own sincere judgments.

7 Conclusion

Caminada and Pigozzi defined the problem of aggregating individual labellings in a judgment aggregation setting [3]. In order to escape the impossibility results that plague judgment aggregation, and that Rahwan and Tohmé proved to hold also for abstract argumentation [24], they relaxed the independence condition. However, in preference and judgment aggregation settings, one reason to impose the independence condition is that it ensures non-manipulability. Thus, aim of this paper was to examine the consequences of dropping independence for two operators of [3]. To this end, we have analyzed the sceptical and credulous aggregation operators introduced in [3] from a social welfare perspective. First, we have addressed the question of whether the two aggregation operators are Pareto optimal, i.e. whether they select an outcome that cannot be improved for one agent without damaging the other individuals. Second, we have investigated whether the two aggregation operators can be manipulated. We have found that, whereas the credulous aggregation operator is prone to manipulation and, by lying, an agent harms the other members, the sceptical aggregation operator also offers incentives to lie but somewhat surprisingly those lies turn out to promote social welfare. Our findings show that relaxing the independence condition does not lead to harmful consequences.

In future work, we plan to consider focal set oriented agents, that is, agents who care only about a subset of the argumentation framework. An agent may not require that his position on *all* the issues at stake becomes the group outcome. In this case, an agent will be prone at strategizing in order to have his labelling on a subset of the arguments only assumed by the whole group. We plan to explore the strategic behavior of the sceptical and credulous outcomes for focal set oriented agents.

References

1. K. Arrow. *Social choice and individual values*. Cowles Foundation Monograph Series, 1963.
2. S Brams, M Kilgour, and R Sanver. A minimax procedure for electing committees. *Public Choice*, 132(3-4):401–420, 2007.
3. M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22:64–102, 2011. 10.1007/s10458-009-9116-7.
4. M. Caminada, G. Pigozzi, and M. Podlaszewski. Manipulation in group argument evaluation (extended abstract). In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, 2011. In print.
5. M.W.A. Caminada. On the issue of reinstatement in argumentation. In M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, pages 111–123. Springer, 2006. LNAI 4160.
6. M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.
7. S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. On the merging of dung's argumentation systems. *Artificial Intelligence*, 171(10-15):730–753, 2007.
8. M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 475–479, 1988.

9. F. Dietrich and C. List. Arrow's theorem in judgment aggregation. *Social Choice and Welfare*, 29(1):19–33, 2007.
10. F Dietrich and C List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23:269–300, 2007.
11. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence*, 77:321–357, 1995.
12. D. Grossi, G. Pigozzi, and M. Slavkovik. White manipulation in judgment aggregation. Proceedings of BNAIC 2009 - The 21st Benelux Conference on Artificial Intelligence, 2009.
13. C. Klamler. Borda and condorcet: Some distance results. *Theory and Decision*, 59:97–109, 2005.
14. S. Konieczny and R. Pino Perez. On the logic of merging. In *Proceedings of KR'98*, pages 488–498, 1998.
15. C. List and P. Pettit. Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18:89–110, 2002.
16. C. List and P. Pettit. Aggregating sets of judgments: Two impossibility results compared. *Synthese*, 140(1-2):207–235, 2004.
17. C. List and B. Polak. Introduction to judgment aggregation. *Journal of Economic Theory*, 145(2):441–466, 2010.
18. H. Nurmi. A comparison of some distance-based choice rules in ranking environments. *Theory and Decision*, 57:5–24, 2004.
19. M. Pauly and M. van Hees. Logical constraints on judgment aggregation. *Journal of Philosophical Logic*, 35:569–585, 2006.
20. G. Pigozzi. Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2):285–298, 2006.
21. J. L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.
22. H. Prakken. Intuitions and the modelling of defeasible reasoning: some case studies. In *Proceedings of the Ninth International Workshop on Nonmonotonic Reasoning (NMR-2002)*, pages 91–99, Toulouse, France, 2002.
23. I. Rahwan and K. Larson. Welfare properties of argumentation-based semantics. In *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC)*, 2008.
24. I. Rahwan and F. Tohmé. Collective argument evaluation as judgment aggregation. In van der Hoek, Kaminka, Lespérance, Luck, and Sen, editors, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.
25. A. K. Sen. *Collective Choice and Social Welfare*. Holden Day, 1970.
26. F.A. Tohmé, G. A. Bodanza, and G. R. Simari. Aggregation of attack relations: A social-choice theoretical analysis of defeasibility criteria. In S. Hartmann and G. Kern-Isberner, editors, *Proceedings of Foundations of Information and Knowledge Systems, FoIKS 2008*, pages 8–23, Pisa, Italy, 2008.
27. B. Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J.-J.Ch. Meyer and L.C. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, Utrecht, 1996. Utrecht University.
28. B. Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India*, pages 623–628, 2007.

Fuzzy Labeling for Argumentation Frameworks

Cristian Gratie[1] and Adina Florea[1]

University "Politehnica" of Bucharest, Romania,
cristian.gratie@cs.pub.ro, adina@cs.pub.ro

Abstract. This paper introduces the use of fuzzy labels in argumentation. The first approach we propose is built as a natural extension of the in, out, undec labeling to real valued labels, coupled with an unsupervised learning algorithm that assigns consistent labels starting from a random initial assignment. The second approach regards argument (fuzzy) labels as degrees of certitude in the argument's acceptability. This translates into a system of equations that provides among its solutions the labelings that describe complete extensions.

Keywords: argumentation framework, fuzzy labeling

1 Introduction

Since the initial work of Dung [9], argumentation frameworks have steadily gained popularity and a lot of work has been done for extending various parts of his proposal.

Argument acceptability is most often dealt with by means of extension-based semantics. The initial semantics proposed by Dung [9] (complete, grounded, preferred, stable) were followed by several others: semi-stable [4], stage [11], ideal [8], eager [5], prudent [7], CF2 [2], resolution-based grounded [1], enhanced preferred [12].

Another approach to argument acceptability is that of argument labeling, proposed by Caminada [3, 6], where each argument is assigned one of three labels: in, out, undec. The labels are assigned so as to obey some restrictions that define complete labelings, which are shown to be in a one-to-one correspondence with complete extensions. Additional set-inclusion related constraints (maximality or minimality) are used for identifying the labelings that correspond to semantics that prescribe extensions that are complete (grounded, preferred, stable, semi-stable).

We feel that this approach can be extended in such a way as to allow for other semantics to be described in terms of labelings as well. The approach we propose does not do that itself, but provides links with domains that were not, to the best of our knowledge, linked with the labeling approach before, namely converging recursive sequences, systems of equations and SCC-recursiveness.

What we propose are real-valued labels in the interval $[0, 1]$. We call these values fuzzy labels. Two approaches for defining the completeness rules for this kind of labeling are defined and analyzed with relevant examples.

Next section will provide some more details about related work, expecially that of Caminada. Section 3 will introduce the fuzzy labels and derive the criteria for complete labelings. We continue in section 4 with an unsupervised learning approach for assigning fuzzy labels. Section 5 introduces a set of different criteria for complete labelings and relates this approach to solving systems of equations. The paper ends with conclusions and remarks about future work.

2 Related Work

As we stated already in the introduction, our work is based on that of Caminada [3, 6] and constitutes an attempt to generalize it. In his work, Caminada proposes the assignment of one of three labels (`in, out, undec`) to each argument. This assignment can be linked with the justification state of the arguments, in the sense that the `in` label corresponds to arguments that are accepted, the `out` label corresponds to arguments that are defeated and the `undec` label corresponds to arguments for which no decision is taken.

In order to ensure that labels do indeed correspond to the justification state of the arguments (with respect to a given semantics) some constraints are enforced on the labelings. A labeling is considered complete if the following criteria are met:

- if an argument is labeled `in`, its attackers are labeled `out`
- if an argument is labeled `out`, at least one of its attackers is labeled `in`
- if all attackers of an argument are labeled `out`, then the argument is labeled `in`
- if an attacker of an argument is labeled `in`, then the argument is labeled `out`

Caminada shows that complete labelings are in a one-to-one correspondence with the complete extensions of the argumentation framework, in the sense that the arguments labeled `in` by a complete labeling form a complete extension of the framework and for each complete extension there is such a complete labeling. Relaxing the constraints yields admissible or conflict-free labelings.

Furthermore, Caminada shows that the preferred extensions correspond to the complete labeling with the maximal set (with respect to set inclusion) of `in` arguments and with the maximal `out`, stable corresponds to empty `undec`, grounded corresponds to minimal `in`, minimal `out` and maximal `undec`, and semi-stable corresponds to minimal `undec`.

Our approach extends the domain of the labels to the real interval $[0, 1]$ and translates the completeness rules for numeric values. We will define a mapping of fuzzy labels to `in, out, undec` so as to benefit from all the results that hold for Caminada's approach, but we will also point out how fuzzy labels can be used for richer information on the justification state of the arguments.

Another important addition of our approach is the fact that we provide an algorithm for assigning labels based on an initial random assignment. The algorithm exhibits fast convergence experimentally and can also be parallelized for better performance.

Our approach may also be linked with fuzzy argumentation frameworks, defined by Janssen et al. [10] in the sense that the fuzzy labels may be interpreted as fuzzy membership to an extension. However, Janssen's approach differs significantly from ours by the fact that the attack relation that defines the framework is taken to be fuzzy and the conflict-free and admissibility definitions are changed accordingly.

We will also link a part of our approach with SCC-recursiveness used by Baroni et al. [2]. We believe that this link might enable a correlation between the CF2 semantics defined in the same paper and argument labeling.

3 Introducing Fuzzy Labels

We begin this section by recalling the definition of argumentation frameworks, then provide the intuition behind our proposal, namely the link with the `in`, `out` and `undec` labels of Caminada, by proposing an alternate definition of the criteria for complete labelings.

Definition 1. *An argumentation framework is a pair $(\mathcal{A}, \mathcal{R})$, where $\mathcal{A}$ is a set of arguments and $\mathcal{R}$ is a relation defined on $\mathcal{A}$, i.e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. An argument a is said to attack an argument b (written $a \to b$) iff $(a, b) \in \mathcal{R}$.*

Note that we have recalled the definition of argumentation frameworks mostly for notational purposes. We will not define the semantics here as well, since they may be looked up in referred papers.

Definition 2 (Complete labeling [6]). *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. A complete labeling of F is a labeling such that for each argument $a \in \mathcal{A}$ it holds that:*

1. *if a is labeled* `in`*, then all attackers of a are labeled* `out`
2. *if all attackers of a are labeled* `out`*, then a is labeled* `in`
3. *if a is labeled* `out`*, then a has an attacker that is labeled* `in`
4. *if a has an attacker that is labeled* `in`*, then a is labeled* `out`

Definition 3. *We define the following (intuitive) order relations on the set of labels:* `out` < `undec` < `in`*. We also define the opposite of each label: opp*(`in`) = `out`*, opp*(`out`) = `in` *and opp*(`undec`) = `undec`*. We will use $\lambda(a)$ to denote the label of the argument a.*

Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. A complete labeling of F is a labeling such that for all arguments $a \in \mathcal{A}$ it holds that

1. $\lambda(a) = opp(\max_{b \in \mathcal{A}, b \to a} \lambda(b))$*, if a has at least one atttacker*
2. $\lambda(a) = \mathtt{in}$*, if a is not attacked*

Alternatively, we can keep just relation 1 of definition 3 and assume that the maximum is defined to take the minimum possible value (`out`) when the set of attackers is empty.

Proposition 1. *Definitions 2 and 3 are equivalent.*

Proof. It is easy to see that all attackers of a are `out` iff $\max_{b \in \mathcal{A}, b \to a} \lambda(b) = \texttt{out}$ and a has an attacker labeled `in` iff $\max_{b \in \mathcal{A}, b \to a} \lambda(b) = \texttt{in}$.

We are now ready to introduce complete fuzzy labelings.

Definition 4 (Complete fuzzy labeling). *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. A complete fuzzy labeling of F is a mapping $\lambda : \mathcal{A} \to [0, 1]$ such that:*

1. *$\lambda(a) = 1 - \max_{b \in \mathcal{A}, b \to a} \lambda(b)$, if a has at least one atttacker*
2. *$\lambda(a) = 1$, if a is not attacked*

Again, we can simplify things if we consider that the maximum is defined as 0 for the empty set.

Definition 5 (Fuzzy labeling conversion). *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and λ a complete fuzzy labeling of F. For $\alpha \in [0, 0.5)$ we define the α-conversion of the fuzzy labeling as the labeling $\lambda^{(\alpha)}$ with:*

$$\lambda^{(\alpha)}(a) = \begin{cases} \texttt{out} & , \textit{if}\, \lambda(a) \in [0, \alpha] \\ \texttt{undec} & , \textit{if}\, \lambda(a) \in (\alpha, 1 - \alpha) \\ \texttt{in} & , \textit{if}\, \lambda(a) \in [1 - \alpha, 1] \end{cases}$$

Proposition 2. *The α-conversion of a complete fuzzy labeling is a complete labeling.*

Proof. Check that the converted labeling satisfies definition 3.

Consider the following simple framework

$$a \leftrightarrow b \to c$$

and the following complete fuzzy labelings:

$$\begin{array}{lll} \lambda_1(a) = 0.3, & \lambda_1(b) = 0.7, & \lambda_1(c) = 0.3 \\ \lambda_1(a) = 0.6, & \lambda_1(b) = 0.4, & \lambda_1(c) = 0.6 \end{array}$$

If we compute $\lambda_1^{(0.2)}$, all arguments will be labeled `undec`, which corresponds to the grounded extension $\varnothing$. The same is true for $\lambda_2^{(0.2)}$. On the other hand, $\lambda_1^{(0.45)}$ will label a and c `out` and b `in`, which corresponds to the complete extension $\{b\}$, while for $\lambda_2^{(0.45)}$ we get the complete extension $\{a, c\}$. From this we see that the actual complete labeling that we get upon conversion depends on the value we choose for α.

It is easy to see that once we apply conversion we get the usual complete labeling that maps to complete extensions. If we keep the fuzzy labels, however, we also get a preference between arguments and also a restriction on preferences (in this case a and c must have the same value). This, coupled with the algorithm in the next section, provides richer information to a rational agent deciding on the acceptability of the arguments.

Proposition 3. *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and λ a complete fuzzy labeling of F. Then $\lambda(a) = 1$ for all arguments a in the grounded extension of F and $\lambda(a) = 0$ for all arguments a attacked by the grounded extension.*

Proof. Consider that we assign labels one by one. First, we can safely assign 1 to all arguments that are not attacked. The arguments attacked by these must then be labeled 0, as the maximum label among their attackers is bound to be 1. Then look at all arguments that only have attackers which are labeled 0 and label them 1. And so on, practically doing the construction of the grounded extension.

Alternatively, consider the fact that any α-conversion of a complete fuzzy labeling is a complete labeling that describes a complete extension. Since this is true for any $\alpha \in [0, 0.5)$, it must also hold for $\alpha = 0$, where only arguments labeled 1 become `in` and only arguments labeled 0 become `out`. The result follows from the fact that the grounded extension is included in all complete extensions.

4 Unsupervised Learning of Fuzzy Labels

In the previous section we have provided two simple examples of complete fuzzy labelings where we have chosen the values to fit the restrictions. We will discuss the actual choice of the labels in this section and provide some experimental results for the approach.

We consider that the rational agent trying to label the arguments starts with an initial labeling, which may be a random assignment or may be based on other criteria that depend on the agent. We denote the initial label of argument a with $\lambda_a^{(0)}$.

At each step k, the label of each argument a is updated according to the following rule.

$$\lambda_a^{(k+1)} = (1-\alpha)\lambda_a^{(k)} + \alpha(1 - max_{b \to a}\lambda_b^{(k)})$$

.

We implicitly assume that the maximum value is taken to be 0 whenever there is no attacker. Running this algorithm we observed that the values converge rapidly on all tests. We take the result labeling as:

$$\lambda(a) = \lim_{k \to \infty} \lambda_a^{(k)}$$

In practice we use a limit on the difference between consecutive values of the labels in order to decide when to stop the algorithm. Note that the limit values satisfy the conditions for complete fuzzy labelings.

One advantage of this simple algorithm is that it can easily be parallelized. Also, experimental results show that convergence occurs in a small number of steps. We will use the rest of this section to show some experimental results.

The first set of tests consisted in checking the number of steps required for convergence for a specially devised framework with overlapping complete extensions and an empty grounded extension (actual convergence of the algorithm toward the grounded extension is obvious from proposition 3).

Out of 20 tests on the given framework (containing 9 arguments with a symmetric structure of attacks) we got the results from table 1.

Steps needed	**Number of occurences**
6	2
7	1
8	3
9	12
10	2

Table 1. Number of steps needed for convergence on 20 tests on the same framework, but with different initial values

From this we can see that the number of steps needed for convergence is rather stable (does not depend on the initial labels of the arguments). Note however that the results are different (a different complete extension is obtained each time).

We have also tested with a different attack relation, but still with 9 arguments and noticed a larger variation in the number of steps, ranging from 9 to 22, which means that the important part of the framework with respect to convergence rate is the attack relation.

We have also tested against large frameworks and the table below shows that the size of the argumentation framework does not have a large impact on the number of steps needed for convergence (the increase is sublinear), as can be seen in table 2.

Number of arguments	**Steps needed (average)**
10	13.8
20	15
30	20.8
40	19.6
50	21.2
100	26.2

Table 2. Average number of steps needed for various number of arguments.

5 Fuzzy Labels as Certitude Factors

In the first approach we proposed for fuzzy labelings, the completeness rules focus on the strongest attacker of an argument. The second approach, to be described further, is aimed at considering all attackers.

For this, we think of the fuzzy labels as certitude factors for the acceptability of the argument or, in other words, as probabilities that the argument is acceptable. With this, the intuitive rule for complete certitude labelings follows.

Definition 6 (Complete certitude labeling). *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework. A complete certitude labeling of F is a labeling $\lambda : \mathcal{A} \to [0, 1]$ such that for all arguments a it holds that:*

- $\lambda(a) = \prod_{b \in \mathcal{A}, b \to a} (1 - \lambda(b))$*, if a has at least one attacker*
- $\lambda(a) = 1$*, if a has no attacker*

Let's consider again the simple framework

$$a \leftrightarrow b \to c$$

The rules of the definition translate into the following system of equations

$$\begin{cases} x = (1 - y) \\ y = (1 - x) \\ z = (1 - y) \end{cases}$$

From the system, we see that x can have any value, $z = x$ and $y = 1 - x$. The result is consistent with that of the previous approach, which is to be expected since we have at most one attack so the product and the maximum return the same value.

Proposition 4. *Let $F = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and λ a complete certitude labeling of F. Then $\lambda(a) = 1$ for all a in the grounded extension of F and $\lambda(a) = 0$ for all arguments a attacked by the grounded extension.*

Proof. Similar to that of proposition 3.

An interesting feature of this approach is the connection with SCC-s. If the framework is broken into multiple SCC-s, the computation can be performed for each SCC separately, starting with the ones that are not attacked by any other SCC. This opens the posibility of using this labeling in conjunction with an analysis based on SCC recursiveness.

6 Conclusions and Future Work

The main contribution of this paper consists in extending the labeling approach to real-valued labels and providing two approaches for working with these labels.

Also, this proposal links the argumentation domain with systems of equations and with convergent recursive sequences.

As future work we are looking for formal proofs to back up the experimental data that we have so far. We also aim to use this approach for describing semantics that are not complete.

Acknowledgements

This work was supported by project number IDEI1315/2008 and project number 264207/FP7-REGPOT-2010-1.

References

1. Pietro Baroni and Massimiliano Giacomin. Resolution-based argumentation semantics. In *Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008)*, pages 25–36, Tolouse, France, 2008.
2. Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, October 2005.
3. M Caminada. On the issue of reinstatement in argumentation. *Lecture Notes in Computer Science*, 4160:111, 2006.
4. Martin Caminada. Semi-Stable Semantics 1. In Paul E Dunne and Trevor J M Bench-Capon, editors, *Computational Models of Argument; Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.
5. Martin Caminada. Comparing Two Unique Extension Semantics for Formal Argumentation: Ideal and Eager. In *Proceedings of the 19th Benelux Conference on Artificial Intelligence (BNAIC 2007)*, pages 81–87, 2007.
6. Martin Caminada and Dov Gabbay. A Logical Account of Formal Argumentation. *Studia Logica*, 93(2):109–145, 2009.
7. Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Prudent Semantics for Argumentation Frameworks. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pages 568–572, Hong-Kong, China, 2005.
8. P Dung, P Mancarella, and F Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10-15):642–674, July 2007.
9. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
10. Jeroen Janssen, Martine De Cock, and Dirk Vermeir. Fuzzy Argumentation Frameworks. In *Proceedings of the 12th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008)*, pages 513–520, 2008.
11. Bart Verheji. Two approaches to dialectical argumentation : admissible sets and argumentation stages. In *Proceedings of the 8th Dutch Conference on Artificial Intelligence (NAIC 1996)*, number 1994, pages 357–368, Utrecht, Netherlands, 1996.
12. Zhihu Zhang and Zuoquan Lin. Enhancing Dung's Preferred Semantics. In *Proceedings of the 6th International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2010)*, pages 58–75, Sofia, Bulgaria, 2010.

ABA: Argumentation Based Agents[⋆]

A. Kakas[1], L. Amgoud[2], G.Kern-Isberner[3], N. Maudet[4], and P. Moraitis[5]

[1] University of Cyprus, antonis@ucy.ac.cy
[2] IRIT-CNRS at Toulouse, amgoud@irit.fr
[3] University of Dortmund, gabriele.kern-isberner@cs.uni-dortmund.de
[4] University Paris 9 Dauphine, maudet@lamsade.dauphine.fr
[5] Paris Descartes University, pavlos@mi.parisdescartes.fr

Abstract. Many works have identified the potential benefits of using argumentation in multiagent settings, as a way to implement the capabilities of agents (eg. decision making, communication, negotiation) when confronted with specific multiagent problems. In this paper we take this idea one step further and develop the concept of a fully integrated argumentation-based agent architecture. Under this architecture, an agent is composed of a collection of modules each of which is responsible for a basic capability or reasoning task of the agent. A local argumentation theory in the module gives preferred decision choices for the module's task in a way that is sensitive to the way the agent is currently situated in its external environment. The inter-module coordination or intra-agent control also relies on a local argumentation theory in each module that defines an internal communication policy between the modules. The paper lays the foundations of this approach, presents an abstract agent architecture and gives the general underlying argumentation machinery minimally required for building such agents, including the important aspects of inter-module coordination via argumentation. It presents the basic properties that we can expect from these agents and illustrates the possibility of this type of agent design with its advantages of high-level of flexibility and expressiveness.

1 Introduction

In recent years, many authors have promoted argumentation as a means to deal with specific multi-agent problems, for instance negotiation or communication with other agents. Indeed, recently argumentation has seen its scope greatly extended, so that it now covers many of the features usually associated to the theories of agency [30]. The benefits of argumentation are well established: a high-level of flexibility and expressiveness, allowing powerful and diverse reasoning tasks to be performed. In particular, different semantics can be used for different purposes without altering the underlying basic principles.

[⋆] This work grew out of the initiative of the 2008 Dagstuhl Workshop on the ”Theory and Practice of Argumentation Systems” to ask groups of researchers to propose ways of consolidating the work on several main themes of argumentation in Computer Science, such as the theme of argumentation in agents, which is the concern of this paper.

We study how this idea can be taken one step further to develop the concept of a fully integrated argumentation-based agent (ABA) architecture. The idea seems natural: for instance, to make the most of argumentation-based protocols, an agent should also demonstrate some argumentative reasoning capabilities. Similarly, for an agent to take informed and coherent decisions it needs to be able to argue about its choices by linking them to its underlying motivations and needs. What is missing is a global framework of how all these features could be glued together, both in terms of abstract design and technical specifications.

This paper lays the foundations of such an approach to agency, presents an abstract agent architecture obeying these principles, and gives the general underlying argumentation machinery minimally required for building such agents. In short, an agent is made of a collection of modules each of which is responsible for a basic capability or reasoning task of the agent. This is governed by a local argumentation theory in the module that gives preferred decision choices for the local task of the module, sensitive to the way the agent is currently situated in its external environment. The inter-module coordination and thus intra-agent control also relies on an argumentation theory that defines an internal communication policy between the modules. This gives an agent architecture that is coherently designed on an underlying argumentation based foundation.

From the early BDI architectures to the recent developments of computational logic based agents, the genealogy of agent architecture is now very dense. We can summarize the main objectives sought by the latest developments of agent architectures as follows:

- make the design easier (for instance by adopting readily understandable languages, or by semi-designing the agent, like introducing typical agent types [21]);
- bridge the gap between specification and implementation, the most typical case being the first BDI specifications vs. its concrete implementations (as noticed for instance by [24]);
- make agents more flexible and sensitive to external events [26], in particular going further than the classical "observe-think-act" cycle (as for instance the cycle theories in the KGP model [14] do);
- introduce new features not originally present in the architectures that now appear to be vital to autonomous agents (for instance social features [7] or learning [1])

We regard the adoption of a unified argumentation based architecture as highly positive regarding the first three issues, in particular. Our argumentation-based agent architecture is a high-level architecture that can also encompass other methods by transparently incorporating them in the architecture as black boxes that generate information or choices to be argued about. Its main concern is indeed to manage its different options by considering the arguments for and against in the light of the currently available information from the environment.

The argumentation basis of the ABA architecture does not depend on any specific argumentation framework but only requires some quite general properties of any such framework to be used. Irrespective of the framework used the argumentation-based foundation of ABA agents provides various advantages, including that of its rational or valued based decisions that facilitates the focus of purpose by the agent and the more effective interaction between agents which can explain their positions or requests.

Our work shares similarities with other argumentation based agent approaches when it comes to addressing specific issues and features of agents, e.g. in the recent KGP model of agency [14] goal decision and cycle theories for internal control are also captured through argumentation. However, the objective of assembling all these features in a single and coherent architecture uniformly based on argumentation has been the main challenge of our approach. The closest connection is with the work in [29] which proposes an Agent Argumentation Architecture (called AAA) and further developments of this in [18]. As in our case, argumentation is used as the primary means to arbitrate between conflicting motivations and goals. More specifically, in this work the high-level motivations of the agents are operationally controlled by *faculties*. These faculties make use of a dialogue game to arbitrate among the conflicting goals, depending on the consequences they foresee, or on favoured criteria of assessment. Also the recent work of Argumentative Agents [27] with their ARGUGRID platform uses argumentation as the main way to support an agent's decisions with particular emphasis on the process of negotiation between such agents.

The wider context of our work is that of modularly composed agent architectures with internal rationality for managing the different internal processes of the agent, as found for example in the works of [25] and [22]. In our proposed approach argumentation plays a central role both for the decisions within each module and for the interaction between the various agent modules. In particular, our approach offers an alternative way to view and possibly extend the use of bridge rules that other architectures use for the intra-agent reasoning.

The rest of the paper is as follows. In the next section we present the basic argumentation machinery for building ABA agents. Sections 3 and 4 present the abstract agent architecture and its intra-agent control. In Section 5, we detail some basic formal properties that we can expect from ABA agents, concluding in Section 6.

2 Argumentation Basics

The backbone of an ABA agent is its use of argumentation for decision making. Argumentation allows an agent to select the "best" or sufficiently "good" *option*(s), given some available information about the current state of the world and the relative benefits of the potential options. For instance, an agent may want to decide its best options of goals to pursue or partners to work with. We will denote with $\mathcal{O}$ the set of possible options of a decision problem. For simplicity of presentation, these options are assumed to be mutually exclusive and pairwise conflicting. For instance, an agent may want to choose between two possible partners, Alice and Carla, for carrying out a task. Thus, $\mathcal{O} = \{$Alice, Carla$\}$.

The overall value of any certain option can be judged through evaluating by means of several *parameters* how much this option conforms to the preferences of the decision maker. An agent may for instance choose between Alice and Carla on the basis of parameters such as reliability and generosity. Each agent is thus equipped with finite sets, $\mathcal{M}$, of parameters that are used in expressing the relative preferences or priority amongst options. This, as we will see below, is done using these parameters to parameterize the various options and the arguments that the agent has for these (c.f. with the Value-Based Argumentation in [2]). Parameters may not be equally important, for example the reliability of a partner may be more important than its generosity. Thus arguments for a partner that carry the parametrization of reliability will be preferred. We will denote by, $\geq$, a partial ordering relation on a set $\mathcal{M}$ of parameters reflecting their importance.

From the current state of the world, as perceived by an agent, *basic arguments* are built in favor of options in $\mathcal{O}$ and these are labelled using appropriate parameter spaces, $\mathcal{M}$, of the agent. Let $\mathcal{A}$ denote the set of all those arguments for a specific decision problem. Each argument supports only one option but an option may be supported by many arguments. Let $\mathcal{F} : \mathcal{A} \longmapsto \mathcal{O}$ be a function which associates to each argument, the option it supports. An argument highlights the positive features of each option, such as the parameters that label the option. For example, an argument in favor of Carla would be that she is generous, while an argument in favor of Alice would be that she is reliable. Let also $\mathcal{H} : \mathcal{A} \longmapsto (2^{\mathcal{M}})$ be a function that returns the parameters that label each argument. Since the parameters are not necessarily equally important, the arguments using them will in general have different strengths. For instance, if we assume that reliability is more important than generosity, then the argument that is based on reliability is stronger than the one that is based on generosity.

We will assume that the relative strength between arguments is based on the an underlying priority ordering on the parameter space that is used to label the arguments. Hence in what follows, $\succeq$, will denote a partial preorder on the set of arguments that expresses the relative strength of arguments, grounded in some way on the relation, $\geq$, on the parameter space of arguments. This lifting of the ordering on the parameters to an ordering on the arguments, that are labelled by the parameters, can be done is several ways and is in general application domain depended.

In most frameworks for argumentation we have two basic components: a set, $\mathcal{A}$, of arguments and an *attack* relation among them. This relation captures the notion of one argument conflicting with another and providing a counter-argument to it. In our case, arguments that support distinct options are conflicting since the options are assumed to be mutually exclusive. So, e.g., we might define that α_1 `Attacks` α_2 iff $\mathcal{F}(\alpha_1) \neq \mathcal{F}(\alpha_2)$, and $\alpha_1 \succeq \alpha_2$, for two arguments $\alpha_1, \alpha_2 \in \mathcal{A}$. This gives the following argumentation theory:

Definition 1 (ABA Argumentation theory). *An argumentation theory,* `AT`*, for decision making of an ABA agent is a tuple* $\langle \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{F}, \mathcal{H}, \geq, \succeq, \texttt{Attacks} \rangle$

where `Attacks` *is chosen by the specific argumentation framework that we base the agents on.*

The process of argumentation is concerned with selecting amongst the (conflicting) arguments the *acceptable* subsets of arguments. This notion of acceptability has extensively been studied by several papers, e.g. [9]. Indeed, there are different proposed semantics for evaluating arguments and the semantics of (maximal) acceptable arguments. One widely used form of such a semantics is based on the notion of *admissible* arguments. According to this semantics, a subset $\mathcal{B}$ of $\mathcal{A}$ is admissible and hence acceptable iff it satisfies the following requirements:

- it is not self attacking, i.e. there is no element of $\mathcal{B}$ that attacks another element of $\mathcal{B}$,
- for every argument $\alpha \in \mathcal{A}$, if α attacks (w.r.t. `Attacks`) an argument in $\mathcal{B}$, then there exists an argument in $\mathcal{B}$ that attacks an argument in $\mathcal{A}$.

Maximal admissible arguments, called *preferred extensions*, are then taken as the maximal acceptable extensions of a given argumentation theory. In an argumentation-based approach, the choice of the "best" option(s) among elements of $\mathcal{O}$ is based on the maximal acceptable arguments associated with the different options as follows.

Definition 2 (Best decision/option(s)). *Let* $\mathtt{AT} = \langle \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{F}, \mathcal{H}, \geq, \succeq,$ $\mathtt{Attacks}\rangle$ *be an argumentation theory for decision making,* $\mathcal{E}_1, \ldots, \mathcal{E}_n$ *its maximal acceptable extensions, and* $d \in \mathcal{O}$*. The option* d *is a possible best (or optimal) decision of AT iff* $\exists \alpha \in \mathcal{A}$ *such that* $\mathcal{F}(\alpha) = d$ *and* $\alpha \in \mathcal{E}_i$ *for some* $i = 1, \ldots, n$.

It is clear that the basic component of this decision theory is the preference relation $\geq$ on the set $\mathcal{M}$ of parameters. This relation may be context dependent on the current situation in which the deciding agent finds itself. For example, the preference of reliability over generosity applies in case the task to do is urgent, while generosity may take precedence over reliability in case the agent is short on resources (money). Furthermore, conflicts between preferences may arise, e.g when an agent is in a situation in which it has an urgent task and it lacks resources. Then our original decision problem for choosing an optimal option is elevated to the decision of which of the preferences is (currently) more important. We are thus faced with a new decision problem on choosing the best priority amongst the basic arguments to answer our original decision problem.

This new problem is of the same form as the decision problems that we have described above where now our options have the special form $m \geq m'$ or its conflicting one of $m' \geq m$, where m and m' are members of $\mathcal{M}$, or of the form $\alpha \succeq \beta$ where α and β are arguments, i.e. members of $\mathcal{A}$. Our argumentation theory thus contains *priority arguments* for these options capturing *higher order preferences*. We can then combine these two argumentation theories to have a single argumentation theory that contains both basic arguments for the object-level options and priority arguments for the relative importance of the parameters and arguments. This extension can be done in several ways, see e.g.

[17, 23, 16, 8]. In [23], where this problem was originally studied, and in [16], basic (object-level) arguments are constructed from rules which are given names or classified in types and then preference arguments are given as rules for a priority ordering between (the names of or the types of) two rules. Such priority rules can also be named or categorized and hence high-order preference can be given as rules for the priority between (lower-level) priority rules.

3 ABA Architecture

The ABA architecture's basic principle is to build an agent from a loosely coupled set of modules that are to a large extent independent from each other with no or minimal central control. Each module is based on an argumentation theory, concerning a certain internal task of the agent, that provides a policy of how to take decisions on this type of tasks. A module contains also another argumentation theory responsible for its involvement in the intra-agent control (IAC) of the agent. Together these local IAC theories in each module give (see the next section) a distributed high-level argumentation-based communication protocol under which the internal operation of the agent is effected. The modularity of the ABA agent approach aims to allow the easy development of an agent by being able to develop separately its modules adding further expertise to it as we see appropriate without the need to reconsider other parts of the agent. An ABA agent module is defined as follows.

Definition 3 (ABA Agent Module). *An ABA agent module is a tuple* $M = \langle IAC, T, R \rangle$ *where:*

- IAC *is an argumentation theory for intra-agent control,*
- T *is an argumentation theory for the task of the module,*
- $R = \langle P, C \rangle$ *where* P *and* C *are sets of names of other modules, the parent and child modules of* M *respectively.*

Each module, M, is based on its own argumentation theory, T, pertaining to its specialized task. This is an expert (preference) policy comprising, as we have described in the previous section, of arguments for the different choices parameterized in terms of preference criteria together with priority arguments on the relative importance of these criteria and hence also on the basic arguments that they parametrize. The information (basic and priority arguments) contained in the argumentation theories in the various modules is given to the agent at its initial stage of development and remains relatively static, although some parts may be further developed during the operation of the agent. The dynamic information of the agent is that of its view of the external world, as we shall see below. This also affects which part of the static information is applicable in each situation.

The sets P and C of a module express a dependence between the modules that captures a request-server relationship where the decisions taken by a parent module form part of the problem task of a child module. For example, a Planning module will be a child of a Goal Decision module since Planning decides on (or selects plans) to achieve the goals decided by GOAL DECISION. The IAC component will be described in more detail in the next section.

Definition 4 (ABA Agent). *An ABA agent is a tuple, $\langle Ms, Mot, WV\rangle$, where*

- $Ms = \{M_1, ..., M_n\}$ *is a set of ABA modules for the different internal capabilities of the agent,*
- Mot *is a module containing an ABA argumentation theory for the agent's Motivations and Needs,*
- WV *is a module that captures the current World View that the agent has about its external environment.*

The number of modules and the capability they each provide to the agent is not fixed but can vary according to the type of application that the agent is built for. However, the MOTIVATIONS AND NEEDS (Mot) and the WORLD VIEW (WV) modules are specialized modules that play a central role and are arguably required to design any ABA agent.

Motivations and needs. An ABA agent contains a special module, Mot, for governing its high-level Motivations and Needs. These in turn can play a role in the decisions of many different modules of the agent. The Mot module comprises of an ABA argumentation theory where, through a preference structure on the Needs of the agent that are parameterized by its Motivations and that also depends on the current world view of the agent, it decides on the current high-level Needs of the agent. It thus defines the current *Desires* of the agent that drive the behaviour of the agent. This is achieved through the use of Needs as a parameter space for the arguments in many of the other modules. For example, the concrete goals that an agent sets in its GOAL DECISION module are selected according to these desires and therefore they come to best serve these desires. One way to formulate the Motivations and Needs policy is to follow a cognitive psychology approach. In particular, as in [16], we can use Maslow's basic motivations $M_1, \ldots, M_5$ for human behaviour: M_1 = *Physiological*, M_2 = *Safety*, M_3 = *Affiliation or Social*, M_4 = *Achievement or Ego*, and M_5 = *Self-actualization or Learning*. The motivations policy is then an argumentation theory for the relative priority or strength of these motivational factors, dependent on the current world view.

Example 1. Consider Alice and her friends $\mathcal{A} = \{\text{Bill, Carla, Dave, Elaine}\}$. Let us suppose that Alice's current needs are $\mathcal{N}_{\mathcal{A}} = \{need_f, need_c, need_m,\ need_e\}$, where $f = food, c = company, m = money, e = entertainment$. The arguments for these may be labelled by the basic motivations in the following way: $\mathcal{H}(need_f) = \{M_1\}$, $\mathcal{H}(need_c) = \{M_3\}$, $\mathcal{H}(need_m) = \{M_2\}$, $\mathcal{H}(need_e) = \{M_5\}$. We will assume that the induced strength relation on the basic arguments for Alice's current needs renders the arguments for the needs of $food$, *company* and *money* acceptable, while the argument for *entertainment* is not. These acceptable needs form the current *desires* of Alice and are part of her current state. These then affect the argumentation in other modules of Alice which use the Needs to parameterize their arguments.

Example 2. Alice decides the high-level goals to serve these desires in her GOAL DECISION module. Given her current World View, she has basic arguments for the following set $\mathcal{D}_A$ of potential goals:

$$\mathcal{D}_A = \begin{cases} G_{cheap} : \text{Have a } \textit{cheap} \text{ dinner with company} \\ G_{free} \ : \text{Be taken out for dinner by someone} \\ G_{home} : \text{Have dinner alone at } \textit{home} \end{cases}$$

From the connections between goals and needs the basic arguments for these potential goals are labelled by the needs they each serve:

$$\begin{array}{l} A_c \text{ with } \mathcal{F}(A_c) = G_{cheap} \text{ and } \mathcal{H}(A_c) = \{need_f, need_c\} \\ A_f \text{ with } \mathcal{F}(A_f) = G_{free} \text{ and } \mathcal{H}(A_f) = \{need_f, need_c, need_m\} \\ A_h \text{ with } \mathcal{F}(A_h) = G_{home} \text{ and } \mathcal{H}(A_c) = \{need_f\} \end{array}$$

Alice makes use of her argumentation theory for determining the priority of these arguments by evaluating the parameter pertaining to each argument. This yields $A_f \succeq A_c \succeq A_h$, and so G_{free} is the only goal that has an acceptable argument and this is the current choice in the GOAL DECISION module.

Example 3. In order to achieve her goal G_{free}, Alice adopts a preferred plan Π_{free} —choice of restaurant, time of dinner etc. — from her plan library in a similar argumentation process. She chooses this plan using her argumentation theory for plan selection in her Plan module based on some parametrization of the plans and a priority ordering of these parameters. The chosen plan cannot be effected entirely by Alice as it requires resources from other agents (it contains the requests for the external resources for *money*, (req_m), and for *company*, (req_c)). Now Alice is faced with the problem of deciding which other (sets of) agent can best serve these requests. This is the task of the COLLABORATION module. In this she has arguments for different agent partners to provide needed resources. These arguments are labelled by a parametric space of agent profiles, such as: $\mathcal{M}_{profile}$ = {Reliable, Likeable, Generous, Boring, Parsimonious, Offensive, Wealthy} = $\{R, L, G, B, P, O, W\}$. In Alice's world view each of the other agents have a profile parametrization, e.g.: $\mathcal{P}_A$(Bill) = $\{R, P, B, W\}$, $\mathcal{P}_A$(Carla) = $\{R, L\}$, $\mathcal{P}_A$(Dave) = $\{O, G, B, W\}$. Alice's argumentation policy for the priority of arguments for the different partner agents makes use of these profile parameters by measuring the extent to which the profiles serve the requested resources. Here Dave is the only agent that has profile attributes (G, W) that serves req_m, and so there is just one acceptable argument and corresponding choice of Dave.

World view. The agent's world view is maintained in the WORLD VIEW module, WV, providing a common view of the current state of the world to all other modules of the agent. The basic arguments and priority arguments in the agent's other modules depend on the world view, thus making them context dependent and adaptable to changes in the external environment of the agent. The WORLD VIEW module is thus a special module in the ABA architecture responsible for this global task. It can be realized in different ways, e.g. in terms of beliefs and a process of belief revision as in a BDI architecture. Then the current beliefs give the current world view that grounds the arguments in the different modules of the agent. Nevertheless, the WV module can can also be based, if the designer so wishes, on an argumentation theory for Reasoning about Actions and Change, as shown for example in [15, 28]. In this the main arguments are those of forward and backward persistence in time of world properties and the preference structure is given by the time ordering of the times from which the persistence starts, e.g. forward persistence that is rooted at later time is stronger than other forward persistence rooted at an earlier time and analogously for backward persistence. The external environment feeds this module with new information on events and properties that have been observed at certain times. An argumentation process then gives the properties of the world that currently hold.

Figure 1 gives a picture of the overall general structure of the basic architecture of an ABA agent. During its operation an ABA agent is characterized by a current *internal state*.

Definition 5 (Agent State). *A state of an ABA agent, $\langle Ms, Mot, WV\rangle$, is a tuple $\langle V, \mathcal{D}\rangle$ where:*

- *V represents the current view of the world as given by WV,*
- *$\mathcal{D} = \{CS_{M_1}, ..., CS_{M_n}\}$ where each, CS_{M_i}, is a tuple $\langle D, L, S\rangle$, representing the current state of the module M_i, where D is its current decision, as given by its argumentation theory, T_i, L is the level of commitment on D and $S \in \{keep, abandon\}$ is the current status of the decision D.*

The level of commitment and status of a module's decision are maintained by the intra-agent control, IAC theory of the module, as we see in the next section.

Feasibility arguments. In deciding the status of a decision it is useful to make a distinction between *feasibility* arguments and *optimality* arguments that an agent can have against a given decision. Feasibility arguments attack the feasibility of a given decision based on current world view information (*e.g.*, the agent may learn that the server it tries to connect to is down), while *optimality* arguments are situation independent arguments for the value of a given decision (*e.g.*, the agent may prefer servers whose storage capacity of the server is above a certain threshold). Part of the world-view module will then enable feasibility arguments specific to the "reality of the situation" for the current decision. Typically, feasibility arguments will parameterize decisions as being: *available*, *currently unavailable* (the current world-view discards this decision but it may be available again later on), or *unavailable* (the world-view discards this decision for ever). These new arguments, $\mathcal{A}^{fea}(V)$, for (or mostly) against the current decision, enabled in the new world view V of the agent, will affect the (meta-level) decision of the IAC theory to keep or abandon the module's decision.

4 Intra-agent Control

The intra-agent control (IAC) of an ABA agent is effected through a *communication protocol* that governs the interaction between the different modules of the

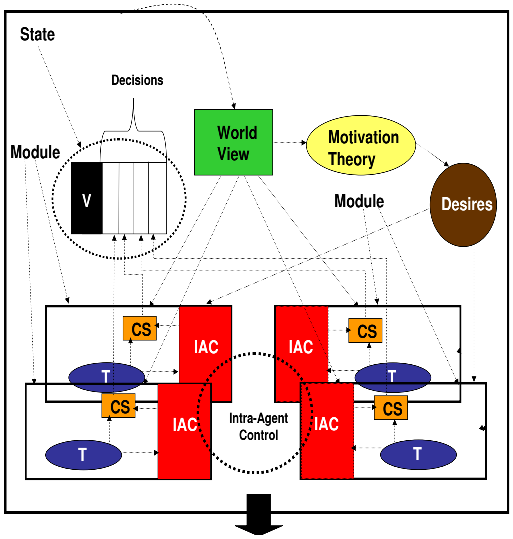


Fig. 1. ABA Architecture

agent. Through this protocol the modules pass messages between them (from parent to child and vice-versa) that in effect determine a distributed flow of control of the agent. For example, the Goal Decision module when it has decided on a new preferred goal it would send a message to its child module of Plan Decision, so that it would start the process of finding a preferred plan for it. Similarly, when a current (preferred) plan becomes untenable then the Plan Decision module would either decide on a new plan or inform the Goal Decision module thus prompting it to reevaluate and perhaps abandon this goal. As such there is no central control per se, except a mechanism for noting in the world view of the agent the passage of time and the changes in its external environment and distributing this to the other modules.

The IAC communication protocol is realized by endowing each of its modules its own ABA argumentation theory, IAC, responsible for governing its communication with the other modules. The basis of each of these IAC theories is (i) to decide *when to reconsider*, in the light of new information coming from the external environment either directly by a change in the current world view or indirectly through messages from other modules, the current decision of the module; and (ii) to decide *how to reconsider* these decisions, examining whether to *abandon or keep* them. Hence, the IAC as a whole, is responsible for updating the set $\mathcal{D}$ of current decisions in the internal state $\langle V, \mathcal{D} \rangle$ of the agent as its world view, V, changes. The IAC theories are argumentation theories of the following form.

Definition 6 (IAC Argumentation Theory). *The intra agent control theory of a module, M, is a tuple $\langle T_L, P_{Status} \rangle$ where:*

- *T_L is a theory for defining the commitment level, L, for the (object-level) decisions in M,*
- *P_{Status} is an ABA argumentation theory for the options $Keep(D)$ or $Abandon(D)$, with D a decision in M, that uses the commitments levels of T_L as parameters of its arguments.*

The levels of commitment, given by T_L, form (part of) the parametric space for the intra-agent control argumentation theory, P_{Status}, of the module. The arguments in P_{Status} for keeping or not a decision can be annotated (or even expressed) in terms of relative changes in these levels of commitment as time passes and new information from the external environment is acquired. The specific parameter space for the commitment levels and the type of theory T_L that assigns these are open to the designer. Nevertheless, the argumentation basis of an ABA agent under which its decisions are taken by its modules in the first place, allows us to define a natural form of commitment as follows.

Definition 7. *Let D be a decision of a module and $T(V)$ denote the module's argumentation theory T grounded on the current world view V. Then the current* commitment level *for D is given as follows:*

- *Level 5, iff D is uniquely sceptically preferred by $T(V)$, i.e. D holds in all maximal acceptable extensions of $T(V)$*
- *Level 4, iff D is credulously preferred by $T(V)$, i.e. D holds in one but not all maximal acceptable extension of $T(V)$*
- *Level 3, iff D does not hold in any acceptable extension of $T(V)$ but there exists a basic argument for D*
- *Level 2, iff D does not have a basic argument in $T(V)$*
- *Level 1, iff neither D nor any other alternative decision D' hold in any maximal acceptable extension of $T(V)$*

Hence the commitment level is a measure of the degree of acceptance (or optimality) of the decision with respect to the agent's optimality arguments for and against this decision in the argumentation theory T of the module. As the world view of the agent changes the structure of the module's argumentation theory, T, changes since different arguments and a different subset of the parameters that annotate the arguments are applicable. This then changes the degree of acceptance of the decision and hence its commitment level.

When and how to reconsider? The reconsideration of the commitment level of the current decision in a module every time that we apply the P_{Status} theory can be computationally non-effective. Under the above definition of commitment, the argumentation reasoning needed to reexamine the degree of acceptance of a decision can in general be costly. Hence to make the operation of P_{Status} more practical we can layer its decision process into two stages. In the first stage we apply a lightweight *Decision Reconsideration* policy that efficiently tells us whether we indeed need to reconsider the current decision. Only if the result from this is affirmative we continue to consider the full P_{Status} reasoning for deciding the fate of the current decision. Otherwise, we keep the current decision. The

Decision Reconsideration policy can be effectively constructed by considering a set of testing conditions that can trigger the possibility for a change in the level of commitment or degree of acceptance when this forms the commitment level. To be more specific, the degree of acceptance of a decision, D, in a module might decrease if new optimality arguments either against D, or in favour of another decision D' are enabled by V. Reconsideration should also be sensitive to the fact that a new feasibility argument against D, in $\mathcal{A}^{fea}(V)$, generated by a new world view, V, occurs. Likewise, the disabling in V of an argument in favour of D may lead to a reconsideration, and similar conditions for priority arguments can be specified. The *cautiousness level* specifies to which of these inputs the agent triggers the reconsideration process. Other factors may be used in this policy, in particular the *time* elapsed, denoted by t, from the time, t_0, that a decision was taken initially, with two important thresholds: t_α before which we have enough time to replace the decision and t_β after which it is too late to replace the decision ($t_0 < t_\alpha < t_\beta$). This allows us to design ABA agents with different characteristics whose operational behaviour can vary across the whole spectrum of "open" to "blind" BDI like agents and whose operation can be dynamically adapted to external changes. An "open" agent would be given by setting $t_\alpha = t_\beta = \infty$ whereas a "blind" agent by setting $t_\alpha = t_\beta = t_0$.

The role then of the argumentation theory component, P_{Status}, of the IAC theory, is to decide whether to keep or abandon the current (task) decision of the module by reexamining its commitment level or in effect by reexamining its degree of acceptance in the face of new information. The basic arguments of P_{Status} (denoted by $Arg([Keep|Abandon], D, level_1, level_2)$ can be built using the following underlying form:

- $keep(D)$ **if** the level of commitment of D is the same or increases
- $abandon(D)$ **if** its level of commitment decreases.

Example 4. The following arguments may define the default behaviour of a module of Alice: [Arg(Keep,D,5,4)] for keeping a decision D when its commitment level has fallen from 5 to 4 (since the decision is still acceptable in the module's theory) or an argument [Arg(Abandon,D,any,3)] for abandoning a decision when its commitment level falls to level 3 (as the decision is now not acceptable). Note though that there can be special circumstances, e.g. special types of decisions or extreme cases of the world view, when the opposite arguments might apply.

The argumentation reasoning of P_{Status} also depends on the current relevant feasibility arguments. For example, a child module may inform its parent module that the child's current decision is now at commitment level 1, i.e. that it can find no solution to the current problem that the parent module has sent it. This may be the result of information that the child module has received from the environment and/or from other modules. Thus a new feasibility argument is enabled in the parent module's P_{Status} theory, denoted by $[Arg(Abandon, D, c - unavailable)]$, for giving up its current decision D, for which it is informed that *currently* it cannot be effected in any way. The newly enabled feasibility arguments in P_{Status} can then be compared, via priority arguments in P_{Status}, with the other arguments based on the commitment level reexamination considered above. For example, should a module abandon its decision when it is informed by a child module that this cannot be (currently) achieved, even if its commitment level for this decision remains at the highest level? In other words, which is the stronger argument amongst the two basic arguments of [Arg(Keep,D,5,5)], which is based on the subjective evaluation of D, and [Arg(Abandon,D,c-unavailable)] based on objective information and under what conditions this is so? The preference structure of P_{Status} addresses such questions so that the IAC can weight up such different factors.

Example 5. We may capture the (default) preference to abandon currently unattainable decisions but not so when they are still optimally the most preferred ones with the priority arguments: [Pr1-Arg(Abandon,Keep)]: [Arg(Abandon,D, c-unavailable)] $\succ$ [Arg(Keep,D,L1,L2)] **if** $L2 \neq 5$ and [Pr2-Arg(Keep,Abandon)]: [Arg(Keep,D,L1,5)] $\succ$ [Arg(Abandon,D, c-unavailable)]. Of course, we may want to condition the second priority on the condition that there is still enough time for the world to change and make the decision D available again, e.g. for a collaborating agent to change its mind and make itself available.

With such priority arguments and the preference structure that follows from them, the designer of an ABA agent can give it a general strategy of operation, a characteristic of how to behave when the agent realizes that the implementation of its decisions in the external world has difficulties. Various factors relating to the cost or feasibility of replacing a decision can also be taken into account. For instance, the default argument to abandon decisions when they become relatively sub-optimal can be counter-balanced using another default argument for keeping decisions (as we want to also minimize loss of effort already done), such as: $[Arg(keep, D, default)]$: *keep*(D) **if** *expensive*(D), where *expensive*(D) is application dependent designating which (types of) decisions are costly to discard.

Example 6. To illustrate the various features of the IAC consider again the Alice example and suppose that Alice finds out that Dave has lost all his money and so W will not be in Dave's profile anymore. This disabling of an argument in favour of *Dave* can trigger the reconsideration, in the IAC theory of her Collaboration module, of her current decision for Dave. The decision to abandon or keep this decision depends on whether there are still acceptable arguments, w.r.t. the module's (task) argumentation theory, for Dave assigning commitment level at least 4 now, or whether there is no acceptable argument for Dave any more assigning commitment level 3 to him. Other feasibility arguments, e.g. arguments related to the time left before dinner, can also play a role in this decision. Should Alice decide to abandon Dave and the Collaboration module has no other choice of partner with an acceptable argument, then the parent module, i.e. the Plan module, will be notified which in turn will reconsider its current choice of plan using its own IAC theory. Similarly, this may eventually lead to Goal Decision module, to re-evaluate its current choice of goal and perhaps abandon this for a new goal to have a cheap dinner, or eat at home.

In general, the reconsideration of decisions and how this is communicated amongst the different parent and children modules of the agent will give an emergent behaviour on the operation of the agent. Under an ideally suited environment we expect that the IAC theory will induce a given pattern of operation on the agent, as we find in many of the proposed agent architectures, e.g. the fixed "Observe-Think-Act" cycle or the more general dynamic cycles given by the cycle theories of the KGP agents defined in [14]. In non-ideal conditions the particular operational behaviour of the ABA agent will be strongly dependent on these IAC theories in its modules.

The communication between modules based on the reconsideration of their decisions and subsequent messages that they send and receive between them can be defined as a form of an internal dialogue policy between the modules. In general, these control dialogue policies can be relatively simple. Nevertheless, it is important that the dialogues generated conform to several required properties of the operation of the agent, e.g. that there is no deadlock (where one module is waiting for a response from another module). We can then draw from the large literature on agent dialogue to ensure such consistency properties of the internal module dialogues. In particular, many of these approaches, e.g. [20, 4] are themselves based on argumentation and hence the link can be made more natural.

5 Properties of ABA Agents

ABA agents are designed so that their operation is based on informed decisions. The working hypothesis that underlies their operation is that the argumentation policies in an agent's different modules capture optimal solutions of the respective decision problems. The argumentation reasoning that they apply in taking their various decisions is such that agents evaluate the current alternatives against each other by comparing the reasons for and against these alternative choices. The acceptable choices in any module are meant to capture the best solutions available at the time. Hence the main property that an ABA agent must satisfy in its operation is that indeed this follows these informed choices. This is the central *soundness* property of an ABA agent in that it follows the intended design as captured in the decision policies of its modules.

In this section we define such desirable properties and indicate how we can design ABA agents (in particular their IAC theories) that would satisfy them.

Property 1. An ABA agent such that for any state, $\langle V, \mathcal{D} \rangle$, of its operation, every decision $D \in \mathcal{D}$ holds in a maximal acceptable extension of the argumentation theory, $T(V)$, of the corresponding module grounded in the state V, (i.e. D is optimal w.r.t. the policy in its module in the world state V), is called a **strongly sound agent.**

A strongly sound agent is therefore one whose decisions are not only optimal at the time that they are taken but remain optimal at any subsequent situation where its view of the world may have changed. It is easy to see that we can build such ABA agents by fixing their cautiousness at the highest level and designing their IAC to abandon decisions as soon as their commitment level falls below level 4 in the course of action and the passage of time. Indeed, let us choose the commitment level of a module's decisions to be given by the degree of acceptance of the decisions according to its (object level) expert policy theory as given in Definition 7. Then the high-level nature of the IAC theory allows us to specify, in the P_{Status} theory part of IAC, an argument: $[Arg(abandon, D, low)]$: $abandon(D)$ **if** $commitment\_level(D, V, C), C < 4$.

By giving, in the P_{Status} theory, to this argument higher-priority than any other argument (for keeping a decision) in P_{Status} we ensure that the IAC argumentation theory will always decide sceptically to abandon any decision when this is no longer preferred in the module's policy for choosing its decisions. In practice though in some applications this may be too strong to require as it may mean that decisions are abandoned too often. This can be mitigated, *e.g.*, by taking the cost induced by discarding this decision into account, or by requiring a weaker form of soundness where only some of the decisions are optimal throughout the operation of the agent. In particular, the higher level decisions in the "hierarchy" of modules, such as the goal decisions should remain optimal. Moreover, whenever any one of its goals is achieved (i.e. holds in the current state) then this should be optimal.

Property 2. An ABA agent such that for any state, $\langle V, \mathcal{D} \rangle$, of its operation, every goal decision, G, in $\mathcal{D}$ is acceptable in the state V, i.e. it holds in a maximal acceptable extension of the argumentation theory of the Goal Decision module grounded in the state V, is called a **sound agent**. Moreover, if whenever G holds in the current view of the world, V, the goal G is acceptable in the state V, then the agent is called a **sound achieving** agent.

Here we are assuming that once goals are achieved (as perceived by the agent in its world view) they are then immediately deleted from the state of the agent and that only goals that do not currently hold are added to the state. Achieved goals may later become suboptimal but this is beyond any reasonable requirement on the operation of an agent.

In effect all these properties of soundness are properties which require adaptability of the agent as it operates in an unknown environment. They require that the operation of the agent adapts to the new circumstances of the environment by changing its decisions accordingly. This high level of adaptability is facilitated in the ABA agents by the high level nature of their intra-agent control which allows them to recognize the changing status of decisions.

The above properties do not emphasize the overall internal coherency of the ABA agents as they are concerned with the individual internal decisions in each module. These individual choices need to be coherent with each other and give some overall sense to the agent's operations. This is given by the Motivations and Needs policy of the agent: the agent must operate in accordance to its current high-level desires and needs. We can therefore (re)formulate properties of soundness of the agent based on its motivations/desires.

Property 3. A **soundly motivated** ABA agent is an agent such that for any state, $\langle V, \mathcal{D} \rangle$, of its operation, and for every decision, D, in $\mathcal{D}$, D is acceptable in the state V with respect to the Motivations and Needs policy of the agent, whenever this policy is applicable to the corresponding module of D. In particular, its goal decisions in any state are always acceptable with respect to the Motivations and Needs policy of the agent.

Therefore a soundly motivated agent always operates according to the underlying motivations and needs policy that generates the agent's current desires. We can build such agents by suitably defining their IAC in a similar way to that of building sound agents, as shown above, where now instead of referring to the status of the decisions wrt object-level policy of the module we refer to the Motivations and Needs policy of the agent when this relates to the decision at hand. Indeed, we note that the soundly motivated property is essentially the only global consistency requirement that makes sense in an ABA agent, as there is no other global or explicit control of the agent.

6 Conclusions

The link between argumentation and multi-agent systems was originally viewed essentially as a way to manage the potentially conflicting knowledge bases of individual agents. With time this link has become much stronger covering several features of modern agency theories, *e.g.* negotiation, decision-making, communication. We have proposed an agent architecture uniformly based on argumentation with a highly modular structure. The focus is on a high-level architecture mainly concerned with managing the currently available best options for the agent's constituent tasks in a way that provides a coherent behaviour, with a focus of purpose, for the agent. This focus of purpose is governed to a certain extend by the agent's internal argumentation theory for its Motivations and Needs that gives the currently preferred high-level desires of the agent which in turn affect other decisions of the agent.

An important distinguishing characteristic of an ABA agent is that the agent's decisions are not rigid but rather they are decisions for currently preferred options or choices that its argumentation reasoning produces. These results of argumentation can be different under a different view of the world. This means that the agent is flexible and versatile in a changing environment, able to adapt graciously to changes in the agent's current situation, without the heavy need for an explicit mechanism of adaptation.

The aim of our work has been to present a high-level architecture based uniformly on argumentation which could then be used as a basis for developing such agents. This architecture and its argumentation basis does not depend critically on any specific argumentation framework but only requires some quite general properties of any such framework to be used. Different realizations can be developed by adopting anyone of the many concrete frameworks of argumentation that are now available, such as [23, 5, 13, 16, 2, 12], particularly those which are preference based. Also aspects from different approaches to argumentation can be exploited together within the ABA architecture. For example, the recent work of [11, 6] can be useful for the modular and distributed nature of the argumentation theories of the agent in its various modules. Moreover, the significant progress, over the recent years, in the study of the computational models of argumentation, e.g [13, 10, 3, 19], can provide a platform for the practical construction of ABA agents. Nevertheless, our work constitutes a first step in the proposal to build agents uniformly based on argumentation. A proper validation of the proposed ABA architecture can only be achieved by developing specific applications with ABA agents and evaluating their performance both in terms of capturing desirable properties of the agents and the approach as a whole and in terms of its computational viability.

6.1 Acknowledgements

We would like to thank the organizers of the 2008 Dagstuhl "Perspectives Workshop: Theory and Practice of Argumentation Systems" where this work was started. In particular we would like to thank Juergen Dix and Paul Dunne for participating in the initial discussions of this work.

References

1. S. Airiau, L. Padham, S. Sardina, and S. Sen. Incorporating learning in bdi agents. In *Proceedings of the ALAMAS+ALAg Workshop*, May 2008.
2. T. J. M. Bench-Capon. Value-based argumentation frameworks. In *NMR*, pages 443–454, 2002.
3. P. Besnard and A. Hunter. *Elements of Argumentation.* The MIT Press, 2008.
4. E. Blanck and K. Atkinson. Dialogues that account for different perscpectives in collaborative argumentation. In *Proc. 8th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 867–874, 2009.
5. A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.*, 93:63–101, 1997.
6. G. Brewka and T. Eiter. Argumentation context systems: A framework for abstract group argumentation. In *LPNMR*, pages 44–57, 2009.
7. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre. The boid architecture: conflicts between beliefs, obligations, intentions and desires. In *AGENTS '01*, pages 9–16, New York, NY, USA, 2001.
8. Y. Dimopoulos, P. Moraitis, and L. Amgoud. Theoretical and computational properties of preference-based argumentation. In *ECAI*, pages 463–467, 2008.
9. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence Journal*, 77:321–357, 1995.
10. P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.
11. P. M. Dung and P. M. Thang. Modular argumentation for modelling legal doctrines in common law of contract. *Artif. Intell. Law*, 17(3):167–182, 2009.

12. P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, 175(2):457–486, 2011.
13. A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. *TPLP*, 4(1-2):95–138, 2004.
14. A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Computational logic foundations of kgp agents. *J. Artif. Intell. Res. (JAIR)*, 33:285–348, 2008.
15. A. C. Kakas, R. Miller, and F. Toni. An argumentation framework of reasoning about actions and change. In *LPNMR*, pages 78–91, 1999.
16. A. C. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *AAMAS '03*, pages 883–890, 2003.
17. S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence Journal*, 2009.
18. M. Morge, K. Stathis, and L. Vercouter. Arguing over motivations within the v3a-architecture for self-adaptation. In *Proc. of the 1st International Conference on Agents and Artificial Intelligence (ICAART)*, pages 1–6, Porto, Portugal, 2009.
19. V. Noël and A. C. Kakas. Gorgias-c: Extending argumentation with constraint solving. In *LPNMR*, pages 535–541, 2009.
20. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261—292, 1998.
21. S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *AAMAS*, pages 394–401, 2002.
22. J. L. Pollock. Oscar: An architecture for generally intelligent agents. In *AGI*, pages 275–286, 2008.
23. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *J. of Applied Non-Classical Logics*, 7:25–75, 1997.
24. A. S. Rao and M. P. Georgeff. BDI-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, USA, 1995.
25. J. Sabater, C. Sierra, S. Parsons, and N. R. Jennings. Engineering executable agents using multi-context systems. *J. Log. Comput.*, 12(3):413–442, 2002.
26. M. C. Schut, M. Wooldridge, and S. Parsons. The theory and practice of intention reconsideration. *J. Exp. Theor. Artif. Intell.*, 16(4):261–293, 2004.
27. F. Toni. Argumentative agents. In *IMCSIT*, pages 223–229, 2010.
28. Q. B. Vo and N. Y. Foo. Reasoning about action: An argumentation - theoretic approach. *J. Artif. Intell. Res. (JAIR)*, 24:465–518, 2005.
29. M. Witkowski and K. Stathis. A dialectic architecture for computational autonomy. In *Agents and Computational Autonomy*, pages 261–274, 2003.
30. M. J. Wooldridge and A. Rao. *Foundations of rational agency.* Kluwer, 1999.

Arguing with Justifications Between Collaborating Agents

Ioan Alfred Letia[1] and Adrian Groza[1]

Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-400391 Cluj-Napoca, Romania
{letia,adrian}@cs-gw.utcluj.ro

Abstract. We exploit the Justification Logic capabilities of reasoning about justifications, comparing pieces of evidence, and measuring the complexity of justifications in the context of argumentative agents. Not knowing all of the implications of their knowledge base, agents use justified arguments for reflection and guidance.

1 Introduction

During argumentation, agents express relevant parts of their *knowledge* through *communicative acts*, which are *contextualized* to the cognitive state of the other party in order to be effective. The minimal framework in which all the elements of the vector

$$\langle \textit{Knowledge}, \textit{Dialog}, \textit{Reasoning about partner} \rangle$$

can be found is given by dynamic epistemic logic.

The role of knowledge in argumentation was stressed by Walton [18], who concludes that "argumentation theory lacks a workable notion of knowledge". One of the situations when argumentation occurs is due to the fact that the agents are not omniscient, which does not favor epistemic logic. Moreover, some implications can be triggered only by rational reflection or guidance [18]. In this study, the constructivist semantics of justification logic is exploited in order to overcome the omniscience problem: an agent cannot claim a formula without having actually constructed a proof term for it.

Argumentation theory did not pay much attention to modeling a mentalist approach of the interlocutor cognitive state [16]. In order to be effective, the content and the form of the conveyed communicative acts should be adapted to the other party. An agent can use its available evidence to persuade the other part about the issue in hand. Consequently, a means of describing "how evidence dynamics can be brought about as a result of communication" is needed [17]. To have proof-based evidence can be seen [17] as synonym to having deductive argumentation.

2 Distributed Justification Logic

This section extends the existing preliminary work regarding the application of justification logic to multi-agent systems [20, 17], by focusing on the expressiveness provided by the language in a multi-agent environment.

Justification Logic combines ideas from epistemology and the mathematical theory of proofs. It provides an evidence-based foundation for the logic of knowledge, according to which "F is known" is replaced by "F has an adequate justification". Simply, instead of "X is known" (KX) consider $t : X$, that is, "X is known for the explicit reason t" [7]. The multi-agent version extends justified logic by introducing an index to designate agents. Consequently $t :_i F$ is read as "based on the piece of evidence t the agent i accepts F as true". The minimum justification logic is axiomatized by axioms A_0 and A_1 in figure 1. The reflection axiom A_1 is logically equivalent with $\neg F \to \neg t :_i F$, meaning that no justification t exists for a false argument.

Definition 1. *The language $\mathcal{L}$ contains proof terms $t \in \mathcal{T}$ and formulas $\varphi \in \mathcal{F}$*

$$t ::= \; c \mid x \mid t \bullet t \mid t + t \mid !_i t \mid ?_i t \mid t \succ t$$
$$\varphi ::= \gamma \mid \varphi^* \varphi \mid \neg \varphi \mid t \gg_i \varphi \mid t :_i \varphi$$

A_0 classical propositional axioms

A_1 $t :_i F \to F$ (weak reflexivity)

A_2 $s :_i (F \to G) \to (t :_i F \to (s \bullet t) :_i G)$ (application)

A_3 $s :_i F \to (s + t) :_i F$ (sum)

A_4 $t :_i F \to !t :_i (t :_i F)$ (proof checker)

A_5 $\neg t :_i F \to ?t :_i (\neg t :_i F)$ (negative proof checker)

Fig. 1. Axioms of Justification Logic.

Evidence represents a piece of knowledge which may come from communication, perception, or from a agent's own knowledge base. Following [17], we distinguish two notions of evidence: the weaker notion of admissible, relevant justification $t \gg_i \varphi$, in which the agent i admits that t is an evidence for φ, and the stronger notion of probative or factive evidence $t :_i \varphi$, in which t is strong enough making the agent i to assert φ as a fact.

Proof terms t are abstract objects that have structure. They are built up from axiom constants c, proof variables x, and agent i' operators on justifications $\bullet$, $+$, !,?, described in figure 1. Such an evidence-based knowledge system (EBK) is based on the following assumptions: i) all formulas have evidence ($F \to t :_i F$), ii) evidence is undeniable and implies individual knowledge of the agent (A_1); iii) evidence is checkable (A_4 and A_5); iv) evidence is monotone, new evidence does not defeat

existing one (A_3) [2]. In order to adapt an EBK framework to an argumentative multi-agent system, considerations should be taken regarding the axioms A_1 and A_3, as follows.
Firstly, note that formula F is global in the multi-agent system; it is not related to any agent. In other words, if an agent $a \in \mathcal{A}$ considers t as relevant evidence to accept F, it means F should be taken as true by all the agents in $\mathcal{A}$. This not the case in real scenarios, where a different agent j might have different evidence that the opposite formula holds: $s :_j \neg F$.
Secondly, observe that the axiom A_3 in figure 1 encapsulates the notion of undefeasibility: if $t :_i F$, then for any other piece of evidence s, the compound evidence $t + s$ is still a justification for F. Our work regards weakening this constraint, by allowing agents to argue based on evidence with respect to the validity of a formula in a multi-agent system. This is in line with [18, 10], according to whom knowledge is incomplete and it remains open to further argument. The proposed distributed justification logic is axiomatised in figure 2.

A_0 classical propositional axioms
A'_1 $t :_{\mathcal{E}} F \rightarrow F$ (e-reflexivity)
A'_2 $s :_i (F \rightarrow G) \rightarrow (t :_j F \rightarrow (s \bullet t) :_k G)$ (distributed application)
A'_4 $t :_i F \rightarrow !^j t :_i (t :_i F)$ (positive proof checker)
A'_5 $\neg t :_i F \rightarrow ?^j t :_i (\neg t :_i F)$ (negative proof checker)
A'_6 $s :_i F \wedge t :_j F \rightarrow (s + t) :_i F,\ s + t \succ t$ (accrual)
A'_7 $F \rightarrow t :_i F$ (internalization)

Fig. 2. Distributed Justification Logic.

E-reflexivity. A given justification of F is factive (or adequate) if it is sufficient for an agent i to conclude that F is true: $t :_i F \rightarrow F$. Knowing that the weak reflexivity property has its merits when proving theorems in justification logic, we argue it is too strong in a multi-agent environment due to:

- if the agent i has evidence t for F it does not necessarily mean that F is a fact, for other agents may provide probative reasons for the contrary;
- the agents accept evidence based on different proof standards: whilst a credulous agent can have a "scintilla of evidence" standard, its partner accepts justification based on the "behind reasonable doubt" standard.
- the same evidence is interpreted differently by the agents in the system.

In our approach, a formula F is considered valid if all the agents in the system have justifications for F (their own or transferred from the other agents). The $E-$*reflexivity* axiom is read as: if every agent in the set E has justifications for F, F is a fact.

Distributed Application. In justified logic, the application operator takes a justification s of an implication $F \rightarrow G$ and an evidence t of its antecedent F, and produces a justification $s \bullet t$ of the consequent G [4]. In the existing multi-agents versions, the i index is introduced to represent the agent i, with the obvious meaning: if the agent i accepts the implication $F \rightarrow G$ based on s and F based on t, then agent i accepts G based on evidence $s \bullet t$ (axiom A_1). In a multi-agent setting, agents can construct their arguments based on justifications or evidence provided by their partners. Reasoning can also be performed based on the fact that the other agents rely their knowledge on a specific piece of evidence. The proposed generalized application operator A_1' allows agent k to construct its own evidence $s \bullet t$ based on the facts i) that the agent i has accepted the justification s as probative for $F \rightarrow G$ and ii) the agent j has accepted the evidence t to be sufficient to accept F.

Example 1. Assuming that agent a after some symptoms visits the physician p. Based on the consultation c, the physician decides there is evidence for the disease G and requests some analysis t to investigate F, which is needed to confirm the hypothesis $(F \rightarrow G)$. Agent a gets confirmation from the laboratory expert e. Consequently, he has the justification $c \bullet t$ to confirm G. The distributed application operator is instantiated as follows:

$$c :_p (F \rightarrow G) \rightarrow t :_e F \rightarrow (c \bullet t) :_a G$$

From the functional programming perspective, assuming that → is right associative, the distributed application operator has the following meaning: when an agent p provides a justification for $F \rightarrow G$, a function is returned which waits for the evidence t confirming F in order to output the justification $c \bullet t$ for G.

Recall, that $t :_i \varphi$ represents strong evidence, opposite to weak evidence $t \gg_i \varphi$. Consider that the laboratory analysis t confirming F may be contaminated, so the agent e accepts only as admissible the piece of evidence t. The corresponding expressiveness holds: "If you provide me defeasible evidence about F, I will have only admissible evidence about G:

$$c :_p (F \rightarrow G) \rightarrow t \gg_e F \rightarrow (c \bullet t) \gg_k G$$

The subjectivity about evidence can be also expressed: what is admissible for one agent is probative for the other one. In this case the agent a considers t as strong enough for F, the evidence transfer being modeled as

$$t \gg_e F \rightarrow t :_a F$$

Assuming that the agent p is the same with e in A_2', a simple justification based dialog takes place: "I have a justification for $F \rightarrow G$. When you provide me evidence or symptom of F, I will have a justification for G" .

$$s :_i (F \rightarrow G) \rightarrow t :_j F \rightarrow (c \bullet t) :_i G$$

Positive proof checker. Justifications are assumed to be verifiable. A justification can be verified for correctness, by the other agents or by the agent who conveyed it. $t :_i F \rightarrow !^j t :_i (t :_i F)$ is read as: if t is a justification for F accepted by the agent i, the agent j can check that piece of evidence. In case the agent checks itself $(j = i)$ we have *positive introspection*: $t :_i F \rightarrow !^i t :_i (t :_i F)$. It assumes that given evidence t for F, the agent i is able to produce a justification $!t_i^i$ for $t :_i F$. Thus, each justification has its own justification.
From the dialogical perspective, the positive proof checker is used to request for details why a formula is accepted based on a specific piece of evidence. The term $!^j t$ describes the agents i's evidence justifying $t :_i F$. Often, such meta-evidence has a physical form, such as a reference or email. Observe that the justification can be adapted to the agents who requested them: $!^j t :_i (t :_i F) \neq !^k t :_i (t :_i F)$. Here, the terms used by the agent i to describe the justification t for accepting F may not be equal $!^j t \neq !^k t$.

Negative proof checker. The negation in our framework is interpreted as follows:

$$\neg t :_i F \sim \textit{t is not a sufficient reason for agent i to accept F}$$

If t is not sufficient evidence for agent i to accept F, given by $\neg t :_i F$, the agent should have a justification for this insufficiency: $\exists\, q \in \mathcal{T}_i$ such that

$$\neg t :_i F \rightarrow q :_i \neg t :_i F$$

The operation ? gets a proof t and a formula F, and outputs a proof q justifying why p is not admissible evidence for F: $? : \textit{prof} \times \textit{proposition} \rightarrow \textit{proof}$. In case the agent checks itself $(j = i)$ we have *negative introspection*: $\neg t :_i F \rightarrow ?^i t :_i (\neg t :_i F)$

Accrual. The axiom A_6' says that if agent i has proved s for F and another agent j has evidence t for the same F, the joint evidence $s + t$ is a stronger evidence for the agent i to accept F, modeled by the preference relation $\succ$ over justifications: $t + s \succ t$. When $i = j$, the same agent has different pieces of evidence supporting the same conclusion.

Internalization. The internalization property assumes that formulas should be verifiable. It says that if F is valid, then there is a at least one agent i, which has accepted F based on the evidence t. From the argumentation viewpoint, every argument should have a justification in order to be supported. Consequently, self defending arguments are not allowed.
Note that, if F is a formula and t is an acceptable justification for agent i then $t :_i F$ is a formula. Thus, relative justifications of the form $s :_i (t :_j F)$ are allowed, where agent i has evidence s that agent j has evidence t for F. Similarly, the formula $t :_i F \rightarrow s(t)_i G$ says that: if t is agent i's justification for F, then $s(t)$ is agent i's evidence for G, where the argument t is inserted in the right place of argument $s(t)$. This

proof-based evidence for G is similar to have deductive argumentation supporting G [17].
Two rules of inference hold
$F, F \rightarrow G \vdash G$ (Modus Ponens)
$\vdash c : A$ (Axiom Internalization)
where A is an axiom and c is a constant. Similarly to [20] we assume that axioms are common knowledge.

3 Argumentation Framework

Firstly, one has to stress that having evidence for something is different from convincing someone of that issue. The justified claim can be rejected if it is too discrepant with the agent knowledge base or due to the lack of understanding of the evidence.
An argument A is consistent with respect to an evidence t if A does not contradict any evidence in t. We say that a piece of evidence t does not defeat evidence s of an agent i if $s :_i F \rightarrow (s + t) :_i F$.

Definition 2 (Undercutting defeater). *The evidence t is an undercutting defeater for F justified by s if the joint evidence $s + t$ does not support F any more. Formally: $s :_i F \rightarrow \neg(s + t) :_i F$.*

Corollary 1 (Justified undercutting defeater). *Note that the undercutting defeater is an implication, which is a formula in justified logic. So, based on the internalisation axiom A_7', it should have a justification: $q :_i (s :_i F \rightarrow \neg(s + t) :_i F)$. Informally, q is agent's i justification why the piece of evidence t attacks evidence s in the context of F formula.*

(m_1) Adam: The movie is a comedy. We should go.
(m_2) Eve: I like comedies. We can go.
How do you know that is it a comedy?
(m_3) Adam John told me.
(m_4) Eve: Then we should consider something else.
(m_5) Adam: Why?
(m_5) Eve: You know John, he laughs from everything.
(m_6) Adam: This usually happens. But it is not the case here.
(m_7) Eve: How is that?
(m_8) Adam: John told me the plot and it is really funny.
(m_9) Eve: You convinced me. Let's go then.

Fig. 3. Justified undercutting defeater.

Example 2. Consider the dialogue in figure 3. Here, m_1 represents Adam's justification for going to the movie: $m_1 :_A Go$. This information (m_1) combined by Eve with the fact that she likes comedies (m_2) is strong enough for *Eve* to accept the invitation: $(m_1 + m_2) :_E Go$. However, she checks for evidence that movie is a comedy: $!^E m_1 :_A m_1 :_A Go$. For *Eve*, the new evidence m_3 is the undercutting defeater for the m_1 justification:

$$(m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

Adam requests some justification, where the complete formulation "Why, given that you like comedies, the movie is a comedy you decided to come, but when you found that John told me this you have changed your mind?" is represented as

$$!^A q :_E (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

where $q = (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$ is the justification that should be provided by Eve to Adam for the above implication. Eve's justification comes from the m_5 message:

$$m_5 :_E (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E$$

Next, Adam confirms that this usually happens

$$m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4 \gg_E$$

but he does not consider the justification m_5 as strong enough:

$$\neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E$$

On Eve's request for justification, Adam provides the m_8 message:

$$m_8 :_A \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

which is eventually accepted by Eve:

$$m_8 :_E \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go.$$

According to axioms A_1' and

$$m_8 :_{\mathcal{E}} \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go.$$

one can state that:

$$\neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

which means that everybody agrees the evidence m_5 is not strong enough to defeat the *Go* formula supported by m_1 and m_2.

Definition 3 (Rebutting defeater). *The evidence t is a rebutting defeater for F if it is accepted as a justification for $\neg F$.*

Example 3. Consider the dialogue in figure 4. Here, *Eve* accepts as joint evidence m_1 and m_2 for the possibility to go: $(m_1 + m_2) \gg_{Eve} Go$. The evidence m_3 is a rebuttal defeater for attending the movie: $m_3 :_E \neg Go$. When Adam asks for clarifications ($?^A m_3 :_E m_3 :_E : \neg Go$) the m_5 message is provided: $m_5 :_E m_3 :_E \neg Go$, which is not considered by *Adam* as strong enough $\neg m_5 :_A (m_3 :_E \neg Go)$. When asking for evidence $?^E \neg m_5 :_A \neg m_5 :_A (m_3 :_E \neg Go)$, the m_8 justification is given: $m_8 :_A (\neg m_5 :_A (m_3 :_E \neg Go))$, which is accepted by Eve too: $m_8 :_E (\neg m_5 :_A (m_3 :_E \neg Go))$.

(m_1) Adam: The movie is a comedy. We should go.
(m_2) Eve: I like comedies. We might go. When does it start?
(m_3) Adam At 6'o clock.
(m_4) Eve: We cannot then.
(m_5) Adam: But why?
(m_5) Eve: I have to be home at 9'o clock.
(m_6) Adam: This is not a problem.
(m_7) Eve: How is that?
(m_8) Adam: The movie takes only 2 hours.
(m_9) Eve: Perfect. Let's go then.

Fig. 4. Justified rebutting defeater.

The following definition follows the Walton's [18] formalisation of knowledge.

Definition 4. *Knowledge represents justified acceptance of a proposition based on evidence and supported by rational argumentation to a specified standard of proof.*

This definition is accommodated in our framework by introducing an index representing the active standard of proof during the debate:

$t :_i^{\beta} F \simeq i$ *accepts* F *based on the evidence* t *under the standard of proof* β

An example of such standards occurs in trials: scintilla of evidence, preponderance of evidence, clear and convincing evidence, or behind reasonable doubt.

Example 4. Consider two standards of proof *scintilla of evidence* (α) and *preponderance of evidence* (β). The piece of evidence $\mathit{false\_alibi} :_j^{\alpha} \mathit{Guilty}$ is accepted by the judge j as a justification for *Guilty* when the active standard of proof is α, but the same justification is not enough to support guiltiness under the β standard: $\neg \mathit{false\_alibi} :_j^{\beta} \mathit{Guilty}$.

4 Argumentative Agents

We assume that: justifications are abstract objects which have structure, and agents do not lose or forget justifications [4].

The omniscience problem. The agents cannot always be expected to follow extremely long or complex argumentation chains [18], even if argumentation formalisms such as hierarchical argumentation frameworks [12], or the AIF ontology [14] do not specify any constraint on the size of argument. A constraint is imposed on proof terms that are too complex with respect to the number of symbols or nesting depth. In justification logic, the complexity of a term is determined by the length of the longest branch in the tree representing this term. The size of terms is defined in a standard way: $| c |=| x |= 1$ for any constant c and any variable x, $| (t \bullet s) |=| (t + s) |=| t | + | s | +1, |!t |=| t | +1$.

Lemma 1. *For each justified argument conveyed by agent i to agent j, agent j has a justification for accepting the argument or a justification for rejecting the argument:*

$$t :_i A \rightarrow s :_j A \vee r :_j \neg A$$

Preference over justifications. Agent i prefers evidence t_1 over t_2 to justify F is represented as $t_1 \succ t_2 :_i F$. It follows that at least t_1 should be an acceptable justification for F.

$$(t_1 \succ t_2) :_i F \rightarrow t_1 :_i F$$

The piece of evidence t_2 can be connected to F in the following ways: i) t_2 is also an accepted justification of F ($t_2 :_i F$), ii) t_2 is justification for the opposite formula $\neg F$, iii) t_2 is independent of the claim F.
Agent j can check why does his partner i prefer t_1 over t_2 to justify F:

$$!(t_1 \succ t_2) :_j (t_1 \succ t_2) :_i F$$

Agent i prefers justification t_1 over t_2 in the context of F based on evidence s:

$$s :_i (t_1 \succ t_2) :_i F$$

Agent i has a justification s why his partner j prefers evidence t_1 over t_2 as justification for F:

$$s :_i (t_1 \succ t_2) :_j F$$

Preference change over evidence can not be expressed without temporality. Based on the accrual axiom the following implications hold:

$$s :_i F \wedge t :_i F \rightarrow t + s \succ t :_i F, s :_i F \wedge t :_i F \rightarrow t + s \succ s :_i F$$

Assume that x is i's justification of A, whilst y is j's evidence regarding B.

Lemma 2. *A distributed proof term $s(x, y)$ can be constructed representing common justification accepted by the two agents to prove the intersection between A and B. Formally:*

$$x :_i A \wedge y :_j B \rightarrow s(x, y) :_{ij} (A \wedge B)$$

Communication of justifications. The following proof terms can be joined to express complex argumentative debates:

- Agent j has a justification r proving that agent i is inconsistent: $r :_j (t :_i F \wedge s :_i \neg F)$.
- Agent j has evidence showing that two agents disagree: $r :_j (t :_i F \wedge s :_k \neg F)$.
- The piece of evidence t does not defeat agent's i evidence s about F: $s :_i \rightarrow (s + t) :_i F$.
- Evidence conversion: $t :_i F \rightarrow t :_j F$. In other words, agent j trusts agent i's evidence regarding F.

5 Running scenario

The proof of concept scenario is a debate regarding the issue ”It is reasonable to accept the theory of evolution”[1]. Sets of arguments are exchanged during rounds between the instigator i and the contender c. Most of the burden of proof is carried by the instigator, however, the contender must defend his position that evolution is untrue ($\neg Evolution$).

Round 1. The instigator starts by stating the claiming formula, noted as *Evolution*. Based on the axiom A'_7 agent i should have evidence t to support his claim, under the standard of proof ”preponderance of evidence” (p). Formally,

$$Evolution \rightarrow t :_i^p Evolution$$

The contender accepts the challenge by stating his position ”Evolution doesn’t exist, but can you convince me?. This two pieces of information are formalized in distributed justified logic as ”$\neg Evolution$, respectively

$$!^c t :_i : t :_i Evolution$$

in which the agent c requests agent i to provide justifications.

Round 2. The instigator develops his speech by stating that: ”As an anthropology student, interested in human evolution, I have some education in this subject”, coded as $m_1 :_i (AntStud \rightarrow Education)$ and $m_2 :_i AntStudent$. Based on the application operator, a justification is derived from the sentence *Education*:

$$m_1 :_i (AntStud \rightarrow Education) \rightarrow m_2 :_i AntStud \rightarrow (m_1 \bullet m_2) :_i Education$$

where the compound justification $m_1 \bullet m_2$ is an instance of the argument from position to know. Then, he continues by pointing towards several categories of evidence and their bibliographic references: ”Evolution is well supported by evidence gathered from multiple fields of study: fossils, comparative anatomy, time and space distribution, computer simulations, and observation (2)(3)(4)(5)(6)”.
$(2) :_i fossils :_i Evolution$
$(3) :_i comp\_anat :_i Evolution$
$(4) :_i time\_space\_dist :_i Evolution$
$(5) :_i simulations :_i Evolution$
$(6) :_i obs :_i Evolution$
in order to strengthen the idea that ”Large amount of evidence support for evolution” ($LAEE$). A justification for it is constructed by applying the *accrual* axiom and checking the complexity of the resulting joint evidence.

$$(fossils + comp\_anat + time\_space\_dist + simulations + obs) :_i LAEE,$$

[1] Adapted from http://www.debate.org/debates/It-is-reasonable-to-accept-the-theory-of-evolution/1/

where large amount of evidence is a criterion to support evolution ($LAEE \rightarrow Evolution$). Note that the justification logic does not permit to include the evidences $(2) - (6)$ in the joint evidence, due to the right associativity of the operator (:) which gets a proof and a formula and returns a formula. The combination $(2) :_i fossils$ would not be a proper proof term of the language.
In addition, "The theory of evolution successfully predicts results in everything from fossils to psychology (9)(10)(13)." is noted as:

$$((9) + (10) + (13)) :_i fitsPrediction :_i Evolution$$

The last conveyed argument by the instigator in this round stresses the "lack of a better theory" and changes the burden of proof on the contender regarding this issue: "Can my opponent name a better theory?"

$$!^i q :_c q :_c (X \succ Evolution)$$

The link between preferred terms and preferred formulas can be:

$$(t_1 \succ t_2) \rightarrow (t_1 : F \succ t_2 : F)$$

The contender starts by clarifying that "Having evidence for something is different from convincing someone of something", denoted by

$$\neg[(t \gg_i F \rightarrow t :_i F) \wedge (t :_i F \rightarrow t \gg_i F)]$$

The justification for the above formula (refereed from now on as G) follows: "for one, they might not like what they hear and two, they might lack understanding":

$$don'tLike :_c G \vee don'tUnderstand :_c G$$

One example of attacking the arguments posted by the instigator follows: regarding fossils, the contender considers that "fossils are facts, and they are down for interpretation like all facts are. The fossils are not evident for evolution.": $fossilsAreFacts :_c \neg fossils :_c Evolution$.

6 Discussion and Related Work

There are many logics used to model argumentation: classical logic [5], defeasible logic [6], FOL [13], possibilistic logic [1], fuzzy logic [11], modal logic [8]. Modal logics lack the capacity to express the agents reasons for holding or changing their beliefs [17] and fail to represent the epistemic closure principle [4]. In our approach the complexity of the argumentation chain is limited by the complexity of the proof terms in justification logic. Yavorskaya's work [20] investigates certain interactions between the terms of different agents, such as "agent j can check agent i's evidence" or"agent j trusts agent i's evidence". Evidence accepted by the two agents are distinct: evidence terms are constructed from agent's own atomic evidence (only constants and variables), assuming that the operations on terms are the same, atomic evidence comes from its own vocabulary or ontology. In the current proposal, the agents have a common set of pieces of evidence $\mathcal{T}$ they can use to prove formulas, but the decision how to interpret these terms is left to each agent. Thus, the same piece of evidence $t \in \mathcal{T}$ can be probative for one agent $(t_i : F)$ and of no importance for the other $(\neg t :_j F)$.

Patterns of human reasoning are captured as argumentation schemes [19], whose structure consists of a set of premises, a conclusion, and a set of critical questions which can block the derivation of the consequent. Because justifications are abstract objects which also have structure, they can model such structured argumentation schemes. In this line, conveying a critical question can be seen as a justification for the fact that the set of premises are not enough evidence for supporting the consequent. Rebutting the issue raised by the critical question would be a valid justification for accepting the conclusion. We argue that the undercutting defeater formalized within the framework of justification logic handles the defeasible nature of the argumentation schemes. Moreover, the dialectical nature of the argumentation schemes [15] in the justification-based dialogues is exemplified here.

The link between epistemic logic and justification logic is stressed by the Platon's viewpoint of knowledge as justified true belief. By connecting justification logic with epistemic logic [3] epistemic schemes like: argument from common knowledge, argument from position to know, popular opinion (everybody knows), argument from ignorance (from lack of evidence) can be represented as structured proof terms in our framework.

When representing agent knowledge with ontologies, justifications are seen as the smallest set of premises that are sufficient for the entailment to hold and used as a mean to signal inconsistencies or to explain entailments to a broader audience of knowledge consumers [9]. In this context, justifications highlight only relevant knowledge in order to support the reasoning mechanism.

Our approach meets the requirements for initial conditions of knowledge in argumentation: i) knowledge bases are incomplete and inconsistent, ii) knowledge is defeasible, iii) knowledge is the result of a process of inquiry, iv) asserting something as knowledge depends on the current standard of proof.

7 Conclusion

This paper presents preliminary work on arguing based on justification logic. Even in its infancy, justification logic seems the adequate technical instrumentation to respond to the observations raised by Walton in [18]. The proposed framework extends Evidence Based Knowledge (EBK) systems, which are obtained by augmenting a multi-agent logic of knowledge with a system of evidence assertions [2], by including argumentation.

Akcnowledgements

We thank the anonymous reviewers for their numerous and detailed comments. This work was supported by CNCSIS-UEFICSU, project number PNII-Idei 170/2009 of the Romanian Ministry of Education and Research.

References

1. Alsinet, T., Chesñevar, C.I., Godo, L., Simari, G.R.: A logic programming framework for possibilistic argumentation: Formalization and logical properties. Fuzzy Sets and Systems 159(10), 1208–1228 (2008)
2. Artemov, S.: Justified common knowledge. Theoretical Computer Science 357(1-3), 4 – 22 (2006), clifford Lectures and the Mathematical Foundations of Programming Semantics
3. Artemov, S., Nogina, E.: On epistemic logic with justification. In: 10th conference on Theoretical aspects of rationality and knowledge. pp. 279–294 (2005)
4. Artemov, S.N.: Why do we need Justification Logic? Tech. Rep. TR–2008014, CUNY Ph.D. Program in Computer Science (Sep 2008)
5. Besnard, P., Hunter, A.: Argumentation based on classical logic. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 133–152 (2009)
6. Cohen, A., García, A.J., Simari, G.R.: Extending delp with attack and support for defeasible rules. In: Morales, Á.F.K., Simari, G.R. (eds.) IBERAMIA. Lecture Notes in Computer Science, vol. 6433, pp. 90–99. Springer (2010)
7. Fitting, M.: Reasoning with justifications. In: Makinson, D., Malinowski, J., Wansing, H. (eds.) Towards Mathematical Philosophy, Papers from the Studia Logica conference Trends in Logic IV, Trends in Logic, vol. 28, chap. 6, pp. 107–123. Springer (2009), published online November 2008
8. Grossi, D.: On the logic of argumentation theory. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S. (eds.) AAMAS. pp. 409–416. IFAAMAS (2010)
9. Horridge, M., Parsia, B., Sattler, U.: Justification oriented proofs in OWL. In: International Semantic Web Conference (2010)
10. Kuhn, D.: The Skills of Argument. Cambridge University Press (1991)
11. Letia, I.A., Groza, A.: Towards pragmatic argumentative agents within a fuzzy description logic framework. In: ArgMAS, Toronto, Canada (2010)
12. Modgil, S.: Hierarchical argumentation. In: 10th European Conference on Logics in Artificial Intelligence. pp. 319–332 (2006)
13. Moguillansky, M.O., Rotstein, N.D., Falappa, M.A., Simari, G.R.: Generalized abstract argumentation: Handling arguments in fol fragments. In: Sossai, C., Chemello, G. (eds.) ECSQARU. Lecture Notes in Computer Science, vol. 5590, pp. 144–155. Springer (2009)
14. Rahwan, I., Zablith, F., Reed, C.: Laying the foundations for a world wide argument web. Artif. Intell. 171(10-15), 897–921 (2007)
15. Reed, C., Walton, D.: Argumentation schemes in dialogue. In: H.V. Hansen, e.a. (ed.) OSSA. pp. 1–11 (2007)
16. Reed, C.: Representing dialogic argumentation. Knowledge-Based Systems 19(1), 22 – 31 (2006)
17. Renne, B.: Evidence elimination in multi-agent justification logic. In: Proceedings of the 12th Conference on Theoretical Aspects of

Rationality and Knowledge. pp. 227–236. TARK '09, ACM, New York, NY, USA (2009)

18. Walton, D., Godden, D.M.: Redefining knowledge in a way suitable for argumentation theory. In: Hansen, H. (ed.) Dissensus and the Search for Common Ground. pp. 1–13 (2007)
19. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press (208)
20. Yavorskaya, T.: Interacting explicit evidence systems. Theory of Computer Systems 43, 272–293 (2008)

Syncretic Argumentation for Multi-Agents by Lattice Homomorphism and Fusion

Yoshifumi Maruyama[1], Taichi Hasegawa[1],
Hajime Sawamura[2], and Takeshi Hagiwara[2]

[1] Graduate School of Science and Technology, Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 JAPAN
{maruyama, hasegawa}@cs.ie.niigata-u.ac.jp
[2] Institute of Science and Technology, Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 JAPAN
{sawamura, hagiwara}@ie.niigata-u.ac.jp

Abstract. In this paper, we describe a novel approach to the syncretic argumentation, which allows agents with different epistemology to engage in argumentation, taking into account the Golden Rule in the ethics of reciprocity and Confucius' Golden Rule. We address this new argumentation framework in two ways. One is by introducing the lattice homomorphism on truth-values (epistemic states) of propositions, and the new definitions of arguments justified under syncretized knowledge base. For the other, we first devise a new way of fusing two lattices through the lattice product, and then give a syncretic argumentation framework in which argumentation is done under the fused lattice.

1 Introduction

Various kinds of argumentation frameworks have been proposed so far in their own right or for a fundamental interaction mechanism for multi-agents [1][2]. They, however, are basically frameworks using two-valued knowledge base, or simply a fixed multi-valued one [3]. And agents engaging in argumentation have been assumed to have knowledge bases in the common knowledge representation language for argumentation. This assumption is not natural since even the world of agents is not homogeneous, having their own world recognition, that is, epistemology.

In this paper, we make a clean break with such a past assumption, directing to a more natural but complex settings of argumentation named "syncretic argumentation". By the term "syncretic argumentation", it is meant to be such an argumentation that each agent can have its own knowledge base, based on its own epistemology, and engage in argumentation with it. Back to the ancient, let us consider such a scene that Aristotle and Lao Tzu encounter, and argue about a proposition p. Perhaps, Aristotle might say p is definitely true with his two-valued epistemology $\mathcal{TWO} = \{\mathbf{f}, \mathbf{t}\}$, and Lao Tze might say p may hold with truth degree $\bot$ under his four-valued epistemology $\mathcal{FOUR} = \{\bot, \mathbf{t}, \mathbf{f}, \top\}$. In this setting, they turn out to find that they can not communicate with each other. This is not just a matter of difference of knowledge, but difference of a way of recognizing things (epistemology), world-view, logic, and so on. In this paper, we are interested in how agents can communicate with each other and attain an agreement among agents with different epistemology.

We address ourself to this problem by setting it on the ring of our own Logic of Multiple-valued Argumentation (LMA) [4] since its knowledge representation language for argumentation is Extended Annotated Logic programming (EALP) [4] that allows to represent various epistemology for propositions as truth-values. In EALP, agent epistemology is to be captured as truth-values and associated with a literal as in $p : \mu$, for example. Thus the annotation μ represents a mode of truth or epistemic state of the proposition p [4]. It should be noted that we use the term epistemology with a slightly different meaning from the ordinary philosophical one. We think that annotations assume epistemology of agents from the perspectives of the truth-values of propositions. Put it differently, truth-values is an apparatus for recognizing things or propositions.

We syncretize different agent epistemology in two ways and construct the syncretic argumentation frameworks. One is by introducing the lattice homomorphism on truth-values (epistemic states) of propositions, and the new definitions of arguments justified under syncretized knowledge base. The reasons for that are twofold. One is that annotations have a lattice structure that comes from the EALP construction [4]. The other is that the lattice homomorphism is a mapping which can yield an equal, fair and bilateral epistemological fusion in our context. This reflects an attitude against unilateralism, so that one can avoid a one-sided view of the world. For the other, we devise the new notions: the lattice fusion operator and fusion lattice that are induced through the lattice product, and can be considered as providing an alternative but amalgamative way towards syncretizing the difference of epistemic states of propositions. In either way, we hold such a standpoint that the total truth may be derived from the integration of all different epistemic viewpoints.

LMA on top of EALP is an argumentation framework that allows agents to participate in uncertain argumentation under uncertain knowledge bases if once the common annotation is shared among agents. It has various sorts of attack such as rebuttal, undercut, defeat, etc. that were defined reflecting multiple-valuedness. Now that the epistemological fusion has been finished, LMA can promote an argumentation among agents as usual [4].

The paper is organized as follows. In Section 2, we describe the syncretic argumentation framework by introducing the lattice homomorphism. This part constitutes an extension of our previous work [8] to multi-agents. In Section 3, we give a new theory on the lattice fusion and fusion lattice construction that are to provide another approach to the syncretic argumentation. In Section 4, we describe the basic ideas and advantages of the syncretic argumentation by the lattice fusion through a simple example of argumentation in LMA, and compare it with the method by the lattice homomorphism in Section 3. In the final section, we argue about some implications of the approach to the syncretic argumentation and future directions to further work.

2 Syncretic Argumentation by Lattice Homomorphism

In this section, we present a first approach to syncretic argumenttion that allows agents to participate in argumentation even if they have knowledge bases with their own annotations as truth-values that reflect agents' epistemic states of propositions. In the Logic of Multiple-valued Argumentation (LMA) [4], the annotation is a complete lattice. Naturally, we introduce the mathematical notion of a homomorphism between lattices. Such a homomorphism enjoys the order-preserving property, so that it guarantees agents to retain agents' epistemic structure when embedding one lattice to the other. We also consider the bi-directional homomorphism on lattices since it allows for a fair, unbiased and pluralistic argumentation, prohibiting unilateral one.

Then, we describe the new definitions to characterize the set of justified arguments, under the knowledge base reconstructed by the homomorphism on lattices.

2.1 Homomorphisms on complete lattices

Definition 1 (Homomorphism [5]). *Let* $< L, \vee_L, \wedge_L, \leq_L >$ *and* $< K, \vee_K, \wedge_K, \leq_K >$ *be complete lattices. A map* $h : L \rightarrow K$ *is said to be a homomorphism if* h *satisfies the following conditions: for all* $a, b \in L$,

- $h(a \vee_L b) = h(a) \vee_K h(b)$
- $h(a \wedge_L b) = h(a) \wedge_K h(b)$
- $h(0_L) = 0_K$ *for the least element*
- $h(1_L) = 1_K$ *for the greatest element*

For simplicity, we omit the suffix denoting a lattice from here on if no confusion arises in the context.

Example 1. Let us consider two typical lattices: the two-valued complete lattice $\mathcal{TWO}$ and the four-valued one $\mathcal{FOUR}$. The former is typical in the West, and the latter in the early philosophical literature and text of Buddhism [6]. $\mathcal{TWO} =< \{f, t\}, \vee, \wedge, \leq >, where\ f \leq t$ in Fig. 1, and $\mathcal{FOUR} =< \{\bot, \mathbf{t}, \mathbf{f}, \top\}, \vee, \wedge, \leq >$, where $\forall x, y \in \{\bot, \mathbf{t}, \mathbf{f}, \top\}\ \ x \leq y \ \Leftrightarrow\ x = y \ \vee\ x = \bot \ \vee\ y = \top$ in Fig. 2. Note that we associate the suffix with annotations to avoid ambiguity of the same annotation names, that is, t_2 represents the annotation t in $\mathcal{TWO}$ and t_4 represents the annotation t in $\mathcal{FOUR}$, for example. For these lattices, there can be two possible homomorphisms as shown in Fig. 3 and 4. Naturally, homomorphism 1 is a reasonable choice in this case, from the original meanings of the annotations t and f. The selection, however, usually depends on various factors such as argument purposes, argument domains and so on.

Given two lattices, there can be many lattice homomorphisms in general, and also there can be no lattice homomorphism. In the latter case, it turns out that agents can not syncretize their knowledge bases, resulting in no argumentation among them. In order to resolve this situation, we will turn to alternative lattice operations such as lattice product [5], or fusion in the next section.

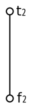


Fig. 1. 2-valued lattice

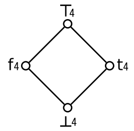


Fig. 2. 4-valued lattice

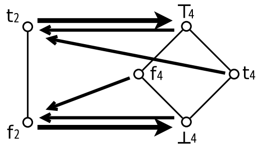


Fig. 3. Homomorphism 1

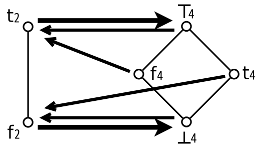


Fig. 4. Homomorphism 2

2.2 Syncretically justified arguments

With the lattice homomorphism above, we will illustrate how agents who have their own epistemology can reach an agreement and accept arguments through the grounded semantics or the dialectical proof theory of LMA [4].

Example 2. Suppose two agents A and B have the following knowledge bases respectively.

$K_A = \{\ \ a : t_2 \leftarrow,\ \ \sim b : t_2 \leftarrow,\ \ c : t_2 \leftarrow,\ \ \sim d : t_2 \leftarrow\ \ \}$
$K_B = \{\ \ \sim a : t_4 \leftarrow,\ \ b : t_4 \leftarrow,\ \ \sim c : \top_4 \leftarrow,\ \ d : \bot_4 \leftarrow,$
$e : t_4 \leftarrow g : f_4,\ \ g : t_4 \leftarrow\ \ \}$

Then the agents A and B can make the following set of arguments $Args_{K_A}$ and $Args_{K_B}$ from their knowledge bases respectively. (See [4] for the precise definition of arguments in LMA.)

$Args_{K_A} = \{\ [a : t_2 \leftarrow], [\sim b : t_2 \leftarrow], [c : t_2 \leftarrow], [\sim d : t_2 \leftarrow]\ \}$
$Args_{K_B} = \{\ [\sim a : t_4 \leftarrow], [b : t_4 \leftarrow], [\sim c : \top_4 \leftarrow], [d : \bot_4 \leftarrow],$
$[g : t_4 \leftarrow]\ \}$

The agents first assimilate their knowledge bases above to each other by the lattice homomorphism 1 in Fig. 3, and compute justified arguments from them using the grounded semantics or the dialectical proof theory [4], in each direction of the homomorphism as follows.

[1] Lattice homomorphism $h1$: $\mathcal{TWO} \rightarrow \mathcal{FOUR}$ (simply written as $\mathcal{T} \rightarrow \mathcal{F}$)
$h1(K_A) = \{\ \ a : \top_4 \leftarrow, \sim b : \top_4 \leftarrow, c : \top_4 \leftarrow, \sim d : \top_4 \leftarrow\}$
$K_B = \{\ \ \sim a : t_4 \leftarrow, b : t_4 \leftarrow, \sim c : \top_4 \leftarrow, d : \bot_4 \leftarrow, e : t_4 \leftarrow g : f_4, g : t_4 \leftarrow\ \}$
$Args_{h1(K_A)} = \{\ [a : \top_4 \leftarrow], [\sim b : \top_4 \leftarrow], [c : \top_4 \leftarrow],$
$[\sim d : \top_4 \leftarrow]\ \}$

$Args_{K_B} = \{ \ [\sim a : t_4 \leftarrow], [b : t_4 \leftarrow], [\sim c : \top_4 \leftarrow],$
$[d : \bot_4 \leftarrow], [g : t_4 \leftarrow] \ \}$

Note that $Args_{h1(K_A)} = h1(Args_{K_A})$ since the homomorphism preserves the lattice ordering. From these argument sets, the agents can have the following set of justified arguments (see [4] for the definition of justified arguments).
$Justified\_Args_{\mathcal{T}\to\mathcal{F}} = \{ \ [\sim b : \top_4 \leftarrow], [\sim d : \top_4 \leftarrow],$
$[b : t_4 \leftarrow], [d : \bot_4 \leftarrow], \ [g : t_4 \leftarrow] \ \}$

[2] Lattice homomorphism $h2$: $\mathcal{FOUR} \to \mathcal{TWO}$ (simply written as $\mathcal{F} \to \mathcal{T}$)
$K_A = \{ \ \ a : t_2 \leftarrow, \sim b : t_2 \leftarrow, c : t_2 \leftarrow, \sim d : t_2 \leftarrow \ \}$
$h2(K_B) = \{ \ \ \sim a : t_2 \leftarrow, \ \ b : t_2 \leftarrow, \ \ \sim c : t_2 \leftarrow, \ \ d : f_2 \leftarrow, \ \ e : t_2 \leftarrow g : f_2,$
$g : t_2 \leftarrow \ \ \}$
$Args_{K_A} = \{ \ [a : t_2 \leftarrow], [\sim b : t_2 \leftarrow], [c : t_2 \leftarrow], [\sim d : t_2 \leftarrow] \ \}$
$Args_{h2(K_B)} = \{ \ [\sim a : t_2 \leftarrow], [b : t_2 \leftarrow], [\sim c : t_2 \leftarrow],$
$[d : f_2 \leftarrow], [g : t_2 \leftarrow], [e : t_2 \leftarrow g : f_2, g : t_2 \leftarrow] \ \}$
Note that $Args_{h2(K_B)} \neq h2(Args_{K_B})$ in case of the homomorphism $h2$ since $[e : t_2 \leftarrow g : f_2, \ \ g : t_2 \leftarrow]$ has been qualified as an argument by $h2$ although its original form $[e : t_4 \leftarrow g : f_4, \ \ g : t_4 \leftarrow]$ in K_B is not an argument. From these argument sets, the agents can have the following set of justified arguments.
$Justified\_Args_{\mathcal{F}\to\mathcal{T}} = \{ \ [\sim d : t_2 \leftarrow], [d : f_2 \leftarrow],$
$[g : t_2 \leftarrow], [e : t_2 \leftarrow g : f_2, g : t_2 \leftarrow] \ \}$

Through the two-way homomorphism, we had two different sets of justified arguments: $Justified\_Args_{\mathcal{T}\to\mathcal{F}}$ and $Justified\_Args_{\mathcal{F}\to\mathcal{T}}$. Next, we are interested in defining a set of justified arguments as a ”common good” that is acceptable for both agents. In what follows, we present three kinds of agent attitudes or criteria to chose it from among two different sets of justified arguments.

Definition 2 (Skeptically justified arguments). *Skeptical justification is defined for each argument a in $Args_K$ as follows.*

- *An argument a in $Args_{K_A}$ is skeptically justified iff $a \in Justified\_Args_{\mathcal{F}\to\mathcal{T}}$ and h1(a) $\in Justified\_Args_{\mathcal{T}\to\mathcal{F}}$.*
- *An argument a in $Args_{K_B}$ is skeptically justified iff $a \in Justified\_Args_{\mathcal{T}\to\mathcal{F}}$ and h2(a) $\in Justified\_Args_{\mathcal{F}\to\mathcal{T}}$.*

This is a fair and unbiased notion of justified arguments in the sense that the both sides can attain a perfect consensus by the two-way homomorphism. Morally, it reflects such a compassionate attitude that agents look from the other agents' viewpoint, or place themselves in the other agents' position.

Example 3 (Example 2 cont.). The skeptically justified arguments in Example 2 are:
$Skeptically\_Justified\_Args = \{ \ [\sim d : t_2 \leftarrow], [d : \bot_4 \leftarrow],$
$[g : t_4 \leftarrow] \ \}$

A weaker version of skeptically justified arguments is the following. This criterion is not uninteresting since it gives a useful information on arguments which are not rejected completely.

Definition 3 (Credulously justified arguments). *Credulous justification is defined for each argument a in $Args_K$ as follows.*

- *An argument a in $Args_{K_A}$ is credulously justified iff either $a \in Justified\_Args_{\mathcal{F}\rightarrow\mathcal{T}}$ or h1(a) $\in$ $Justified\_Args_{\mathcal{T}\rightarrow\mathcal{F}}$.*
- *An argument a in $Args_{K_B}$ is credulously justified iff either $a \in Justified\_Args_{\mathcal{T}\rightarrow\mathcal{F}}$ or h2(a) $\in$ $Justified\_Args_{\mathcal{F}\rightarrow\mathcal{T}}$.*

Example 4 (Example 2 cont.). The credulously justified arguments in Example 2 are:
$Credulous\_Justified\_Args = \{\ [\sim b : t_2 \leftarrow], [\sim d : t_2 \leftarrow],$
$[b : t_4 \leftarrow], [d : \bot_4 \leftarrow], [g : t_4 \leftarrow]\}$

The third criterion is somewhat deviant reflecting a unilateral attitude, but it can be seen in our daily life often.

Definition 4 (Self-centeredly justified arguments). *Self-centered justification is defined for each argument a in $Args_K$ as follows.*

- *An argument a in $Args_{K_A}$ is self-centeredly justified iff $a \in Justified\_Args_{\mathcal{F}\rightarrow\mathcal{T}}$.*
- *An argument a in $Args_{K_B}$ is self-centeredly justified iff $a \in Justified\_Args_{\mathcal{T}\rightarrow\mathcal{F}}$.*

Example 5 (Example 2 cont.). The self-centeredly justified arguments in Example 2 are:
$Self-centerdly\_Justified\_Args = \{\ [\sim d : t_2 \leftarrow], [b : t_4 \leftarrow],$
$[d : \bot_4 \leftarrow], [g : t_4 \leftarrow]\ \}$

Which criteria are the most suitable to argument-based agent computing depend on agent purposes, agent attitudes, and so on. Here we just mention only a relationship of those criteria as follows. The proof is straightforward from the definitions.

Proposition 1. $Skeptically\_Justified\_Args \subseteq Self-centerdly\_Justified\_Args \subseteq Credulously\_Justified\_Args$

2.3 Created arguments

In the example 3, the argument $[e : t_2 \leftarrow g : f_2, g : t_2 \leftarrow]$ is included in $Justified\_Args_{\mathcal{F}\rightarrow\mathcal{T}}$, but its original $[e : t_4 \leftarrow g : f_4, g : t_4 \leftarrow]$ is not in $Args_{K_B}$. That is, a new argument has been created in the new world by $\mathcal{F} \rightarrow \mathcal{T}$. We single out for special treatment such arguments to distinguish from the preexisted arguments.

Definition 5 (Created arguments). *Arguments are created through the lattice homomorphism as follows.*

- *An argument a is said to be a creatively justified argument if $a \notin Args_{K_B}$ and $a \in Justified\_Args_{\mathcal{F}\rightarrow\mathcal{T}}$.*

- *An argument a is said to be a creatively justified argument iff* $a \notin Args_{K_A}$ *and* $a \in Justified\_Args_{\mathcal{T}\to\mathcal{F}}$.

Example 6 (Example 2 cont.). The creatively justified arguments in Example 2 are:
$Creatively\_Justified\_Args = \{\ [e : t_2 \leftarrow g : f_2, g : t_2 \leftarrow]\ \}$

Specifying "Creatively justified arguments" is not trivial since they reveal indiscernible arguments in ourselves by standing on each other's positions and ways of thinking. We also sometimes change our thinking or notice new ideas by standing on the opposite side of an argumentation in our daily life. It, however, leads to expanding the range of argumentation.

Creatively justified arguments turn to have only the property of the credulously justified arguments.

Proposition 2.

- *If an argument* $a \in Justified\_Args_{\mathcal{F}\to\mathcal{T}}$ *is a creatively justified argument, h1(a)* $\notin Justified\_Args_{\mathcal{T}\to\mathcal{F}}$.
- *If an argument* $a \in Justified\_Args_{\mathcal{T}\to\mathcal{F}}$ *is a creatively justified argument, h2(a)* $\notin Justified\_Args_{\mathcal{F}\to\mathcal{T}}$.

So far, we have given those definitions in a way specialized to the lattices $\mathcal{TWO}$ and $\mathcal{FOUR}$ for brevity of explanation. They can be carried on to any two lattices in a similar manner.

2.4 For more than 2 agents

We have described the first approach to syncretic argumentation undertaken by 2 agents. The method can be easily extended to the case of more than 2 agents. For example, in addition to the homomorphism between $\mathcal{TWO}$ and $\mathcal{FOUR}$ in Figure 3, let us consider the lattice $\mathcal{FIVE} =< \{\bot, \mathbf{t}, \mathbf{f}_1, \mathbf{f}_2, \top\}, \vee, \wedge, \leq >$. Then, we need to set up the following homomorphisms:

- $h1$: $\mathcal{TWO} \uplus \mathcal{FIVE} \to \mathcal{FOUR}$
- $h2$: $\mathcal{FOUR} \uplus \mathcal{FIVE} \to \mathcal{TWO}$
- $h3$: $\mathcal{TWO} \uplus \mathcal{FOUR} \to \mathcal{FIVE}$

based on the possible homomorphisms listed in Figure 5, 6, 7 and 8 plus the homomorphism in Figure 3, where $\uplus$ stands for disjoint union, and each least and greatest element in each lattice are mapped to those of the target lattice respectively. With these $h1, h2$ and $h3$, the new knowledge bases, argument sets and sets of justified arguments are to be constructed. Under these preparations, we can obtain skeptically justified arguments and credulously justified arguments as the results of the syncretic argumentation for the 3 agents society. We also can define a new notion of justification proper to multi-agents. For example, we can define that an argument A is *democratically justified* in the lattice field (such as $\{\mathcal{TWO}, \mathcal{FOUR}, \mathcal{FIVE}\}$) iff it is justified in more than or equal to the half number of the size of the lattice field. In either way, such an extension is a desideratum in the argumentation in more than 2 agents society.

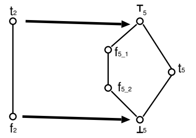


Fig. 5. $\mathcal{TWO} \rightarrow \mathcal{FIVE}$

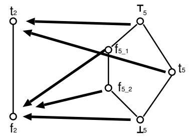


Fig. 6. $\mathcal{FIVE} \rightarrow \mathcal{TWO}$

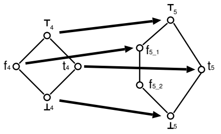


Fig. 7. $\mathcal{FOUR} \rightarrow \mathcal{FIVE}$

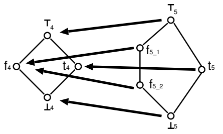


Fig. 8. $\mathcal{FIVE} \rightarrow \mathcal{FOUR}$

3 Lattice Fusion

In this section, we assume that agents have their own epistemology that is represented by annotation with a complete lattice structure as in the previous section, and consider how two different lattices can be fused by way of the lattice product [5]. In addition, we consider complete lattices as finite sets for the time being.

3.1 Product of complete lattices

Let L and K be ordered sets. The Cartesian product $L \times K$ can be made into an ordered set by imposing the coordinate-wise order defined by
$(x_1, y_1) \leq_{L\times K} (x_2, y_2)$ iff $x_1 \leq_L x_2$ and $y_1 \leq_K y_2$ for $x_i \in L$ and $y_i \in K (i = 1, 2)$.

Definition 6 (Product [5]). *Let $< L, \vee_L, \wedge_L, \leq_L >$ and $< K, \vee_K, \wedge_K, \leq_K >$ be lattices. For the product $L \times K$, we define $\vee_{L\times K}$ and $\wedge_{L\times K}$ as follows.*

- $(l_1, k_1) \vee_{L\times K} (l_2, k_2) = (l_1 \vee_L l_2, k_1 \vee_K k_2)$
- $(l_1, k_1) \wedge_{L\times K} (l_2, k_2) = (l_1 \wedge_L l_2, k_1 \wedge_K k_2)$

It should be noted that the product $L \times K$ is a lattice, $< L \times K, \vee_{L\times K}, \wedge_{L\times K}, \leq_{L\times K} >$ [5].

Example 7. Let us again consider two typical lattices: the two-valued complete lattice $\mathcal{TWO}$ and the four-valued one $\mathcal{FOUR}$ in Fig. 1 and Fig. 2 respectively. Then we have the product lattice as depicted in Fig. 9.

The product is a form of the combination of two different lattices: one component of an ordered pair from one lattice and another component from another lattice. The $\mathcal{TWO} \times \mathcal{FOUR}$ lattice consists of 8 $(= 2 \times 4)$ elements. Then, each element of $\mathcal{TWO}$ is associated with four elements of $\mathcal{TWO} \times \mathcal{FOUR}$ and each element of $\mathcal{FOUR}$ is associated with two elements of $\mathcal{TWO} \times \mathcal{FOUR}$. For example, $t_2 \in \mathcal{TWO}$ is associated with $(t_2, \top_4)$, (t_2, t_4), (t_2, f_4) and $(t_2, \bot_4)$, and $t_4 \in \mathcal{FOUR}$ is associated with (t_2, t_4) and (f_2, t_4).

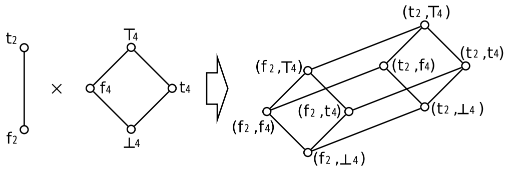


Fig. 9. Product of $\mathcal{TWO}$ and $\mathcal{FOUR}$

3.2 Fusion of complete lattices

The lattice product itself, however, can not be said to be a genuine fusion of lattices since it simply yields an ordered pair of two lattices, and even worse, agents do not have knowledge annotated by the product lattice. So we turn to devise a method that allows for a new lattice construct towards an intrinsic fusion of lattices, using the amount of the order information of the lattice product.

Definition 7. *Let L and K be lattices, and $L_1, \ldots, L_m$ and $K_1, \ldots, K_n$ be elements of L and K respectively. Then, we define the ordering relation $\leq_{L \otimes K}$ in-between an element of L, L_i $(1 \leq i \leq m)$ and an element of K, K_j $(1 \leq j \leq n)$.*

- $K_j \leq_{L\otimes K} L_i$ *iff* $\mid \{(L_s, K_r) \mid (L_i, K_r) \leq_{L\times K} (L_s, K_j), 1 \leq r \leq n, 1 \leq s \leq m\} \mid \leq \mid \{(L_s, K_r) \mid (L_s, K_j) \leq_{L\times K} (L_i, K_r), 1 \leq r \leq n, 1 \leq s \leq m\} \mid$
- $L_i \leq_{L\otimes K} K_j$ *iff* $\mid \{(L_s, K_r) \mid (L_i, K_r) \geq_{L\times K} (L_s, K_j), 1 \leq r \leq n, 1 \leq s \leq m\} \mid \leq \mid \{(L_s, K_r) \mid (L_i, K_r) \leq_{L\times K} (L_s, K_j), 1 \leq r \leq n, 1 \leq s \leq m\} \mid$

We use the notations $\leq_{L\otimes K}$ and $\geq_{L\otimes K}$ interchangeably, and omit the suffix $L \otimes K$ or $L \times K$ if no confusion arises.

Definition 8. $L_i =_{L\otimes K} K_j$ *iff* $L_i \leq_{L\otimes K} K_j$ *and* $K_j \leq_{L\otimes K} L_i$.

Definition 9. *Let L and K be lattices. A tuple $< L \cup K, \leq_{L\otimes K} >$ is a fusion of L and K, denoted by $L \otimes K$, where $L \cup K$ is a set in which $L_i \in L$ and $K_j \in K$ such that $L_i =_{L\otimes K} K_j$ are identified, and the original order relations $\leq_L$ and $\leq_K$ are preserved but with those order relations renamed to $\leq_{L\otimes K}$.*

Example 8 (Example 9 cont.). Let us consider the order between t_2 in $\mathcal{TWO}$ and $\top_4$ in $\mathcal{FOUR}$. We first pick up the ordered pairs including t_2 or $\top_4$, and compare them with each other as follows. We will write $\mathcal{T} \times \mathcal{F}$ and $\mathcal{T} \otimes \mathcal{F}$ for $\mathcal{TWO} \times \mathcal{FOUR}$ and $\mathcal{TWO} \otimes \mathcal{FOUR}$ respectively.

- $(t_2, \top_4) = (t_2, \top_4)$
- $(t_2, t_4) \leq (t_2, \top_4)$
- $(t_2, f_4) \leq (t_2, \top_4)$
- $(t_2, \bot_4) \leq (t_2, \top_4)$

- $(t_2, \top_4) \geq (f_2, \top_4)$
- (t_2, t_4) ? $(f_2, \top_4)$
- (t_2, f_4) ? $(f_2, \top_4)$
- $(t_2, \bot_4)$? $(f_2, \top_4)$

Note that the order relations: $=, \leq, \geq$ above have the suffix $\mathcal{T} \times \mathcal{F}$. From Definition 7, we have $t_2 \leq_{\mathcal{T} \otimes \mathcal{F}} \top_4$ since
$\mid \{(f_2, \top_4)\} \mid \leq \mid \{(t_2, t_4), (t_2, f_4), (t_2, \bot_4)\} \mid$. Similarly, we can obtain the order for all other elements as follows.

- $t_2 \leq \top_4$
- $t_2 \geq t_4$
- $f_2 \leq f_4$
- $t_2 \geq f_4$
- $t_2 \geq \bot_4$
- $f_2 \geq \bot_4$
- $f_2 \leq \top_4$
- $f_2 \leq t_4$

Note that the order relations: $=, \leq, \geq$ above have the suffix $\mathcal{T} \otimes \mathcal{F}$.

The Hasse's diagram of these relationship is shown in Fig. 10.

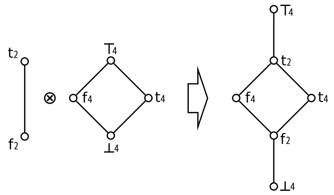


Fig. 10. Fusion of $\mathcal{TWO}$ and $\mathcal{FOUR}$

The ordering relation of Definition 7 is construed in a more lucid way as follows.

Proposition 3. *Let L and K be complete lattices, and $L_1, \ldots, L_m$ and $K_1, \ldots, K_n$ be elements of L and K respectively. For any $L_i \in L(1 \leq i \leq m)$ and $K_j \in K(1 \leq j \leq n)$,*

- $K_j \leq_{L \otimes K} L_i$ *iff*
  $\mid \{L_s \mid L_s \geq_L L_i,\ 1 \leq s \leq m\} \mid \times \mid \{K_r \mid K_r \leq_K K_j,\ 1 \leq r \leq n\} \mid \leq$
  $\mid \{L_s \mid L_s \leq_L L_i,\ 1 \leq s \leq m\} \mid \times \mid \{K_r \mid K_r \geq_K K_j,\ 1 \leq r \leq n\} \mid$
- $L_i \leq_{L \otimes K} K_j$ *iff*
  $\mid \{L_s \mid L_s \leq_L L_i,\ 1 \leq s \leq m\} \mid \times \mid \{K_r \mid K_r \geq_K K_j,\ 1 \leq r \leq n\} \mid \leq$
  $\mid \{L_s \mid L_s \geq_L L_i,\ 1 \leq s \leq m\} \mid \times \mid \{K_r \mid K_r \leq_K K_j,\ 1 \leq r \leq n\} \mid$

Proof. Refer to [7] for the proof.

Example 9 (Example 9 cont.). Let us examine the order between t_2 in $\mathcal{TWO}$ and $\top_4$ in $\mathcal{FOUR}$ by Proposition 3.

- $\{L_s \mid L_s \geq_{\mathcal{TWO}} t_2,\ 1 \leq s \leq 2\} \ = \{t_2\}$
- $\{K_r \mid K_r \leq_{\mathcal{FOUR}} \top_4,\ 1 \leq r \leq 4\} \ = \{\top_4, t_4, f_4, \bot_4\}$

- $\{L_s \mid L_s \leq_{\mathcal{TWO}} t_2,\ 1 \leq s \leq 2\} \ = \{t_2, f_2\}$
- $\{K_r \mid K_r \geq_{\mathcal{FOUR}} \top_4,\ 1 \leq r \leq 4\} \ = \{\top_4\}$

Therefore, we have $t_2 \leq_{\mathcal{T}\otimes\mathcal{F}} \top_4$ since $\mid \{t_2, f_2\} \mid \times \mid \{\top_4\} \mid \leq \mid \{t_2\} \mid \times \mid \{\top_4, t_4, f_4, \bot_4\} \mid$.

We have the following property when the fusion is a lattice. The proof is straightforward from Proposition 3.

Proposition 4. *Let L and K be complete lattices. Let $0_L, 0_K$ and $0_{L\otimes K}$ be the least elements of L, K and $L \otimes K$ respectively, and $1_L, 1_K$ and $1_{L\otimes K}$ be the greatest elements of them respectively.*

- $0_{L\otimes K} = 0_L$ *and* $1_{L\otimes K} = 1_L$ *iff* $\mid L \mid \geq \mid K \mid$
- $0_{L\otimes K} = 0_K$ *and* $1_{L\otimes K} = 1_K$ *iff* $\mid L \mid \leq \mid K \mid$

Proof. The proof is straightforward from Proposition 3.

Example 10. In Fig. 11, the case of the fusion of 3-valued lattice and 5-valued lattice is illustrated. These lattices represent the linear order of the grade points in a different way. Obviously, 5-valued-lattice allows for a finer grade than 3-valued-lattice. According to Definition 7 and Proposition 4, we have the fusion in which the greatest (least) element of 5-valued lattice, 5 (1) is located at the higher (lower) position than the greatest (least) element of 3-valued lattice, *good*(*bad*) respectively. And 3 and *normal* are located at the same position since $3 =_{3\otimes 5}$ *normal*. The resulting fusion gives a vivid account of the difference between fine and coarse recognition for the grade and a goodness of our amalgamation method.

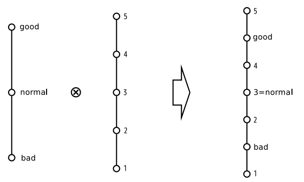


Fig. 11. Fusion of 3-valued-lattice and 5-valued-lattice

However, the problems of the fusion are two-fold. One is that the original orders of L and K are not necessarily preserved in the fusion of L and K. (Fortunately in Example 8 and 9, the fusion kept the order-preserving.) In this paper, we take such a standpoint that each agent should maintain their own epistemology since they have their own knowledge bases and arguments on the basis of their epistemology. The other is that the fusion does not always produce a lattice.

Fig. 12 illustrates the fusion of 4-valued lattice and 8-valued lattice in which the original orders of those lattices break down. In fact, the elements g and l in 8-valued lattice are non-comparable, but they turn out to be in the ordering relation $g \geq l$ in the fusion of two lattices. Furthermore, the fusion is not even a lattice. So we will consider a method to restore the fusion so that it preserves the original order and yields a (complete) lattice.

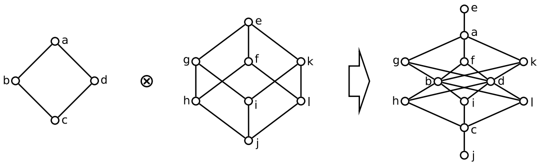


Fig. 12. Fusion of 4-valued lattice and 8-valued lattice

Definition 10. *Let L and K be lattices, and $L \otimes K = < L \cup K, \leq_{L\otimes K} >$ be a fusion of L and K, where $L \cup K$ is a set in which $L_i \in L$ and $K_j \in K$ such that $L_i =_{L\otimes K} K_j$ are identified. Then, the lattice $L \circ K = < L \cup K, \vee, \wedge, \leq_{L\circ K} >$ is said to be fusion lattice, where $\leq_{L\circ K} = S \cup \leq_L \cup \leq_K$ with $S \subseteq \leq_{L\otimes K}$ and $\mid S \mid$ being a maximum.*

The basic idea to restore the fusion to the fusion lattice so that it preserves the original order and yields a (complete) lattice is as follows. We first inherit the original orders of L and K in $L \circ K$ ($\leq_L \cup \leq_K$) since the fusion contains all elements of the original lattices, and then we prune some ordered pairs that were newly produced by 9, so that lub and glb are guaranteed for any two elements in $L \circ K$ and at the same time non-preexistent ordered pairs that were produced by the fusion are nullified. We employ the resulting fusion lattices that were obtained in the least steps of such a untangling pruning.

Proposition 5. *Let L and K be any lattice, and $L \otimes K = < L \cup K, \leq_{L\otimes K} >$ be the fusion of L and K. Then, the fusion lattice $L \circ K$ can be constructed from $L \otimes K$.*

Proof. Refer to [7] for the proof of the construction procedure and its correctness.

Example 11. Given the fusion in Figure 12, we can have four fusion lattices as the result of the restoration as shown in Figure 13. In the fusion lattices, the order relation $\leq_{4\circ 8}$ is $S \cup \leq_4 \cup \leq_8$ such that $S = \leq_{4\otimes 8} -\{b(d) \leq_{4\otimes 8} k,\ d(b) \leq_{4\otimes 8} g,\ b \geq_{4\otimes 8} i,\ d \geq_{4\otimes 8} i,\ b(d) \geq_{4\otimes 8} l,\ d(b) \geq_{4\otimes 8} h\}$ or $S = \leq_{4\otimes 8} -\{b(d) \leq_{4\otimes 8} k,\ d(b) \leq_{4\otimes 8} g,\ b \leq_{4\otimes 8} f,\ d \leq_{4\otimes 8} f,\ b(d) \geq_{4\otimes 8} l,\ d(b) \geq_{4\otimes 8} h\}$. For these two cases, $\mid S \mid = \mid \leq_{4\otimes 8} \mid - 6 = 32 - 6 = 26$ and $\mid S \mid$ is the maximum under the condition that S is compatible with the order relation $\leq_4$ and $\leq_8$. The choice from among two Ss is left to users' preferences.

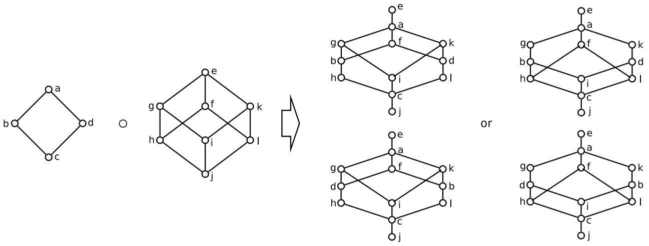


Fig. 13. Fusion lattices of 4-valued lattice and 8-valued lattice

3.3 Fusion lattice construction for infinite sets

So far we have restricted lattices to be fused to finite ones for the sake of easy construction. There can be some ways to remove them so that annotations may be infinite sets like a unit interval of reals $\Re[0,1]$. Fortunately, the fusion $\otimes$ yields the fusion lattice $\circ$ for such an infinite lattice. The following definition of the fusion lattice is for two infinite lattices with different intervals of reals, $R_1[a_1, b_1]$ and $R_2[a_2, b_2]$.

Definition 11. *Let $x_1 \in R_1[a_1, b_1]$ and $x_2 \in R_2[a_2, b_2]$. The ordering $\leq_{R_1 \otimes R_2}$ between x_1 and x_2 is defined in a similar way to the previous Definition 7, 8 and Proposition 3 as follows.*

- $x_1 \leq_{R_1 \otimes R_2} x_2$ *iff* $d(a_1, x_1) \times d(x_2, b_2) \ \leq d(x_1, b_1) \times d(a_2, x_2)$
- $x_2 \leq_{R_1 \otimes R_2} x_1$ *iff* $d(x_1, b_1) \times d(a_2, x_2) \ \leq d(a_1, x_1) \times d(x_2, b_2)$

  where $d(x, y)$ stands for the distance or segment between x and y on the real number line.

Figure 14 depicts the fusion lattice $T \otimes R$ of $\mathcal{TWO}$ and $\Re[0,1]$.

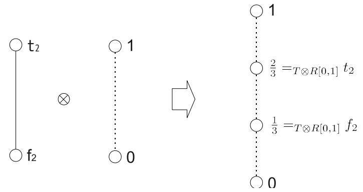

t_2
f_2
$\otimes$
1
0
1
$\frac{2}{3} =_{T \otimes R[0,1]} t_2$
$\frac{1}{3} =_{T \otimes R[0,1]} f_2$
0

Fig. 14. Fusion lattice $\mathcal{T} \otimes \mathcal{R}$ of $\mathcal{TWO}$ and $\Re[0,1]$

3.4 Advantages of the fusion and fusion lattice construction

Let L and K be lattices. We summarize characteristics and advantages of the fusion $L \otimes K$ and fusion lattice $L \circ K$ as follows.

- Majority rule: The fusion reflects a sort of majority rule for the agent epistemology by annotation as can be seen in Definition 7.
- Order preserving: The fusion $\otimes$ gives an ordering between the elements of L and those of K. The original orders of L and K are untouched by the fusion operator $\circ$. (See our former paper [8] for the contrary case by lattice homomorphism.)
- Commutativity: The products $L \times K$ and $K \times L$ determine the same fusion since $L \times K$ and $K \times L$ can be order-isomorphic. This is a most superior property of the fusion and hence lattice fusion since our fusion construction turns out yield an equal and fair argumentation among agents.

4 Syncretic Argumentation by Lattice Fusion

In this section, we illustrate the basic ideas and advantages of the syncretic argumentation by the lattice fusion through a simple example of argumentation in LMA, and compare it with the method by the lattice homomorphism in Section 2.

4.1 An example of the syncretic argumentation by the lattice fusion

Let us look at an argument about the plan of one day. Assume that the complete lattices of truth values of two agents' knowledge bases are the power sets $\mathcal{P}(\{A.M., P.M.\})$ and $\mathcal{P}(\{morning, daytime, evening\})$ ordered by set inclusion $\subseteq$ respectively. This means they have a different sense of time in a day. The result of the lattice fusion is shown in Fig. 15, where $m, d,$ and e stand for $morning, daytime,$ and $evening$ respectively. The construction of these lattices are basically the same as that in Fig. 13. Here we use two lattice fusions which preserve the lateral position relation of original lattices as in Fig. 15 since the original lattices are constructed in such a way that the elements which represents an earlier time in a day are positioned in the left side and the ones which represents a later time in a day are positioned in the right side. Thus we may need to choose a meaningful fusion lattice, in addition to the least restoration process. In this example, the result of the argumentation is the same by employing either fusion (the left one or the right in Fig. 15). In the both lattice fusions, the information of the node {A.M., P.M} includes the information of the nodes {morning, daytime} and {daytime, evening}, and {morning, daytime} includes {A.M.}. However, {daytime, evening} does not include {A.M.}.

Under these complete lattices, agents b and c have the following sets of knowledge bases respectively.

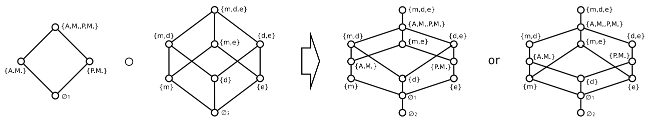


Fig. 15. Fusion lattices of $\mathcal{P}(\{A.M., P.M.\})$ and $\mathcal{P}(\{m, d, e\})$

$$\begin{aligned}
KB_b = \{\ & \mathit{leave}(a)\!:\!\{P.M.\} \leftarrow \mathit{wake\_up}(a)\!:\!\{A.M.\}\,\&\\
&\qquad\qquad\qquad\qquad \mathbf{not}\,\mathit{plan}(a)\!:\!\{P.M.\},\\
& \mathit{wake\_up}(a)\!:\!\{A.M.\} \leftarrow,\\
& \sim \mathit{plan}(a)\!:\!\{A.M., P.M.\} \leftarrow,\\
& \sim \mathit{weak}(a)\!:\!\{A.M.\} \leftarrow \quad \}\\
KB_c = \{\ & \sim \mathit{wake\_up}(a)\!:\!\{\mathit{morning}\} \leftarrow \mathit{weak}(a)\!:\!\{\mathit{morning}\},\\
& \mathit{weak}(a)\!:\!\{\mathit{morning}\} \leftarrow,\\
& \mathit{plan}(a)\!:\!\{\mathit{daytime}, \mathit{evening}\} \leftarrow \quad \}
\end{aligned}$$

wherein the annotated atom $\mathit{plan}(a) : \{\mathit{daytime}, \mathit{evening}\}$ reads "Agent a has a plan in the daytime and the evening", and $\sim \mathit{plan}(a) : \{A.M., P.M.\}$ reads "Agent a does not have a plan both in the morning and in the afternoon (perhaps agent a has a plan either in the morning or the afternoon)".

As the result of the argumentation based on these knowledge bases, we know that the argument which has the conclusion $\mathit{leave}(a)\!:\!\{P.M.\}$ ("Agent a should leave in the afternoon") is not justified by the dialectical proof theory [4] as shown in Fig. 16.

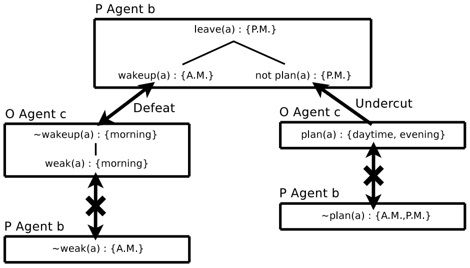


Fig. 16. A dialogue tree of the argumentation

In the dialogue tree, agents b and c are arguing about agent a's plan. Agent b begins with saying "Agent a will leave in the afternoon because he wakes up before noon and does not have a plan in the afternoon". Then agent c defeats it by saying "He can't usually wake up in the morning", and also undercuts by saying "He has a plan in the daytime and the evening". However, for these counter-arguments by agent c, agent b can not put forward further counter-arguments such as “a can wake up before noon” and “a does not have a plan in the morning and the afternoon” since $\{morning\} \leq \{A.M.\}$ and $\{daytime, evening\} \leq \{A.M., P.M.\}$. To be more specific, the argument “$plan(a) : \{daytime, evening\} \leftarrow$” by agent c does not atack the argument “$\sim plan(a) : \{A.M., P.M.\}$ $\leftarrow$” by agent b since although agent c says just about a’s plan in the daytime and the evening, agent b’s argument is the negation of a’s all-day plan. Consequently, the first argument of agent b is not justified in this argumentation. All the justified arguments we have in this syncretic argumentation is as follows.

$Justified\_Args = \{$
$[\sim plan(a) : \{A.M., P.M.\} \leftarrow],\ [\sim weak(a) : \{A.M.\} \leftarrow],$
$[weak(a) : \{morning\} \leftarrow],\ [plan(a) : \{daytime, evening\} \leftarrow]\}$

The complete lattice of truth values used in this example is irregular differently from the ones used in previous section. It allows for a temporal reasoning by argumentation. The example showed that our method have a due effect on the syncretic argumentation by the lattice fusion on various truth values as well.

4.2 Comparison to the syncretic argumentation by the lattice homomorphism

There is no work similar to this paper. So we compare the approach in this paper with our former work [8] by the lattice homomorphism, using the example of this section.

The possible bi-directional homomorphisms between the lattices $\mathit{4} =< \mathcal{P}(\{A.M., P.M.\}), \leq_{\mathit{4}}>$ and $\mathit{8} =< \mathcal{P}(\{m, d, e\}), \leq_{\mathit{8}}>$ are shown in Fig. 17 and 18.

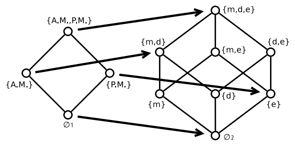


Fig. 17. $\mathcal{P}(\mathit{4}) \rightarrow \mathcal{P}(\mathit{8})$

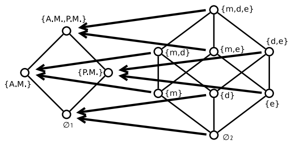


Fig. 18. $\mathcal{P}(\mathit{8}) \rightarrow \mathcal{P}(\mathit{4})$

Then, four types of justified arguments are calculated under the knowledge base embedding by the lattice homomorphisms according to the definition of [8] as follows.

$Skeptically\_Justified\_Args = \{$
$[\sim plan(a) : \{A.M., P.M.\} \leftarrow],\ [plan(a) : \{daytime,$
$evening\} \leftarrow]\ \}$

$Credulously\_Justified\_Args = \{$
$[\sim plan(a)\!:\!\{A.M., P.M.\} \leftarrow],\ [\sim weak(a)\!:\!\{A.M.\} \leftarrow],$
$[weak(a)\!:\!\{morning\} \leftarrow],\ [plan(a)\!:\!\{daytime, evening\}$
$\leftarrow]\ \}$
$Self - centerdly\_Justified\_Args = \{$
$[\sim plan(a)\!:\!\{A.M., P.M.\} \leftarrow],\ [weak(a)\!:\!\{morning\} \leftarrow],$
$[plan(a)\!:\!\{daytime, evening\} \leftarrow]\ \}$

$Creative\_Justified\_Args = \phi$

The fusion lattice contains more elements than the lattice targeted by the lattice homomorphism since the fusion lattice consists of all elements of the original lattices. Therefore, from the property of LMA, arguments are harder to defeat other arguments and more arguments will be justified in the argumentation by lattice fusion. In fact, the set of justified arguments of the argumentation by lattice fusion is equivalent to the set of Credulously justified arguments of the argumentation by lattice homomorphism which is maximal set in three kinds of justified arguments.

The lattice homomorphism h is weakly order-preserving in that for any a and $b \in L$, $a \leq b$ implies $h(a) \leq h(b)$, while the lattice fusion is strongly order-preserving in that the ordering and non-ordering relations are strictly preserved. Which approach we should use in the argumentation depends on the situation and the purpose of the argumentation. If agents do not so much emphasize their sense of value and can accept opponents' attitudes, they may use the lattice homomorphism. On the other hand, if they insist on their sense of value and epistemology, they may use the lattice fusion.

5 Concluding Remarks and Future Work

In this paper, we have undertaken two attempts to a new argumentation framework named syncretic argumentation. Actually, we presented two complementary approaches to it: the syncretic argumentation by lattice homomorphism and by the lattice fusion. The former in particular allows to syncretize the agent epistemology even for more than 2 agents.

Agents have to live in the multi-cultural computer-networked virtual society as well as humans living in the global multi-cultural society. This implies that agents also get involved in arguing about issues of mutual interest on the basis of their own belief and knowledge. But, if they insisted only on their epistemology, we would lose chances to interact or communicate with each other. The enterprise in this paper is an attempt to avoid such a cul-de-sac appearing even in argument-based problem solving.

There has been no work on argumentation frameworks in which each agent has its own knowledge representation language, its own epistemology, and its own argumentation framework. They have been all common to agents who participate in argumentation. Our work goes to the polar opposite direction from the perspective of the past works. In the area of *ontology mapping* [9], they claim that a single ontology is no longer enough to support the tasks envisaged by a distributed environment like the Semantic Web, and multiple ontologies need to be accessed from several applications. In a very general sense, our work might deal with issues similar to those in the ontology mapping, but we have not found any technical relationship to the ontology mapping in which epistemology mapping like in this paper is not concerned with. We have not considered a morphism of ontological signatures (vocabulary), which we think is needed for realizing a full-fledged syncretic argumentation.

In the near future, we will undertake mainly two major works: (i) introducing other types of lattice operations such as sum, and a common ground like the unit interval of reals $[0, 1]$ to which every lattice is mapped by the homomorphism, in order to produce more versatile and well-rounded arguments, (ii) extending the syncretic argumentation to the case of more than two agents, such as $L \circ K \circ M$, and the case of infinite sets of annotations such as the unit interval of reals $[0, 1]$. It is expected that the incorporation of the syncretism into LMA as well as the past argumentation systems could allow to expand application domains extensively.

References

1. Rahwan, I., Simari, G.R.E.: Argumentation in Artificial Intelligence. Springer (2009)
2. Prakken, H., Vreeswijk, G.: Logical systems for defeasible argumentation. In: In D. Gabbay and F. Guenther, editors, Handbook of Philosophical Logic, Kluwer (2002) 219–318
3. Chesñevar, C.I., Simari, G., Alsinet, T., Godo, L.: A logic programming framework for possibilistic argumentation with vague knowledge. In: Proc. of the Intl. Conference on Uncertainty in Artificial Intelligence (UAI2004). (2004)
4. Takahashi, T., Sawamura, H.: A logic of multiple-valued argumentation. In: Proceedings of the third international joint conference on Autonomous Agents and Multi Agent Systems (AAMAS'2004), ACM (2004) 800–807
5. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge (2002)
6. Sawamura, H., Mares, E.: How agents should exploit tetralemma with an eastern mind in argumentation. In: Mike Barley and Nik Kasabov (eds.): Intelligent Agents and Multi-Agent Systems VII, Lecture Notes in Artificial Intelligence, Springer. Volume 3371. (2004) 259–278
7. Hasegawa, T.: Syncretic argumentation by means of lattice fusion. `http://www.cs.ie.niigata-u.ac.jp/Paper/Storage/jurisin2009.pdf` (2009) Master Thesis, Niigata university.
8. Hasegawa, T., Abbas, S., Sawamura, H.: Syncretic argumentation by means of lattice homomorphism. In: Principles of Practice in Multi-Agent Systems, 12th International Conference(PRIMA 2009). Volume 5925 of Lecture Notes in Computer Science., Springer (2009) 159–174
9. Kalfoglou, Y., Schorlemmer, W.M.: Ontology mapping: The state of the art. In Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., Uschold, M., eds.: Semantic Interoperability and Integration. Number 04391 in Dagstuhl Seminar Proceedings (2005)

Using argumentation to reason with and about trust

Simon Parsons[1,2], Elizabeth Sklar[1,2], and Peter McBurney[3]

[1] Department of Computer & Information Science, Brooklyn College,
City University of New York, 2900 Bedford Avenue, Brooklyn, NY 11210, USA
{sklar,parsons}@sci.brooklyn.cuny.edu
[2] Department of Computer Science, The Graduate Center
City University of New York, 365 5th Avenue, New York, NY 10016, USA
[3] Department of Informatics, King's College London
Strand, London WC2R 2LS, United Kingdom
peter.mcburney@kcl.ac.uk

Abstract. Trust is an approach to managing the uncertainty about autonomous entities and the information they store, and so can play an important role in any decentralized system. As a result, trust has been widely studied in multiagent systems and related fields such as the semantic web. Here we introduce a simple approach to reasoning about trust with logic, describe how it can be combined with reasoning about beliefs using logic, and demonstrate its use on an example. The example highlights a number of issues related to resolving weighted arguments.

1 Introduction

Trust is an approach to managing the uncertainty about autonomous entities and the information they deal with. As a result, trust can play an important role in any decentralized system. As computer systems have become increasingly distributed, and control in those systems has become more decentralized, trust has become steadily more important within Computer Science [4, 18].

Thus, for example, we see work on trust in peer-to-peer networks, including the EigenTrust algorithm [22] — a variant of PageRank [34] where downloads from a source play the role of outgoing hyperlinks and which is effective in excluding peers who want to disrupt the network — and the work in [1] that prevents peers from manipulating their trust values to get preferential downloads. [52] is concerned with manipulation in mobile ad-hoc networks, and looks to prevent nodes from getting others to transmit their messages while refusing to transmit the messages of others.

The internet, as the largest distributed system of all, is naturally a target of much of the research on trust. There have been studies, for example, on the development of trust in ecommerce [31, 43, 51], on mechanisms to determine which sources to trust when faced with multiple conflicting sources [10, 39, 50], on mechanisms for identifying which individuals to trust based on their past activity [2, 20, 27], and on the manipulation of online recommendation systems [25]. The work we have just cited can be thought of as helping agents to decide who is worthy of trust. A development from a slightly different perspective — that of making it possible to trust individuals who might otherwise be deemed untrustworthy — is the idea of having individuals indemnify each other by placing some form of financial guarantee on transactions that others enter into [8, 9]. Thus I might indemnify you against a third party that I trust, thus making you feel comfortable doing business with them.

Trust is an especially important issue from the perspective of autonomous agents and multiagent systems [48]. The premise behind the multiagent systems field is that of developing software agents that will work in the interests of their owners, carrying out their owners' wishes while interacting with other entities. In such interactions, agents will have to reason about the amount that they should trust those other entities, whether they are trusting those entities to carry out some task, or whether they are trusting those entities to not misuse crucial information. As a result we find much work on trust in agent-based systems [45, 49], including work that identifies weaknesses in some of the major trust models [46].

In the work in this area, it is common to assume that agents maintain a *trust network* of their acquaintances, which includes ratings of how much those acquaintances are trusted, and how much those acquaintances trust their acquaintances, and so on. One natural question to ask in this context is what inference is reasonable in such networks. The propagation of trust — both the transitivity of trust relations [44, 49] and more complex relationships like "co-citation" [19] — has been studied. In many cases this work has been empirically validated [19, 23, 24].

In a previous paper [37], we suggested that, given the role that provenance plays in trust [16, 17], *argumentation* — which tracks the origin of data used in reasoning — might play a role. We have developed a graph-based model to explore the relationship between argumentation and trust [47]. Here we explore a different direction, discussing how the usual approach to dealing with trust information can be captured in logic, how it can be integrated with argumentation-based reasoning about beliefs, and how it might be used in a combined system.

2 Trust

We are interested in a finite set of agents Ags and how these agents trust one another. Following the usual presentation (for example [23, 44, 49]), we start with a *trust relation*:

$$\tau \subseteq Ags \times Ags$$

which identifies which agents trust one another. If $\tau(Ag_i, Ag_j)$, where $Ag_i, Ag_j \in Ags$, then Ag_i trusts Ag_j. This is not a symmetric relation, so it is not necessarily the case that $\tau(Ag_i, Ag_j) \Rightarrow \tau(Ag_j, Ag_i)$.

It is natural to represent this trust relation as a directed graph, and we define a *trust network* to be a graph comprising, respectively, a set of nodes and a set of edges:

$$\mathcal{T} = \langle Ags, \{\tau\} \rangle$$

where Ags is a set of agents and $\{\tau\}$ is the set of pairwise trust relations over Ags so that if $\tau(Ag_i, Ag_j)$ is in $\{\tau\}$ then $\{Ag_i, Ag_j\}$ is a directed arc from Ag_i to Ag_j in $\mathcal{T}$ indicating that Ag_i trusts Ag_j.

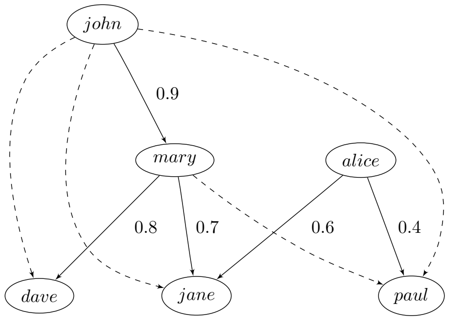


Fig. 1. An example trust graph. The solid lines represent direct trust relations, and the dashed lines represent derived trust. The link between $john$ and $jane$ and the link between $john$ and $dave$ are the result of direct propagation. The link between $mary$ and $paul$ is the result of co-citation (see below).

In this graph, the set of agents is the set of vertices, and the trust relations define the arcs. A directed path between agents in the trust network implies that one agent indirectly trusts another. For example if:

$$\langle Ag_1, Ag_2, \dots Ag_n \rangle$$

is a path from agent Ag_1 to Ag_n, then we have:

$$\tau(Ag_1, Ag_2), \tau(Ag_2, Ag_3), \dots, \tau(Ag_{n-1}, Ag_n)$$

and the path gives us a means to compute the trust that Ag_1 has in Ag_n. The usual assumption in the literature is that we can place some measure on the trust relation, quantifying the trust that one agent has in another, so we have:

$$tr : Ags \times Ags \mapsto \Re$$

where tr gives a suitable trust value. In this paper, we take this value to be between 0, indicating no trust, and 1, indicating the greatest possible degree of trust. We assume that tr and τ are mutually consistent, so that:

$$\begin{aligned} tr(Ag_i, Ag_j) \neq 0 \Leftrightarrow (Ag_i, Ag_j) \in \tau \\ tr(Ag_i, Ag_j) = 0 \Leftrightarrow (Ag_i, Ag_j) \notin \tau \end{aligned}$$

Now, this just deals with the direct trust relations encoded in τ. It is usual in work on trust to consider performing inference about trust by assuming that trust relations are transitive. This is easily captured in the notion of a trust network. The notion of trust embodied here is exactly Jøsang's "indirect trust" or "derived trust" [21] and the process of inference is what [19] calls "direct propagation". If we have a function tr, then we can compute:

$$\begin{aligned} & tr(Ag_i, Ag_j) = \\ & \quad tr(Ag_i, Ag_{i+1}) \otimes^{tr} tr(Ag_{i+1}, Ag_{i+2}) \otimes^{tr} \dots \otimes^{tr} tr(Ag_{j-1}, Ag_j) \end{aligned} \tag{1}$$

for some operation $\otimes^{tr}$. Here we follow [49] in using the symbol $\otimes$, to stand for this generic operation.[1] The superscript distinguishes this from a similar operation $\otimes^{bel}$ on belief values which we will meet below.

Sometimes it is the case that there are two or more paths through the trust network between Ag_i and Ag_j indicating that Ag_i has several opinions about the trustworthiness of Ag_j. If these two paths are

$$\langle Ag_i, Ag'_{i+1}, \ldots Ag_j \rangle \quad \text{and} \quad \langle Ag_i, Ag''_{i+1}, \ldots Ag_j \rangle$$

and

$$tr(Ag_i, Ag_j)' = tr(Ag_i, Ag'_{i+1}) \otimes^{tr} \ldots \otimes^{tr} tr(Ag'_{j-1}, Ag_j)$$
$$tr(Ag_i, Ag_j)'' = tr(Ag_i, Ag''_{i+1}) \otimes^{tr} \ldots \otimes^{tr} tr(Ag''_{j-1}, Ag_j)$$

then the overall degree of trust that Ag_i has in Ag_j is:

$$tr(Ag_i, Ag_j) = tr(Ag_i, Ag_j)' \oplus^{tr} tr(Ag_i, Ag_j)'' \tag{2}$$

Again we use the standard notation $\oplus$ for a function that combines trust measures along two paths [49]. Clearly we can extend this to handle the combination of more than two paths.

As an example of a trust graph, consider Figure 1 which shows the trust relationship between $john$, $mary$, $alice$, $jane$, $paul$ and $dave$. This is adapted from the example in [23] by normalizing the values to lie between 0 and 1 and adding $paul$. The solid lines are direct trust relationships and the dotted lines are indirect links derived from the direct links. Thus, for example, $john$ trusts $jane$ and $dave$ because he trusts $mary$ and $mary$ trusts $jane$ and $dave$.

The standard approach in the literature on trust is to base the computation of derived trust values on the the trust graph, for example using a path algebra [44]. Our aim in this paper is to demonstrate how we might use logic, and in particular argumentation, to propagate trust values. In other words we want an argumentation-based approach that $john$ can use to determine that he has a reason to trust $dave$, and then use to combine this trust with his other knowledge to make decisions.

3 Reasoning about trust

We will start by considering how to capture reasoning about trust in logic. We will assume that every agent Ag_i has some collection of information about the world, which we will call Δ_i, that is expressed in logic. Δ_i is made up of a number of partitions, one of which, Δ_i^{tr}, holds information about the degree of trust Ag_i has in other agents it knows. For example, the agent $john$ from the above example might have the following collection of information:

$$\begin{array}{ll} \Delta^{tr}_{john} & (t1 : trusts(john, mary) : 0.9) \\ & (t2 : trusts(mary, jane) : 0.7) \\ & (t3 : trusts(mary, dave) : 0.8) \\ & (t4 : trusts(alice, jane) : 0.6) \\ & (t5 : trusts(alice, paul) : 0.4) \end{array}$$

[1] [19, 23, 44, 49], among others, provide different possible instantiations of this operation.

$$Ax^{tr} \quad \frac{(n : trusts(x,y) : \tilde{d}) \in \Delta_i^{tr}}{\Delta_i^{tr} \vdash_{tr} (trusts(x,y) : \{n\} : \{Ax^{tr}\} : \tilde{d})}$$

$$dp \quad \frac{\Delta_i^{tr} \vdash_{tr} (trusts(x,y) : G : R : \tilde{d}) \text{ and } \Delta_i^{tr} \vdash_{tr} (trusts(y,z) : H : S : \tilde{e})}{\Delta_i^{tr} \vdash_{tr} (trusts(x,z) : G \cup H : R \cup S \cup \{dp\} : \tilde{d} \otimes^{tr} \tilde{e})}$$

$$cc \quad \frac{\Delta_i^{tr} \vdash_{tr} (trusts(x,y) : G : R : \tilde{d}) \text{ and } \Delta_i^{tr} \vdash_{tr} (trusts(x,z) : H : S : \tilde{e}) \text{ and } \Delta_i^{tr} \vdash_{tr} (trusts(w,z) : K : T : \tilde{f})}{\Delta_i^{tr} \vdash_{tr} (trusts(w,y) : G \cup H \cup K : R \cup S \cup T \cup \{cc\} : \tilde{d} \otimes^{tr} \tilde{e} \otimes^{tr} \tilde{f})}$$

Fig. 2. Part of the tr consequence relation

where the elements of $\mathit{\Delta}^{tr}_{john}$ are the kind of triples that we have discussed in earlier work [35]. Each element has the form:

$$(\langle index \rangle : \langle data \rangle : \langle value \rangle)$$

The first is a means of referring to the element, the second is a formula, and here the third is the degree of trust between the individuals mentioned in the $trust$ relation.

From $\mathit{\Delta}^{tr}_{john}$ we can then construct arguments mirroring the trust propagation discussed above. Rules for doing this are given in Figure 2.[2] For example, using the first two rules, from Figure 2, Ax^{tr} and dp, we can construct the argument:

$$\mathit{\Delta}^{tr}_{john} \vdash_{tr} (trusts(john, jane) : \{t1, t2\} : \{Ax^{tr}, Ax^{tr}, dp\} : \tilde{t})$$

where all arguments in our approach take the form:

$$(\langle conclusion \rangle : \langle grounds \rangle : \langle rules \rangle : \langle value \rangle)$$

The $\langle conclusion \rangle$ is inferred from the $\langle grounds \rangle$ using the rules of inference $\langle rules \rangle$ and with degree $\langle value \rangle$. In this case the argument says $john$ trusts $jane$ with degree $\tilde{t}$ (which is $0.9 \otimes^{tr} 0.7$), through two applications of the rule Ax^{tr} and one application of the rule dp to the two facts indexed by $t1$ and $t2$.[3]

The rule Ax^{tr} says that if some agent Ag_i has a triple:

$$(t1 : trusts(john, mary) : 0.9)$$

in its $\mathit{\Delta}^{tr}_{i}$ then it can construct an argument for $trusts(john, mary)$ where the grounds are $t1$, the degree of trust is 0.9, and which records that the Ax^{tr} rule was used in its derivation.

The rule dp captures direct propagation of trust values. It says that if we can show that $trusts(x, y)$ holds with degree $\tilde{d}$ and we can show that $trusts(y, z)$ holds with degree $\tilde{e}$, then we are allowed to conclude $trusts(x, z)$ with a degree $\tilde{d} \otimes^{tr} \tilde{e}$, and that the conclusion is based on the union of the information that supported the premises, and is computed using all the rules used by both the premises.

Why is this interesting? After all, it does no more than trace paths through the trust graph.

Well, one of the strengths of argumentation, and the reason we are interested in using argumentation to handle trust, is that we want to record, in the form of the argument for some proposition, the *reasons* that it should be believed. Since information on the source of some piece of data, and the trust that an agent has in the source, is relevant, then it should be recorded in the argument. This is easier to achieve if we encode data about who trusts whom in logic.

[2] Note that the consequence relation in Figure 2 is not intended to be comprehensive. There are many other ways to construct arguments about trust — for some examples see [36] — which could be included in the definition of $\vdash_{tr}$.

[3] There are good reasons for using the formulae themselves in the grounds and factoring the whole proof into the set of rules (as we do in [37]) to obtain structured arguments like those in [15, 41]. However, for simplicity, here we use the relevant indices.

One of the nice things that this approach allows us to do is to track the application of the rules for propagating trust. When we just use direct propagation, this is not terribly interesting (though it does allow us to distinguish between the bits of information used in the formation of arguments, which may be a criterion for preferring one argument over another [28]), but it becomes more obviously useful when we start to allow other rules for propagating trust. For example, [19] suggests a rule the authors call *co-citation*, which they describe as:

> For example, suppose i_1 trusts j_1 and j_2 and i_2 trusts j_2. Under co-citation, we would conclude that i_2 should also trust j_1.

In our example (see Figure 1), therefore, co-citation suggests that since $alice$ trusts $jane$ and $paul$, and $mary$ trusts $jane$, then $mary$ should trust $paul$. (Presumably the idea is that since $alice$ and $mary$ agree on the trustworthiness of $jane$, $mary$ should trust $alice$'s opinion about $paul$). [19] also tells us how trust values should be combined in this case — $mary$'s trust in $paul$ is just the combination of trust values along the path from $mary$ to $jane$ to $alice$ to $paul$.

This form of reasoning is captured by the rule cc in Figure 2, and the rule also takes care of the necessary bookkeeping of grounds, proof rules and trust values. Combining the application of cc with dp as before allows the construction of the argument:

$$\mathit{\Delta}^{tr}_{john} \vdash_{tr} (trusts(john, paul) : \{t1, t2, t4, t5\} : rules_1 : \tilde{r})$$

indicating that $john$ trusts $paul$, where $rules_1$ is:

$$\{Ax^{tr}, Ax^{tr}, Ax^{tr}, Ax^{tr}, cc, dp\}$$

and $\tilde{r}$ is $0.9 \otimes^{tr} 0.7 \otimes^{tr} 0.6 \otimes^{tr} 0.4$.

Now, when we have several rules for propagating trust, keeping track of which rule has been used in which derivation is appealing, especially since one might want to distinguish between arguments that use different rules of inference. For example, one might prefer arguments, no matter the trust value, which only make use of direct propagation over those that make use of co-citation.[4]

4 Reasoning with trust

What we have presented so far explains how agent Ag_i can reason about the trustworthiness of its acquaintances. The reason for doing this is so Ag_i can use its trust information to decide how to use information that it gets from those acquaintances. To formalize the way in which Ag_i does this, we will assume that, in addition to $\mathit{\Delta}^{tr}_i$, Ag_i has a set of beliefs about the world $\mathit{\Delta}^{bel}_i$ (which we assume come with some measure of belief), and some information $\mathit{\Delta}^j_i$ provided by each of its acquaintances Ag_j, and that:

$$\mathit{\Delta}_i = \mathit{\Delta}^{tr}_i \cup \mathit{\Delta}^{bel}_i \cup \bigcup_j \mathit{\Delta}^j_i$$

[4] Though [19] shows that propagation based on co-citation matches empirical results for the way people propagate trust, our experience is that people also often find the notion of co-citation somewhat unconvincing when they are first exposed to it.

$$Ax^{bel} \quad \frac{(n : \theta : \tilde{d}) \in \Delta_i^{bel}}{\Delta_i \vdash_{bel} (\theta : G : \{Ax^{bel}\} : \tilde{d})}$$

$$\text{Trust} \quad \frac{\Delta_i^{tr} \vdash_{tr} (trusts(i,j) : G : R : \tilde{d}) \;\text{ and }\; \Delta_i^{j} \vdash_{bel} (\theta : H : S : \tilde{e})}{\Delta_i \vdash_{bel} (\theta : G \cup H : R \cup S \cup \{Trust\} : ttb(\tilde{d}) \otimes^{bel} \tilde{e})}$$

$$\wedge\text{-I} \quad \frac{\Delta_i \vdash_{bel} (\theta : G : R : \tilde{d}) \;\text{ and }\; \Delta_i \vdash_{bel} (\phi : H : S : \tilde{e})}{\Delta_i \vdash_{bel} (\theta \wedge \phi : G \cup H : R \cup S \cup \{\wedge\text{-I}\} : \tilde{d} \otimes^{bel} \tilde{e})}$$

$$\rightarrow\text{-E} \quad \frac{\Delta_i \vdash_{bel} (\theta : G : R : \tilde{d}) \;\text{ and }\; \Delta_i \vdash_{bel} (\theta \rightarrow \phi : H : S : \tilde{e})}{\Delta_i \vdash_{bel} (\phi : G \cup H : R \cup S \cup \{\rightarrow\text{-E}\}) : \tilde{d} \otimes^{bel} \tilde{e})}$$

Fig. 3. Part of the bel consequence relation

All of this information can then be used, along with the consequence relation from Figure 3, to construct arguments that combine trust and beliefs.

The proof rules in Figure 3 are based on those we introduced in [30]. The rule Ax^{bel}, as in the previous set of proof rules, bootstraps an argument from a single item of information, while the rules ∧-I and →-E are typical natural deduction rules — the rules for introducing a conjunction and eliminating implication — augmented with the combination of degrees of belief, and the collection of information on which data and proof rules have been used. (The full consequence relation would need an introduction rule and elimination rule for every connective in the language, and the definition of these is easy enough — we omit them here in the interest of space.)

The key rule in Figure 3 is the rule named Trust. This says that if it is possible to construct an argument for θ from some Δ_j^i, indicating that the information comes from Ag_j, and Ag_i trusts Ag_j, then Ag_i has an argument for θ. The grounds of this argument combine all the data that was used from Δ_j^i and all the information about trust used to determine that Ag_i trusts Ag_j, and the set of rules in the argument record all the inferences needed to build this combined argument. Finally, the belief that Ag_i has in the argument is the belief in θ as it was derived from Δ_j^i combined with the trust Ag_i has in Ag_j. We carry out this last combination by first turning the trust value into a belief value using some suitable function $ttb(\cdot)$.

In other words, this rule sanctions the use of information from an agent's acquaintances, provided that the degree of belief in that piece of information is modified by the agent's trust in that acquaintance. Thus one agent can only import information from another agent if the first agent can construct a trust argument that determines it should trust the second (and so trigger the Trust rule).

5 Example

To see how this combined system might work, consider the rest of the example from [23] that goes with Figure 1 (suitably modified to provide an example of co-citation, which is not considered in the original). The trust network from [23] is based on data from the FilmTrust site[5] which features social networks centered around the exchange of information about films.

In the example, $john$ has the following information, where x is a universally quantified variable, $almodovar$ is the director Pedro Almodovar, and hce is an abbreviation for the 2002 film *Hable con ella* (Talk to her):

$$\begin{array}{ll} \Delta^{bel}_{john} & (j1 : SpanFilm(hce) : 1) \\ & (j2 : DirBy(almodovar, hce) : 1) \\ & (j3 : Comedy(x) \rightarrow \neg Watch(x) : 0.8) \end{array}$$

We take this to mean that $john$ thinks that hce is a Spanish language film, and that it is directed by Almodovar. In addition, he doesn't much like to watch comedies. $john$ also has some information from FilmTrust connections:

$$\Delta^{mary}_{john} \ (jm1 : IndFilm(hce) : 1)$$

$$\Delta^{jane}_{john} \ (jj1 : IndFilm(x) \wedge SpanFilm(x) \rightarrow \neg Watch(x) : 1)$$

$$\Delta^{dave}_{john} \ (jd1 : DirBy(x, almodovar) \rightarrow Watch(x) : 1)$$

$$\Delta^{paul}_{john} \ (jp1 : Comedy(hce) : 0.6)$$

Thus $john$ hears from $mary$ that hce is an independent film, from $jane$ that her advice is to not watch Spanish independent films, from $dave$ who says any of Almodovar's films are worth seeing, and from $paul$ who points out that he thinks hce is a comedy.

Now, we have already seen how $john$ can construct arguments for trusting $jane$ and $paul$, though we did not say what $\otimes^{tr}$was so that we could not compute the degrees of trust. For now, we follow [44] in taking $\otimes^{tr}$to be minimum, thus giving us:

$$\Delta^{tr}_{john} \vdash_{tr} (trusts(john, jane) : \{t1, t2\} : \{Ax^{tr}, Ax^{tr}, dp\} : 0.7)$$

and

$$\Delta^{tr}_{john} \vdash_{tr} (trusts(john, paul) : \{t1, t2, t4, t5\} : rules_1 : 0.4)$$

$john$ can also infer:

$$\Delta^{tr}_{john} \vdash_{tr} (trusts(john, dave) : \{t1, t3\} : \{Ax^{tr}, Ax^{tr}, dp\} : 0.7)$$

in exactly the same way as he infers trust about $jane$. He can also construct the following argument for trusting $mary$:

$$\Delta^{tr}_{mary} \vdash_{tr} (trusts(john, mary) : \{t1\} : \{Ax^{tr}\} : 0.9)$$

Each of the arguments can then be used with $\vdash_{bel}$ (Figure 3) to construct arguments that are relevant to the question of whether $john$ should watch hce. Using information from $jane$ he can determine:

$$\Delta_{john} \vdash_{bel} (\neg Watch(hce) : \{t1, t2, jj1, jm1, j1\} : rules_2 : \tilde{b})$$

[5] http://trust.mindswap.org/FilmTrust/

where

$$rules_2 = \{Ax^{tr}, Ax^{tr}, dp, Trust, Trust, Ax^{bel}, \wedge\text{-I}, \rightarrow\text{-E}\}$$

This shows that after the derivation of information about trusting $jane$, the proof of $\neg Watch(hce)$ requires the application of $Trust$ to establish a degree of belief in $jane$'s information, $Trust$ to import $jm1$ from $mary$, an application of Ax^{bel} to create an argument from $j1$, the use of $\wedge$-I to combine the data from $j1$ and $jm1$, and then $\rightarrow$-E to get the conclusion.

To establish $\tilde{b}$, we need to determine what the function $\otimes^{bel}$ is, and how to convert trust values to beliefs using $ttb(\cdot)$. For our purposes here, the choice doesn't matter greatly — we aren't arguing that any particular combination of operations for trust combination, belief combination and $ttb(\cdot)$ is best, just that if we have these operations then $john$ can use information in a way that seems to be useful. For now we handle beliefs using possibility theory [5] — which is basically equivalent to the approach adopted by [3] to handle variable strength arguments — and interpret the degree of trust in an agent to be a degree of belief that what the agent says is true [14, 32], so that $ttb(\cdot)$ is just the identity. All of this means that $\tilde{b} = 0.7$.

$john$ can also construct the following arguments as a result of information from, respectively, $paul$ and $dave$, in much the same way as the argument above. First we have:

$$\varDelta_{john} \vdash_{bel} (\neg Watch(hce) : \{t1, t2, t4, t5, jp1, j3\} : rules_3 : 0.4)$$

where

$$rules_3 = \{Ax^{tr}, Ax^{tr}, Ax^{tr}, Ax^{tr}, dp, cc, Trust, Ax^{bel}, \rightarrow\text{-E}\}$$

and second we have:

$$\varDelta_{john} \vdash_{bel} (Watch(hce) : \{t1, t3, jd1, j1, j2\} : rules_4 : 0.6)$$

where

$$rules_4 = \{Ax^{tr}, Ax^{tr}, dp, Trust, Ax^{bel}, Ax^{bel}, \rightarrow\text{-E}\}$$

This means that $john$ has three arguments that bear on his decision about whether to watch hce, one in favor and two against.

6 Using trust values

At this point in the example, we have arguments for opposing conclusions — $john$ should watch hce and $john$ should not watch it. To reach a decision about hce, $john$ needs to choose between these conclusions. There are a number of different approaches to using the trust information to do this, and in this section we discuss some of them, showing how they affect the example. The aim here is not to provide a definitive answer but to explain some of the options — as we hope that these examples will demonstrate, it is not immediately clear which is the best approach.

6.1 Flattening

The first approach is for $john$ to proceed by combining the arguments for the formula $\neg Watch(hce)$ (what [35] calls "flattening" the arguments) and seeing if the resulting combination outweighs the argument for $Watch(hce)$. We have three arguments to consider:

$$\begin{array}{ll} A_1 & (\neg Watch(hce) : \{t1, t2, jj1, jm1, j1\} : rules_2 : 0.7) \\ A_2 & (\neg Watch(hce) : \{t1, t2, t4, t5, jp1, j3\} : rules_3 : 0.4) \\ A_3 & (Watch(hce) : \{t1, t3, jd1, j1, j2\} : rules_4 : 0.6) \end{array}$$

Flattening combines the two beliefs, 0.7 and 0.4 for $\neg Watch(hce)$, to get a combined measure. Given that we are taking the values to be possibility values, it makes sense to combine them using $\max$, thus getting a combined value of 0.7 for $\neg Watch(hce)$. This is greater than the 0.6 for $Watch(hce)$, and so under this scheme, $john$ would conclude that he should not watch hce.

Given the choice of combination operator for flattening, this approach is very simple — the choice supported by the strongest single argument will always win. It also largely ignores conflicts between the arguments. In the example so far, we just have arguments that rebut one another, and the result of flattening seems very reasonable. But what if we have more conflicts? Consider extending the example so that $john$ has additional information:

$$\begin{array}{ll} \Delta^{bel}_{john} & (j1 : SpanFilm(hce) : 1) \\ & (j2 : DirBy(almodovar, hce) : 1) \\ & (j3 : Comedy(x) \rightarrow \neg Watch(x) : 0.8) \\ & (j4 : DirBy(almodovar, x) \rightarrow \neg IndFilm(x) : 1) \end{array}$$

so $john$ is now certain that anything directed by Almodovar is not an independent film. This gives him an additional argument:

$$A_4 \qquad (\neg IndFilm(hce) : \{j2, j4\} : \{Ax^{bel}, Ax^{bel}, \rightarrow\text{-E}\} : 1)$$

Thus $john$ now has a strong argument against hce being an independent film, and this clearly conflicts with A_1 since it contradicts the information from $mary$ about hce being an independent film. A_4 however, is ignored by flattening, and this doesn't seem very reasonable.

6.2 Acceptability analysis

Of course, handling this kind of conflict is exactly what Dung's acceptability semantics [11] and subsequent variations on this theme [6, 12] are intended to do. Let's examine what they tell $john$ in this scenario. [11] starts from the position of knowing which arguments conflict, assuming a relation that specifies:

$$attacks(A_n, A_m)$$

for all conflicts between arguments. Since we are starting from a less abstract position, we need to define what constitutes this relation in our example. The notion of conflict

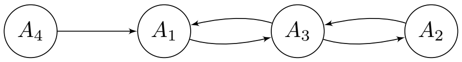


Fig. 4. The argumentation graph for the film example when the strengths of arguments are not taken into account.

between arguments used in [3] translates into our formulation of an argument as saying that $(c : G : R : v)$ attacks $(c' : G' : R' : v')$ if there is some $g \in G'$ such that $c \equiv \neg g$. That is one argument attacks another by disputing the truth of one of its grounds, "undercutting" it in the usual terminology.[6] ([3] also places some constraints on the strengths of the arguments v and v', but we will leave those for now.)

We will extend this notion of attack to include arguments rebutting each other, so that for our purposes $(c : G : R : v)$ attacks $(c' : G' : R' : v')$ if either $c \equiv \neg c'$ or there is some $g \in G'$ such that $c \equiv \neg g$. With this definition we have:

$attacks(A_1, A_3)$
$attacks(A_3, A_1)$
$attacks(A_2, A_3)$
$attacks(A_3, A_2)$
$attacks(A_4, A_1)$

and the argument graph is that of Figure 4. What $john$ concludes from this depends on the way that he computes which arguments are acceptable. However, none of the different approaches from [11] will help him decide what to watch. If he applies the grounded semantics, the only acceptable argument is A_4, which doesn't tell him what to watch. If he applies the complete, preferred or stable semantics, they will all tell him that A_4 is acceptable along with A_2 or A_3, but give no further guidance. As a result, while in other scenarios this analysis may suffice, in this case it leaves $john$ no wiser about whether he should watch hce or not.[7]

Since the basic acceptability analysis is not very informative, and since we have a degree of belief associated with each argument, we can incorporate the degrees of belief into the analysis. To do this, we extend our notion of $attack$ with the mechanism that [3] uses to handle strength of arguments. Broadly speaking (and counting rebutting as well as undercutting arguments), what [3] says is that $(c : G : R : v)$ attacks $(c' : G' : R' : v')$ if either $c \equiv \neg c'$ or there is some $g \in G'$ such that $c \equiv \neg g$, and $v \geq v'$. Thus if an argument has a conflict with a strictly stronger argument, that conflict is ignored in establishing the $attacks$ relation. With this definition we have:

[6] The term "undercutting" was originally used by Pollock, for example in [40], to refer to the situation in which one argument attacked an inference in another, but in the computer science community the term was rapidly co-opted to mean the kind of attack we describe here [3, 7, 42].

[7] The grounded semantics can't untangle the rebutting conflict between A_2 and A_3, while the other semantics tell $john$ that the rebutting means one of the arguments is acceptable, but they can't make a choice between the arguments. All the semantics determine that A_4 makes A_1 unacceptable, and hence unable to have any effect on the conflict between A_2 and A_3.

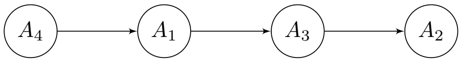


Fig. 5. The argumentation graph for the film example when the strengths of arguments are taken into account.

$attacks(A_1, A_3)$
$attacks(A_3, A_2)$
$attacks(A_4, A_1)$

and the argument graph is that of Figure 5. This time, any of the standard semantics from [11] tells $john$ that the acceptable arguments are A_3 and A_4, and so his conclusion using this approach is that he should watch hce.

The approaches we have discussed up to now are direct applications of existing approaches to using arguments with some form of belief value, and only use the trust information as a mechanism to establish arguments about beliefs. Our investigation is also considering three other approaches, in which we use the trust value directly. We will discuss these next.

6.3 Trust thresholds

The first of these new approaches is the use of *trust thresholds*. The formal model we are using here considers an agent to have information from a number of acquaintances, each of which has some trust rating that is applied to the information from that agent. A natural approach to using the trust rating is to specify a threshold value below which information from an agent is disregarded.

In the case of our film example, $john$ might set his trust threshold to 0.5, thus not accepting information from any acquaintance y for which he cannot infer:

$$(trusts(john, y) : G : R : v)$$

for some $v > 0.5$. (One might formulate this as an additional condition in the Trust rule in the $\vdash_{bel}$ relation.) Doing this would rule out any information from $paul$, and hence $john$ would only have A_1, A_3 and A_4. Of course, using the threshold doesn't answer $john$'s question on its own — he still has arguments for and against watching hce, so he will have to use a method like those outlined above to resolve the conflict. If, for example, $john$ chooses to use the acceptability semantics without considering the strengths of the arguments, this time he will find that all the standard semantics say that A_3 is acceptable and so he should $watch(hce)$. (The outcome of the two other approaches are not affected by the threshold, but it does mean that there are fewer arguments to consider.)

A number of questions arise about the use of thresholds. To what extent, for example, does imposing such a threshold on the information from its acquaintances protect an agent from using untrustworthy information? In other words, does excluding information from acquaintances with a trust value below some α mean that all of the agent's conclusions will be more trustworthy than α? Or are there circumstances under which less trustworthy conclusions could be reached even if data from agents below the threshold is excluded? We have shown that under some circumstances the trust threshold will give us this protection [38], but in case of our example, it won't. Imagine that the threshold is set to 0.65, ruling out data from any agent except $mary$ and $jane$, so $john$ has just A_4 and A_1 (and so no opinion about whether to watch hce because the only attack is that of A_4 on A_1 which makes A_1 unacceptable). Can this be altered by information below the threshold, say from $mary$, who is highly trusted, but maybe has some low belief information about the watchability of hce? It might. If $mary$ has information that leads to an argument A_5 with conclusion $watch(hce)$ and a belief of 0.5 say, it won't be excluded by the threshold (which only applies to $mary$ not to data from $mary$), and A_5 will be acceptable (because the attacking argument A_1 is itself attacked by A_4), giving the conclusion $watch(hce)$. Our current work is trying to establish what are reasonable levels of protection that may be provided by trust thresholds, and for which combinations of interpretations for trust and belief values the levels of protection hold.

Now, given an arbitrary threshold, there may be no arguments for or against watching hce for which the grounds are all above the threshold — meaning that $john$ has no arguments to consider — but many arguments with elements of their grounds just below the threshold — meaning that $john$ would consider them if the threshold was lower. For such cases $john$ might want to consider altering the threshold, and so we are interested in how the protection offered by the threshold is altered when the threshold moves.

Another interesting question is to examine the interaction between thresholds and propagation in the trust network. What correspondence is there between imposing a trust threshold and pruning the acquaintances from the network? Clearly when we combine trust values along a path through the network using $\min$, a threshold will rule out trusting any agent downstream of an agent below the threshold, but this may not necessarily be the case when trust values are computed in different ways. Again, this is a matter that we are currently investigating.

6.4 Trust budget

The second new approach is, in some ways, an extension of the first. Using a trust threshold rules out acquaintances — or alternatively conclusions that are supported by information from those acquaintances — when the level of trust in an acquaintance drops below a particular level. Thus very untrustworthy acquaintances, and the information they provide, are ruled out. But equally, information from sources above the threshold is ruled in, along with conclusions based on it, even if a given conclusion depends upon lots of items of information that came from sources close to the threshold, and so might be considered more suspect than others based on sources further from the threshold.

The notion of a *trust budget* is intended to deal with this situation. A trust budget specifies the total amount of distrust that is permitted in the sources of data that lead to a single conclusion. In situations where trust values are, as in our example, between

0 and 1, we can compute the "cost" of Ag_i accepting information from a series of acquaintances Ag_j as:

$$\sum_j 1 - tr(i, j)$$

To illustrate this idea on the example, let us first imagine that $john$ sets the trust budget to 1. Given the levels of trust that $john$ has in his acquaintances, this allows him to accept information from at most any three of $jane$ (cost to the trust budget of 0.3), $paul$ (0.6), $dave$ (0.3) and $mary$ (0.1). For example, $john$ might spend the whole trust budget and accept information from $jane$, $paul$ and $mary$, giving him the conclusion that he should not watch hce. Or he might spend part of the budget accepting information from $jane$, $dave$ and $mary$, from which he would conclude that he should watch hce.

Given a specific budget, $john$ can identify which conclusion or conclusions that fit within the budget have the highest belief (here it is $\neg watch(hce)$). Alternatively $john$ might consider slowly increasing his trust budget from 0 until he reaches a conclusion about the question he is interested in — here he would have to "spend" at least 0.3 to get a conclusion (in this case to not watch hce, based on A_1 obtained by accepting information from $jane$). Another approach to using the trust budget would be to have $john$ establish what he needs to "spend" in order to find a conclusion he wants. In the context of the example, let's imagine he is interested in watching hce but wants to know how trusting he has to be to decide that it is a good idea. The minimum budget necessary to establish $watch(hce)$ as a conclusion is 0.6, the cost of trusting $paul$, since it only takes information from $paul$ to construct an argument for $watch(hce)$ (in more complex examples it might be necessary to trust several agents to reach an interesting conclusion).[8]

In general, the questions to ask about a trust budget are similar to those for a trust threshold, identifying how well-behaved this notion is, and what protection an agent gets by imposing such a budget. These questions are, like those for trust thresholds, subjects of our ongoing research. Furthermore, as suggested by [13], in the context of the related notion of an "inconsistency budget", and [26], in the context of optimal trust path selection, the kinds of uses we are seeking to make of the trust budget are uses that will require considerable computation. This is another topic we are considering.

6.5 Meta-argumentation

The previous two approaches are concerned with handling the values derived from the trust network. These values are then used to make decisions about which piece of information, and thus which arguments (since arguments are derived from the information) are considered by an agent. The final approach we are looking at leans more towards the kind of structural analysis described by Loui [28], where heuristic patterns of evidence and argument structure are used to decide which arguments are preferred. An example is the preference for arguments using only data from agents that are directly trusted by Ag_i over arguments that use data from agents that Ag_i trusts by co-citation. The aim of

[8] We are mainly interested in incorporating trust into planning, where the concept of establishing how much trust it "costs" to build an argument (plan) makes more sense than in the domain of the example.

this approach is to identify general heuristics for dealing with trust data, and to verify the plausibility, or otherwise, of the kinds of inference that they sanction.

7 Summary

In this paper, we have outlined work on reasoning about trust using a form of argumentation which, as the paper demonstrated, can be integrated with a system of argumentation that uses the conclusions about trust. A notable feature of the system for reasoning about trust is its flexibility — new approaches to propagating trust can easily be added (or, indeed, removed) by altering the proof rules that are used in propagation. The combined system was illustrated with an example, and current directions sketched.

Clearly the systems we have described are work in progress. Neither of the formal systems is complete as presented — both are missing much of the proof mechanism and a proper description of the syntax at the very least — and neither is rigorously evaluated. Our aim was simply to illustrate the basic ideas captured in the systems, and to illustrate the possibilities that they offer. We have also completely ignored the computational aspects of implementing a software system that employs these approaches. Our future work will, in due course, fill in the details that are missing here, more completely relate this work to approaches with similar aims, such as [29, 33], and provide an implementation. However, we believe that the work we have presented here has value in describing an area of research that we think is interesting and identifying some new approaches to handling it.

Acknowledgements

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

We would like to thank the reviewers of this paper for their many interesting ideas, most of which we did not, thanks to the short time between notification and the deadline for completing revisions, have time to incorporate. We will strive to do so in future work. We would also like to thank a reviewer of an earlier paper on trust who suggested that we extend the film example with the notion that Almodovar is not really an independent film director. That reviewer was responsible for all the structural interest in the example.

References

1. Z. Abrams, R. McGrew, and S. Plotkin. Keeping peers honest in EigenTrust. In *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.

2. B. T. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th International World Wide Web Conference*, Banff, Alberta, May 2007.
3. L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artifical Intelligence*, 34(3):197–215, 2002.
4. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics*, 5(2):58–71, June 2007.
5. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, 1992.
6. M. W. A. Caminada. Semi-stable semantics. In *Proceedings of the 1st International Conference on Computational Models of Argument*, Liverpool, UK, September 2006.
7. C. I. Chesñevar, A. G. Maguitman, and R. P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
8. P. Dandekar, A. Goel, R. Govindan, and I. Post. Liquidity in credit networks: A little trust goes a long way. Technical report, Department of Management Science and Engineering, Stanford University, 2010.
9. D. B. DeFigueiredo and E. T. Barr. TrustDavis: A non-exploitable online reputation system. In *Proceedings of the 7th IEEE International Conference on E-Commerce Technology*, 2005.
10. X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. In *Proceedings of the 35th International Conference on Very Large Databases*, Lyon, France, August 2009.
11. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
12. P. M. Dung, P. Mancarella, and F. Toni. A dialectical procedure for sceptical, assumption-based argumentation. In *Proceedings of the 1st International Conference on Computational Models of Argument*, Liverpool, UK, September 2006.
13. P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, (in press).
14. D. Gambetta. Can we trust them? In D. Gambetta, editor, *Trust: Making and breaking cooperative relations*, pages 213–238. Blackwell, Oxford, UK, 1990.
15. A. J. García and G. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
16. F. Geerts, A. Kementsiedtsidis, and D. Milano. Mondrian: Annotating and querying databases through colors and blocks. In *Proceedings of the 22nd International Conference on Data Engineering*, Atlanta, April 2006.
17. J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In *Proceedings of the International Provenance and Annotation Workshop*, Chicago, Illinois, May 2006.
18. T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 4(4):2–16, 2000.
19. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on the World Wide Web*, 2004.
20. C-W Hang, Y. Wang, and M. P. Singh. An adaptive probabilistic trust model and its evaluation. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-agent Systems*, Estoril, Portugal, 2008.
21. A. Jøsang, C. Keser, and T. Dimitrakos. Can we manage trust? In *Proceedings of the 3rd International Conference on Trust Management*, Paris, May 2005.

22. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th World Wide Web Conference*, May 2004.
23. Y. Katz and J. Golbeck. Social network-based trust in prioritzed default logic. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.
24. Y. Kuter and J. Golbeck. SUNNY: A new algorithm for trust inference in social networks using probabilistic confidence models. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007.
25. J. Lang, M. Spear, and S. F. Wu. Social manipulation of online recommender systems. In *Proceedings of the 2nd International Conference on Social Informatics*, Laxenburg, Austria, 2010.
26. G. Li, Y. Wang, and M. A. Orgun. Optimal social trust path selection in complex social networks. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, GA., 2010.
27. L. Li and Y. Wang. Subjective trust inference in composite services. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, GA., 2010.
28. R. P. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence*, 3(3):100–106, 1987.
29. P-A. Matt, M. Morge, and F. Toni. Combining statistics and arguments to compute trust. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagents Systems*, Toronto, Canada, May 2010.
30. P. McBurney and S. Parsons. Tenacious tortoises: A formalism for argument over rules of inference. In *Proceedings of the ECAI Workshop on Computational Dialectics*, Berlin, 2000.
31. L. Mui, M. Moteashemi, and A. Halberstadt. A computational model of trust and reputation. In *Proceedings of the 35th Hawai'i International Conference on System Sciences*, 2002.
32. D. Olmedilla, O. Rana, B. Matthews, and W. Nejdl. Security and trust issues in semantic grids. In *Proceedings of the Dagstuhl Seminar, Semantic Grid: The converegance of technologies*, volume 05271, 2005.
33. N. Oren, T. Norman, and A. Preece. Subjective logic and arguing with evidence. *Artificial Intelligence*, 171(10–15):838–854, 2007.
34. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford InfoLab, 1999.
35. S. Parsons. On precise and correct qualitative probabilistic reasoning. *International Journal of Approximate Reasoning*, 35:111–135, 2004.
36. S. Parsons, K. Haigh, K. Levitt, J. Rowe, M. Singh, and E. Sklar. Arguments about trust. Technical report, Collaborative Technology Alliance, 2011.
37. S. Parsons, P. McBurney, and E. Sklar. Reasoning about trust using argumentation: A position paper. In *Proceedings of the Workshop on Argumentation in Multiagent Systems*, Toronto, Canada, May 2010.
38. S. Parsons, Y. Tang, E. Sklar, P. McBurney, and K. Cai. Argumentation-based reasoning in agents with varying degrees of trust. In *Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems*, Taipei, Taiwan, 2011.
39. J. Pasternak and D. Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China, 2010.
40. J. Pollock. *Cognitive Carpentry*. MIT Press, Cambridge, MA, 1995.
41. H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
42. H. Prakken and G. Sartor. Argument-based logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 1997.

43. P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. In M. R. Baye, editor, *The Economics of the Internet and E-Commerce*, pages 127–157. Elsevier Science, Amsterdam, 2002.
44. M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the 2nd International Semantic Web Conference*, 2003.
45. J. Sabater and C. Sierra. Review on computational trust and reputation models. *AI Review*, 23(1):33–60, September 2005.
46. A. Salehi-Abari and T. White. Trust models and con-man agents: From mathematical to empirical analysis. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, 2010.
47. Y. Tang, K. Cai, E. Sklar, P. McBurney, and S. Parsons. A system of argumentation for reasoning about trust. In *Proceedings of the 8th European Workshop on Multi-Agent Systems*, Paris, France, December 2010.
48. W. T. L. Teacy, G. Chalkiadakis, A. Rogers, and N. R. Jennings. Sequential decision making with untrustworthy service providers. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, 2008.
49. Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA, 2006.
50. X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the Conference on Knowledge and Data Discovery*, 2007.
51. B. Yu and M. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–349, 2002.
52. S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003.

Position Statement: Arguments Cooperatively Built by Groups of Agents

Samy Sá[1]
Computer Science Department

Universidade Federal do Ceará
samy@ufc.br

Abstract. Most work done in argumentation theory or argumentation based approaches only consider a single set of sentences. However, in a Multiagent System (MAS) setting, each agent can have a different perspective and proper (generally incomplete) knowledge base. As a consequence, it might be hard for a single agent to build an admissible position to argue with others about a given issue. We believe that, in such situations, a group of agents might be able to collaboratively build good arguments or, in any other situation, provide more complex admissible sets of arguments than a single agent. In this paper, we defend the collaborative construction of admissible positions by groups.

Keywords: Abductive Logic Programming, Argumentation, Joint Deliberation

1 Position

We consider an admissible position as a set of arguments that is internally consistent (conflict-free) and able to defend itself from all attackers (mutually defensive), such as in [5,1,6,8]. In some settings, however, an agent might not be able to build an admissible position on its own. On top of that, in MAS, agents usually have incomplete knowledge. We believe that, in such cases, a group of agents can benefit of deliberation in order to cooperatively *build* a group admissible position as a set of arguments provided by different agents. Some related work include a framework for agents to consider the conclusions of other agents as context to their own reasoning [2], to individually build inductive arguments for group learning [7] or persuasion [12] and for agents to *choose* a group position through judgment aggregation [3,9]. Alternative arguments for deliberation in face of incomplete information has been addressed in [6]. There is also work in argumentation based deliberation [8], but none, as far as our knowledge goes, focused in collaboratively *building* such positions in groups. In any case, we would like to motivate and encourage more work and discussion in that direction.

2 A Possibility

Abductive reasoning is usually taken as inference to the best explanation of a certain data. An abductive explanation is a set of conditions enough to make the observed data consistent to a theory. On its turn, logic programs have been used in a number of approaches to argumentation such as in [6,10] or even defended to be a form of argumentation [5]. Abductive logic programming (ALP) [4,11], brings the power of abductive reasoning to logic programs. Abductive explanations can be used to build *conditionally admissible arguments* as they provide a set of conditions enough for a program to prove a goal it could otherwise not.

We study the use of ALP as a mean for groups of agents to deliberate and build admissible positions. In this approach, the agents produce conditional arguments based on explanations to support an observation or point of view. Other agents propose rewrites of such arguments by adding rules and facts to satisfy some of the conditions, while possibly adding new ones (helping to build the argument), or criticizing a path of thought (attacking the argument as it is being built). A position admissible to the group is a conditional argument with zero conditions. For this purpose, we use the abductive framework presented in [11], as it also considers falsifying parts of a program to explain data. This difference adds expressibility to explanations and allows different kinds of arguments and attacks to arguments. Our higher goal is to enable group decision making to be taken as the result of discussion in a group of agents collaboratively seeking consensus. This is an ongoing research.

References

1. Leila Amgoud, , Leila Amgoud, Simon Parsons, and Nicolas Maudet. Arguments, dialogue, and negotiation. *JAIR*, 23:2005, 2000.
2. Gerhard Brewka and Thomas Eiter. Argumentation context systems: A framework for abstract group argumentation. In *LPNMR*, LNCS, pages 44–57. Springer, 2009.
3. Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.
4. Marc Denecker and Antonis C. Kakas. Abduction in logic programming. In *Computational Logic. Logic Programming and Beyond*, pages 402–436. Springer, 2002.
5. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
6. Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. AAMAS '03, pages 883–890. ACM, 2003.
7. Santiago Ontañón and Enric Plaza. Multiagent inductive learning: an argumentation-based approach. In *ICML*, pages 839–846. Omnipress, 2010.
8. Simon Parsons and Peter Mcburney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, page 2003, 2004.
9. Iyad Rahwan and Fernando Tohmé. Collective argument evaluation as judgement aggregation. In *AAMAS*, pages 417–424. IFAAMAS, 2010.
10. F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: An abductive approach. In *In Proceedings AISB'01 Convention*. AISB, 2001.
11. Chiaki Sakama and Katsumi Inoue. Negotiation by abduction and relaxation. In *AAMAS*, pages 242–249. IFAAMAS, 2007.
12. Maya Wardeh, Trevor J. M. Bench-Capon, and Frans Coenen. Multi-party argument from experience. In *ArgMAS*, LNCS, pages 216–235. Springer, 2009.

Toward the Application of Argumentation to Interactive Learning Systems

Elizabeth Sklar[1,2] and M. Q. Azhar[2]

[1] Dept of Computer & Information Science, Brooklyn College,
The City University of New York, 2900 Bedford Avenue, Brooklyn NY 11210, USA
[2] Dept of Computer Science, The Graduate Center,
The City University of New York, 365 Fifth Avenue, New York NY 10016, USA
sklar@sci.brooklyn.cuny.edu, mqazhar@gmail.com

Abstract. This paper explores the application of argumentation dialogues to an Interactive Learning System (ILS). The goal of an ILS is to provide an adaptive learning experience for a student within a particular domain, where the system adjusts dynamically as the student makes mistakes and learns from them. The system needs to be able to represent beliefs about the student's knowledge, and to update these beliefs as the student learns. The system also needs to have models of the domain and of an expert's actions within the domain, in order to compare and evaluate the student's actions. Finally, the system needs to provide appropriate feedback to the student, in such a way as to encourage learning. The work presented here describes a framework for such a system, built upon our earlier work on education dialogues.

1 Introduction

We explore the application of argumentation dialogues to an *Interactive Learning System (ILS)*. The goal of an ILS is to provide an adaptive learning experience for a student within a particular domain, where the system adjusts dynamically as the student makes mistakes and learns from them. The system needs to be able to represent beliefs about the student's knowledge, and to update these beliefs as the student learns. The system also needs to have models of the domain and of an expert's actions within the domain, in order to compare and evaluate the student's actions. Finally, the system needs to provide appropriate feedback to the student, in such a way as to encourage learning. The work presented here describes a framework for such a system.

Our model builds on earlier work in which we introduced the notion of an *education dialogue* [28]. Proposed for use in an interactive learning environment, an education dialogue is derived from previous work in the argumentation dialogue field [11, 20, 33]. Dialogues for education take place between two agents, each having specific roles: a *Tutor*, T, and a *Learner*, L. We focus on two types of interactions between these agents: $T \rightarrow L$ and $L \rightarrow T$, where the agent on the left side of the arrow initiates the dialogue, which is directed to the "target" agent on the right side of the arrow. Note that here we will not discuss $T \rightarrow T$

or $L \rightarrow L$ interactions, which, while possible in a general education dialogue, are not relevant for the specific instance discussed here.

Education dialogues are similar to *information seeking* dialogues [19, 33], but there are some key differences. When one agent asks another agent a question in an information seeking dialogue, the initiating agent does not know the answer and assumes that the target agent does. If the target agent does indeed know the answer, then she responds with the answer. However, in an education dialogue, there are reasons for the initiating agent to ask a question to which she already knows the answer and reasons for the target agent to not simply supply an answer she knows. Two such reasons are outlined below.

First, consider an education dialogue where the Tutor is the initiator, represented as $T \rightarrow L$. The Tutor actually does know the answer to the question she is posing. A good Tutor, pedagogically speaking, will ask a question that builds on the Learner's knowledge and coaxes him to learn; the answer will be something that the Tutor believes the Learner has the ability to find[3]. The Tutor is also refining her beliefs about the Learner's knowledge. Here, the Tutor is seeking information that is not the direct answer to the question, but rather she is seeking *meta-level knowledge* about the Learner—to see if the Learner knows the answer—instead of seeking the direct answer to her question (which, as stated, she already knows). Note that we make the assumption in the $T \rightarrow L$ interaction that the Learner will supply the answer if he knows it. There is a sizable literature from the educational psychology community on student motivation that explores reasons why a student might not answer a teacher's question correctly even if he knows the answer, but this avenue is outside the scope of the work discussed in this paper.

Second, consider an education dialogue where the Learner is the initiator, represented as $L \rightarrow T$. The Learner does not know the answer to the question she is posing (just like in a normal information seeking dialogue). If the Tutor knows the answer to the question, she may answer the question directly (as in an information seeking dialogue); or she may not provide the answer to the Learner, even though she knows it (unlike an information seeking dialogue). Since the Tutor's goal is to coax the learner to progress, she may decide to answer the Learner's question by providing more information about the answer, without providing the answer itself—to engage him in a thinking process that results in him learning.

The remainder of this paper is organized as follows. Section 2 discusses the specifics of education dialogues, reviewing some key components and introducing some new locutions. Section 3 briefly reviews the field of interactive learning systems, and focuses on highlighting components that are relevant to our framework. Section 4 describes our framework. Section 5 closes with a summary and discussion of future work.

[3] Note that for the remainder of this paper, we have arbitrarily chosen to use feminine pronouns to refer to the Tutor and masculine pronouns to refer to the Learner.

2 Education Dialogue Theory

The components of an education dialogue are as follows [22, 28]:

- Σ_i represents the *knowledge base*, or beliefs of each agent i. Thus, the Tutor's knowledge base is Σ_T and the Learner's knowledge base is Σ_L. The term Σ loosely refers to all the beliefs of an agent.

- An argument (S, p) is a pair, where p is the conclusion and S is the support for that conclusion. p is a logical consequence of S, and S is a minimal subset of Σ from which p can be inferred.

- $\mathcal{A}(\Sigma)$ is the set of all arguments that can be made from Σ.

- $\underline{S}(\Sigma)$ is the set of all acceptable arguments in Σ. Arguments that are acceptable are those that an agent has no reason to doubt: there are either no arguments that *undercut* them, or all the arguments that undercut them are themselves undercut by an acceptable argument.

- An agent's *commitment store*, $CS \in \Sigma$, refers to statements that have been made in the dialogue and which the agents are prepared to defend. CS_T refers to the Tutor's commitment store (statements the Tutor has made), and CS_L refers to that of the Learner. We can think of Σ as the agent's private knowledge base—all of the agent's beliefs—whereas CS is the agent's public knowledge base—all the beliefs that the agent has discussed in public (i.e., with other agents).

Parsons *et al.* [22] show how these simple elements can be used to construct common dialogues, such as information seeking dialogues.

In our earlier work [28], we introduced a new type of knowledge, which we call *meta-knowledge*. This is knowledge about the other agent(s) engaged in the dialogue, as perceived by each agent. We represent this meta-knowledge using Γ, which is a partition of Σ, in the same way that CS is. (Later, we will see that it is convenient to maintain these separate partitions of Σ.) We use the term $\Gamma_i(j)$ to refer to the meta-knowledge held by agent i about agent j. So, $\Gamma_T(L)$ refers to the Tutor's beliefs about the Learner's beliefs, i.e., what the Tutor believes is in the Learner's knowledge base, Σ_L. In addition, we use the $+$ modifier, as in $\Gamma_T(L+)$, to refer to the Tutor's beliefs about what the Learner can acquire. There is a vast literature on modeling the knowledge of learners, which is formally called *student modeling* (or *user modeling* in the more general sense) [21, 32]. Such models typically are designed for a specific domain, often in conjunction with the development of a particular tutoring system. Here we are not concerned with the precise details of individual student models, but rather use the concept abstractly in order to refer to the Tutor's meta-knowledge about the Learner—the Tutor's beliefs about what the Learner knows.

We can also use $\Gamma_L(T)$ to refer to the Learner's beliefs about the Tutor. This concept is useful, for example, for considering the Learner's motivation to learn and his emotional state, both of which are discussed as important aspects for understanding human learners [16] and have been used in agent-based models of human behavior [30]. If the Learner does not believe that the Tutor knows the (correct) answers to questions about the Learner's domain, then he may be less motivated to progress when interacting with the Tutor. Take a real-world example: when students evaluate faculty members at the end of a term, it is common to ask the students to rate their professor's knowledge of the subject (the domain) covered in the course they just completed. Such a question assumes that the students form an opinion (acquire a model) of their professor's knowledge of (beliefs about) the domain. Elaboration on this aspect is beyond the scope of this paper, so here we will limit our discussion of Γ to refer only to the Tutor's beliefs about the Learner, $\Gamma_T(L)$.

2.1 Fundamental interactions

Below, we describe the fundamental steps in an interaction taking place between a Tutor and a Learner. The interaction is illustrated in Figure 1. Five fundamental steps and seven locutions are depicted. The goal is to arrive at the knowledge acquisition state in step 4. As noted below, some of the locutions are taken or derived from earlier work, primarily [1] and [23]. The *operational semantics* for each locution are detailed below (in boxes), in the order in which the locutions appear in the dialogue. The exchange is assumed to be a synchronized, turn-taking interaction that starts with the Tutor.

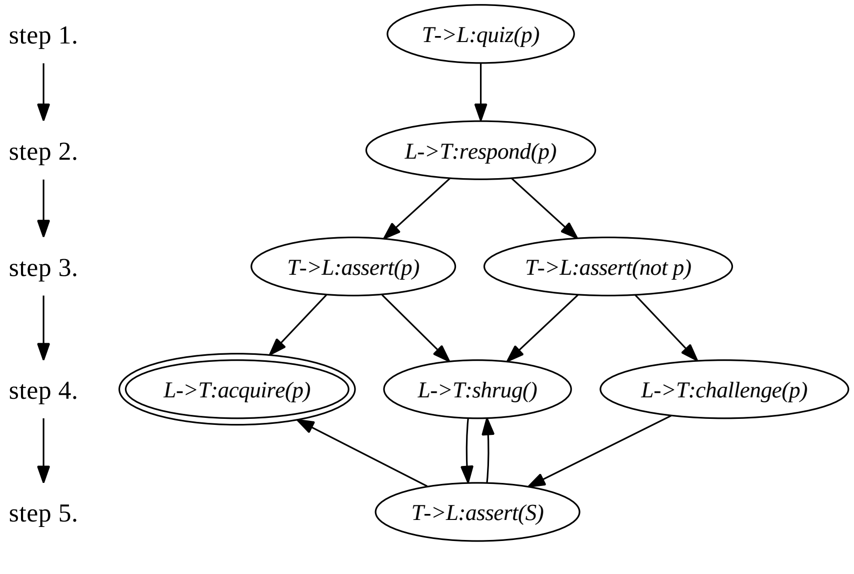


Fig. 1. Interaction sequence between a Tutor (T) and a Learner (L).

1. First, T poses a question to L about the verity of a proposition, p:

   $$T \rightarrow L : quiz(p)$$

   T is seeking to determine if the proposition p is in L's belief set. The Tutor knows the answer to the question, but does not know whether the Learner knows the answer. The goal of this dialogue is for the Tutor to determine if the Learner knows the answer.

   quiz

LOCUTION:	$T \rightarrow L : quiz(p)$
PRE-CONDITIONS:	1. $p \in \Sigma_T$
	2. $(S, p) \in \underline{S}(\Sigma_T)$
	3. $(S, p) \in \underline{S}(\Sigma_T \cup CS_L)$
	4. $p \in \Gamma_T(L+)$
	5. $(S, p) \in \underline{S}(\Gamma_T(L+))$
POST-CONDITIONS:	1. $CS_{T,i} = CS_{T,i-1} \cup \{p\}$ (update)
	2. $CS_{L,i} = CS_{L,i-1}$ (no change)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1}$ (no change)

   Note that this is semantically different from *question*(p) in an information seeking dialogue [1] because T already knows the answer to the quiz and so the purpose of the locution is to determine if L knows the answer. Although the format is similar to *question*(p), the pre and post conditions are sufficiently different that we have defined this new locution, *quiz*(p). A post-condition adds p to CS_T, even though $p \in \Sigma_T$, because this provides an explicit means for the Tutor to keep track of which propositions she has already discussed with the Learner. For convenient comparison, the operational semantics for *question* are listed in the Appendix at the end of this paper.
   Also note the use of $\Gamma_T(L+)$ which represents the Tutor's belief that the Learner can find the answer to the question posed. The Tutor does not know for sure that $p \in \Gamma_T(L)$, but this notation permits the locution to be uttered and a reason for T believing that L can respond correctly.

2. Then, L responds to T's question:

   $$L \rightarrow T : respond(p)$$

   The Learner may or may not know the "right" answer—the correctness will be determined later in the dialogue by the Tutor. But in order to utter p, the Learner must possess some knowledge about p, either in its own knowledge base, Σ_L (meaning that the Learner has acquired p at some point), or in the Tutor's commitment store, CS_T (meaning that the Learner has not yet acquired p in its own knowledge base, Σ_L, but has the opportunity to do so, because it has heard p uttered by T at some earlier point in the dialogue and thus $p \in CS_T$).

respond

LOCUTION:	$L \rightarrow T : respond(p)$
PRE-CONDITIONS:	1. $p \in (\Sigma_L \cup CS_T)$
POST-CONDITIONS:	1. $CS_{T,i} = CS_{T,i-1}$ (no change)
	2. $CS_{L,i} = CS_{L,i-1} \cup \{p\}$ (update)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1}$ (no change)

The *respond* locution differs from the *assert* locution, discussed below, because it puts fewer requirements in the pre-conditions of the uttering agent. In order to be able to use $assert(p)$, the agent must believe p ($p \in \Sigma$) and must be able to support p ($(S, p) \in \Sigma$); whereas in order to be use $respond(p)$, the agent must either believe p or have heard the other agent state p: $p \in (\Sigma \cup CS)$.

3. The locution uttered by the Tutor in the next step depends on the correctness of the response given in the previous step.

   (a) If L has responded with the "correct" answer (i.e., T believes p), then T provides positive feedback, asserting p as described in [1]:

   $$T \rightarrow L : assert(p)$$

   assert(p)

LOCUTION:	$T \rightarrow L : assert(p)$
PRE-CONDITIONS:	1. $p \in \Sigma_T$
	2. $(S, p) \in \underline{S}(\Sigma_T \cup CS_L)$
POST-CONDITIONS:	1. $CS_{T,i} = CS_{T,i-1} \cup \{p\}$ (update)
	2. $CS_{L,i} = CS_{L,i-1}$ (no change)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1}$ (no change)

   (b) If L has responded with the "incorrect" answer, (i.e., T believes $\neg p$), then T provides negative feedback by asserting $\neg p$:

   $$T \rightarrow L : assert(\neg p)$$

   The operational semantics of $assert(\neg p)$ are the same as above, by consistently substituting $\neg p$ for p.

4. The next step depends on the Tutor's response in the previous step, described above.

   (a) If L receives feedback from T that L understands, then L acknowledges that feedback and adds p (in the case of positive feedback) or $\neg p$ (in the case of negative feedback) to its knowledge base[4]:

$$L \rightarrow T : acquire(p)$$

We leave discussion of how exactly the model of the Learner's knowledge base is updated to future work, and refer to [23] for the basis of that discussion.

acquire	
Locution:	$L \rightarrow T : acquire(p)$
Pre-conditions:	1. $p \in (\Sigma_L \cup CS_T)^\dagger$
	2. $(S, p) \in \underline{S}(\Sigma_L \cup CS_T)$
Post-conditions:	1. $CS_{T,i} = CS_{T,i-1}$ (no change)
	2. $CS_{L,i} = CS_{L,i-1} \cup \{p\}$ (update)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1} \cup \{p\}$ (update)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1} \cup \{p\}$ (update)

†We note that it is not standard to allow an agent to utter p when p is not in its knowledge base (Σ), but this is not exactly the case here. In this case, the implementation of the locution includes processing steps in which the uttering agent (L) first adds p to Σ_L and then confirms that acquisition by uttering (essentially, reiterating) p.

(b) If L receives feedback from T that he does not understand, then L can pose a follow-up request for clarification. The appropriate locution is *challenge*(p), as outlined in [1], because the goal of L is to make T subsequently state her arguments in support of p:

$$L \rightarrow T : challenge(p)$$

challenge	
Locution:	$L \rightarrow T : challenge(p)$
Pre-conditions:	1. $p \in CS_T$
Post-conditions:	1. $CS_{T,i} = CS_{T,i-1}$ (no change)
	2. $CS_{L,i} = CS_{L,i-1}$ (no change)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1}$ (no change)

(c) If L receives feedback from T that he does not understand and L is so confused that he does not know what to say next, then he can shrug:

$$L \rightarrow T : shrug()$$

shrug	
Locution:	$L \rightarrow T : shrug()$
Pre-conditions:	none
Post-conditions:	none

[4] For simplicity, we use p in the operational semantics description, but $\neg p$ could also be substituted, as long as the substitution was consistent.

This locution simply serves as a “no-op” (null operation) in order to be consistent with the turn-taking synchronized interaction in the implementation of our interactive learning framework (discussed in Section 4).

5. A final, optional, step occurs if the Learner does not understand the Tutor’s feedback and has replied with a *shrug* or *challenge* locution in the previous step. In both cases, the Tutor responds by providing an explanation for p, using the $assert(S)$ locution described in [1]:

$$T \rightarrow L : assert(S)$$

assert(S)

Locution:	$T \rightarrow L : assert(S)$
Pre-conditions:	1. $p \in \Sigma_T$
	2. $(S, p) \in \underline{S}(\Sigma_T)$
Post-conditions:	1. $CS_{T,i} = CS_{T,i-1} \cup (S, p)$ (update)
	2. $CS_{L,i} = CS_{L,i-1}$ (no change)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)
	5. $\Gamma_T(L)_i = \Gamma_T(L)_{i-1}$ (no change)

3 Interactive Learning Systems

Intelligent Tutoring Systems (ITS) are a type of Interactive Learning System that provide users with opportunities to learn by interacting with a computer [26]. Unlike traditional computer-aided instruction, ITSs are not static, preprogrammed systems; rather, they adapt to students’ responses. ITSs interject methodologies from artificial intelligence (AI) to manage that adaptivity, dynamically orchestrating users’ learning experiences. An ITS uses a range of AI techniques to make decisions about which problem or information to present to a learner, and when and how to intervene if the learner makes mistakes.

Beck *et al.* [5] identify five major components in an ITS:

- The *domain model* contains the essential knowledge representation of the instructional domain. Both the pedagogical module and the student model (below) use the domain knowledge module to interpret a student’s solution and track her skills.
- The *student model* records information about a student’s performance with or misconception of the materials being taught. The idea is to build up a representation of a student’s knowledge and skill set, updating this representation over time, as the student interacts with the system.
- The *pedagogical module*, or *tutor*, is the instructional, or teaching, component [27] which contains a set of rules about how to control and influence the student’s learning process. The tutoring system uses this module to guide the student through the knowledge domain [29].

- The *expert model* contains knowledge about the cognitive structures and solution strategies underlying expertise in that particular domain. By using this model, the tutor can compare the student's solution with the expert's solution in order to figure out where learners have difficulties.
- The *communication module* provides the interface between the user and the tutor.

Classic ITS systems include the *LISP Tutor* [3, 8] for teaching the LISP programming language, and the *Andes* tutor [31] for teaching physics. Each is described briefly below.

The LISP Tutor [3, 8] incorporates *ACT**, a psychological theory of skill acquisition [2] and uses production rules and model tracing to model the tutor. Model tracing models errors that students make at each step on the basis of known misconceptions. By comparing the students' responses to the set of possible legal actions and the set of known wrong actions, the tutor is able to recognize whether the student is on a correct solution path, or appears be suffering from known misconceptions, or something unrecognizable. The student model in the LISP Tutor is partly descriptive and partly prescriptive. It is based on the authors' observations of students learning LISP and from the analysis of the required knowledge for LISP programming, as well as good programming styles. Procedural knowledge of how to write LISP code is modeled by a set of production rules.

Andes uses Bayesian networks for its student modeling component [7, 10, 18]. Every time the student selects a new problem, a Bayesian network is automatically generated. The structure of the the network is taken directly from a solution graph embedded in the system. The network contains five kinds of nodes:

- *Context-Rule nodes* model the ability to apply a rule in a specific problem-solving context in which it may be used. Each Context-Rule corresponds to a rule in Andes' ruled-based problem solver.
- *Fact nodes* represent the probability that the student knows a fact that is part of the problem solution.
- *Goal nodes* represents the probability that the student has been pursuing a goal that is part of the problem solution.
- *Strategy nodes* correspond to points where the student can choose among alternative plans to solve a problem.
- *Rule-Application nodes* represent the probability that the student has applied a piece of physics knowledge represented by a context-rule to derive a new fact or goal.

The Bayesian networks in Andes encode two kinds of knowledge: *domain-general knowledge*, which holds information about general concepts and procedures that define proficiency in Newtonian physics, and *task-specific knowledge*, which holds information related to student performance on a specific problem or example. Andes constitutes a probabilistic student model that provides long-term knowledge assessment, plan recognition, and prediction of students' actions during problem solving.

Over the last three decades, an extensive number of ITSs have been built using a range of techniques. Bayesian networks have been employed in multiple systems [6, 13]. Many have branched out to incorporate other techniques, such as object-oriented architectures (e.g., [34]). Various methodologies have been explored for emulating human best teaching practices, such as *coached program planning* [15], which helps students decompose problems. Some systems use natural language dialogues for interacting with students (e.g., [14]). An increasing number of systems take advantage of agent-based and multi-agent architectures [25]. Some incorporate intelligent interface components such as *pedagogical agents* [12]. However, to the best of our knowledge, no ITS system uses an argumentation-based framework or the education dialogues we have described above.

4 The ArgILS Framework

In this paper, we are concerned with the student model (the Learner), and the Tutor. Section 2 explained how to represent the Learner's knowledge (Σ_L), the expert's knowledge (Σ_T)[5], and the Tutor's knowledge about the Learner ($\Gamma_T(L)$); and provided an interaction structure for using that knowledge. This section introduces our *ArgILS* framework, which we have designed as a means for applying argumentation-based education dialogue theory to an interactive learning system. We describe our framework and ground it with an example.

The interaction sequence illustrated in Figure 1 and detailed in Section 2.1 outlines the fundamental series of steps in a theoretical education dialogue. This sequence is reasonable for interacting about *declarative* (factual) knowledge, where p can represent a fact and step 1 can be the Tutor asking the Learner if p is true. But the theory needs to be expanded in order to handle *procedural* knowledge. We need to provide a mechanism to communicate procedural information that cannot be expressed simply as a single proposition p. For example, the Tutor may ask the Learner how to execute a particular task, to which the Learner should be able to respond by uttering a series of propositions that all belong to a sequential procedure.

We represent a procedural sequence, $\overrightarrow{p}$ as:

$$\overrightarrow{p} = \{p_0, p_1, p_2, \ldots, p_{n-1}\}$$

Such a procedural sequence can be integrated into the interaction steps shown in Figure 1 in multiple ways. The first step, in which the Tutor puts forth a question to the Learner, remains essentially unchanged, with the substitution of $\overrightarrow{p}$ for p:

$$L \rightarrow T : quiz(\overrightarrow{p})$$

The second step, in which the Learner responds, however, will necessarily change.

[5] We make the assumption that the Tutor is the "expert".

Because the procedural knowledge is broken down into a number of pieces, there is a choice about redefining step 2 to:

$$L \rightarrow T : respond(\overrightarrow{p}) \tag{1}$$

where $\overrightarrow{p}$ represents all steps in the procedural sequence, or

$$L \rightarrow T : respond(\{p_i, p_{i+1}, \ldots, p_{j-1}, p_j\}) \tag{2}$$

where $\{p_i, p_{i+1}, \ldots, p_{j-1}, p_j\}$ represents some number of steps in the sequence, or

$$L \rightarrow T : respond(p_i) \tag{3}$$

where p_i represents one step in the procedural sequence.

One of the architecture decisions that arises in building an interactive learning system concerns *feedback*: when should the tutoring system provide help to the Learner? Equations 1 and 2 represent *delayed feedback*, where the Learner completes all or part of the task before receiving any feedback from the Tutor. Equation 3 represents *immediate feedback*, where the Learner completes only one step in the task before receiving feedback from the Tutor.

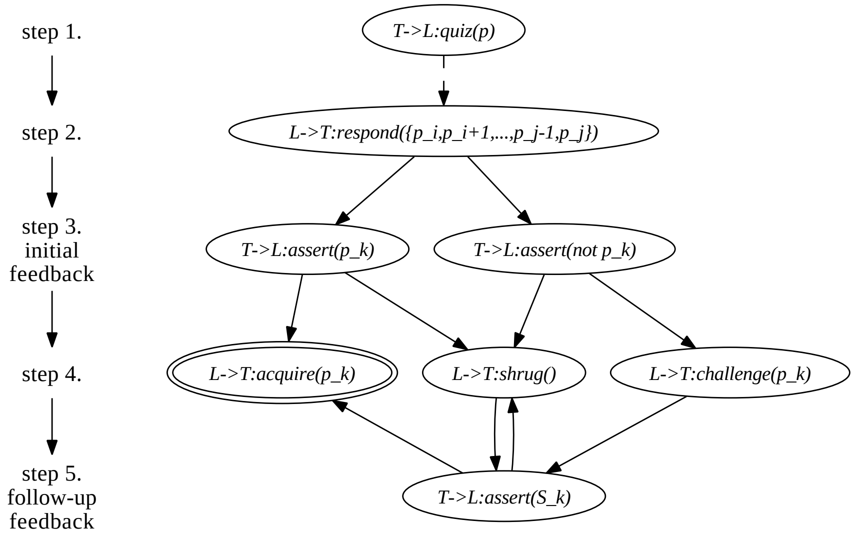


Fig. 2. Interaction sequence for *delayed feedback*.

Figure 2 illustrates an interaction sequence with delayed feedback. The first step is the same as in Figure 1, with the Tutor asking about the entire procedural sequence. The difference from Figure 1 lies in the second step, which is highlighted by the dashed line that leads from the first to the second step. In the second step, the Learner responds with some number of propositions in the procedural sequence, as in equation 2. This can also correspond to equation 1, if $i = 0$ and $j = n - 1$ (i.e., equation 1 is just a specialized case of equation 2). The third step is the initial feedback step, where the Tutor comments on one of the propositions posited by the Learner, where $i \leq k \leq j$. The fourth, follow-up step and the fifth, follow-up feedback step proceed the same as in Figure 1, in response to the proposition p_k chosen by the Tutor in step 3. Note that the Tutor must decide which p_k to provide feedback for. Indeed, it is possible for the Tutor to comment on multiple p_k's; though for simplicity here, we only consider situations where the Tutor comments on one p_k at a time, and leave simultaneous commenting on multiple p_k's to future work.

Figure 3 illustrates an interaction sequence with immediate feedback. The picture is almost identical to that of Figure 1, with the difference being that the Tutor starts with $quiz(p)$, asking about the entire procedural sequence, and the Learner answers with a single step in the procedure: $respond(p_i)$. The dashed line from the first step to the second step highlights this difference.

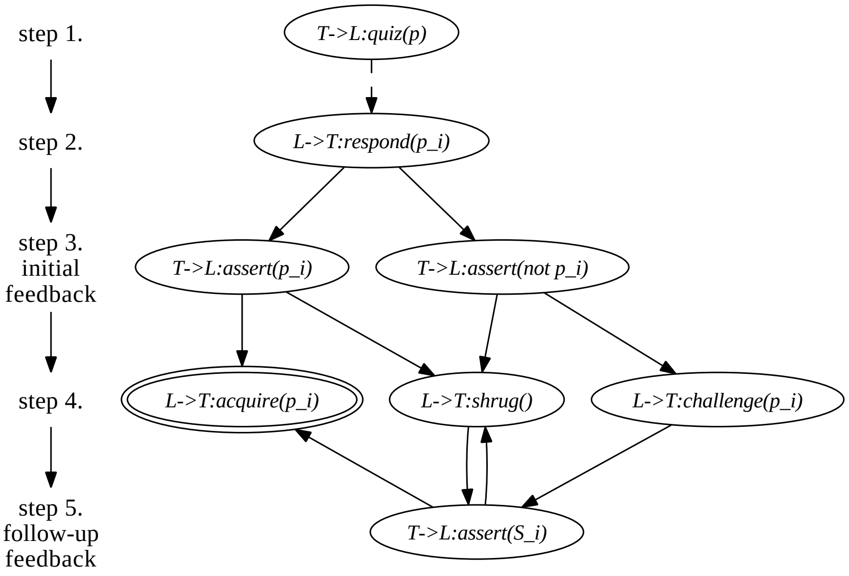
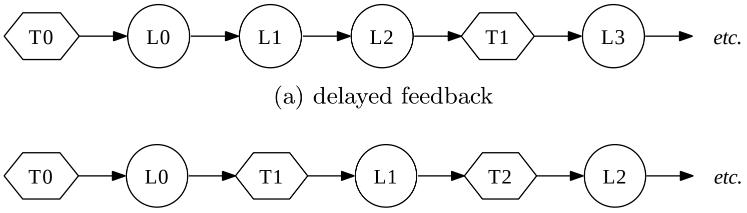


Fig. 3. Interaction sequence for *immediate feedback*.

Figures 2 and 3 illustrate the interactions over some portion of the procedural sequence. Unless the Learner provides the entire $\overrightarrow{p}$ and delayed feedback is employed and the Learner's response is entirely correct, some amount of iteration must occur before the Learner has received feedback on the entire procedural sequence. Figure 4 illustrates abstractly the differences in iteration patterns between delayed feedback and immediate feedback. With immediate feedack, every time the Learner makes an utterance, the Tutor replies immediately. With delayed feedback, the Tutor waits for the Learner to make several utterances, and then replies. The timing of the reply on the part of the Tutor in a delayed feedback system is another open area of research, and is something we will examine in future work. The important observation to make here is that we can model these differences in our ArgILS framework.



(a) delayed feedback

(b) immediate feedback

Fig. 4. Iterative sequences

Finally, we introduce one more locution, for use in iterative situations, as above, where the system is using immediate feedback—requiring that the Tutor respond immediately to every action on the part of the Learner. However, once the Learner acquires a proposition in the procedural sequence, he continues by positing the next step, without the Tutor reiterating the initial question. For just this case, in order to maintain the synchronized turn-taking in the iterative process, we introduce a "no-op" for the Tutor, which we call *nod*:

$$T \rightarrow L : nod()$$

nod
Locution: $T \rightarrow L : nod()$
Pre-conditions: none
Post-conditions: none

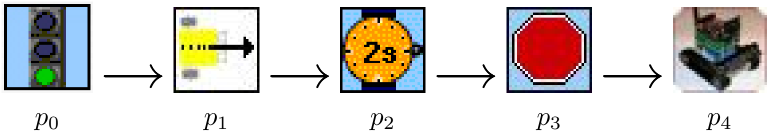
4.1 An example interaction

Below we enumerate an example using our Human-Robot Tutoring System (HRTS) in which a Learner is trying to acquire knowledge about how to program a robot. Our HRTS is called RoboLite [4], and is based on the popular RoboLab [9, 24] programming interface originally designed for LEGO Mindstorms RCX robots [17].

In the first step, the Tutor utters:

$$T \rightarrow L : quiz(\overrightarrow{p}) \tag{step 1.}$$

where $p =$ "How do you program a robot to go forward for 2 seconds and then stop?" Our system uses a graphical interface, where each command given to control the robot is represented as a building block icon. The expert's solution to the question is shown below:



In the second step, the Learner posits an icon. Let's say that the Learner starts with the correct icon, represented here by proposition p_0, so the Learner utters:

$$L \rightarrow T : \mathit{respond}(p_0) \tag{step 2.}$$

In an immediate feedback system, the Tutor would immediately reply with positive feedback:

$$T \rightarrow L : \mathit{assert}(p_0) \tag{step 3a.}$$

Since this is correct and the Learner's belief is confirmed, the Learner updates his knowledge base: $\Sigma_L = \Sigma_L \cup p_0$, and reiterates with:

$$L \rightarrow T : \mathit{acquire}(p_0) \tag{step 4a.}$$

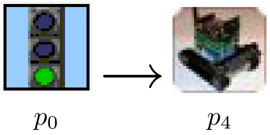
This is where the null operation is needed for the Tutor, to maintain the turn-taking pattern, but without reiterating any propositions unnecessarily or introducing anything new. Thus, the Tutor indicates that the Learner should proceed by uttering:

$$T \rightarrow L : \mathit{nod}()$$

Now the Learner adds another icon. Let's say he makes a mistake and enters p_4:

$$L \rightarrow T : \mathit{respond}(p_4) \tag{step 2.}$$

so his partial solution would look like this:



Again, in an immediate feedback system, the Tutor would reply immediately. The Tutor compares the Learner's sequence, $\{p_0, p_4\}$, with the expert sequence, $\{p_0, p_1, \ldots\}$, and detects an anomaly with the second proposition in the sequence. So this time, the Tutor comments with negative feedback:

$$T \rightarrow L : \mathit{assert}(\neg p_4) \tag{step 3b.}$$

The Learner does not understand why his sequence is incorrect, so he requests clarification by uttering:

$$L \rightarrow T : \mathit{challenge}(\neg p_4) \tag{step 4b.}$$

whereby the Tutor responds by providing the reasons why p_4 is the incorrect proposition in the sequence:

$$T \rightarrow L : \mathit{assert}((S, \neg p_4)) \qquad \text{(step 5.)}$$

An alternative to the Tutor providing a negative assertion (as in step 3b, above) is for the Tutor to provide the Learner with the right answer by uttering:

$$T \rightarrow L : \mathit{assert}(p_1) \qquad \text{(step 3a.)}$$

If the Learner does not understand, then he would again ask for clarification:

$$L \rightarrow T : \mathit{challenge}(p_1) \qquad \text{(step 4b.)}$$

and the Tutor would supply the reasons why p_1 is the correct proposition in the sequence:

$$T \rightarrow L : \mathit{assert}((S, p_1)) \qquad \text{(step 5.)}$$

In a delayed feedback system, the Tutor would wait until the Learner had entered several icons before commenting. The questions of when to respond and how to respond are areas of future research to be addressed in the development of our implemented system. The ArgILS provides a solid framework in which to model the possibilities.

5 Summary

We have described an extended education dialogue system, expanding on our earlier work and that of others in the argumentation dialogue community. We have introduced ArgILS, our general framework for an interactive learning system in which interactions between a Tutor and a Learner can be modeled. An example was provided, demonstrating the use of ArgILS in the development of our work-in-progress Human-Robot Tutoring System. Multiple avenues of future work have been identified, such as the Tutor's choice of which proposition to comment on in a delayed feedback system for procedural knowledge and when to provide comments in a delayed feedback system.

Appendix

Below are the operational semantics of the *question* locution, adapted from [1]. A question is posed when the initiating agent, T in the description below, asks a question of another agent, L in the description below. In the case of a question, it is assumed that the asking agent does not know the answer to the question, p in the description below. In addition, the asking agent does not know whether the target agent knows the answer or not. (This is revealed in the choice of response locution subsequently executed by the target agent.)

question	
LOCUTION:	$T \rightarrow L : question(p)$
PRE-CONDITIONS:	1. $(S, p) \notin \underline{S}(\Sigma_T)$
	2. $(S, \neg p) \notin \underline{S}(\Sigma_T)$
POST-CONDITIONS:	1. $CS_{T,i} = CS_{T,i-1}$ (no change)
	2. $CS_{L,i} = CS_{L,i-1}$ (no change)
	3. $\Sigma_{T,i} = \Sigma_{T,i-1}$ (no change)
	4. $\Sigma_{L,i} = \Sigma_{L,i-1}$ (no change)

Acknowledgments

We would like to thank Simon Parsons, for his collaboration on our original work on education dialogues. We would also like to extend our gratitude to the reviewers of this paper, who provided many thought-provoking comments that will help us shape some of the next steps in the development of our framework.

References

1. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: Proceedings of the 4th Conference on Multi-Agent Systems. Boston (2000)
2. Anderson, J.R.: The Architecture of Cognition. Harvard University Press (1983)
3. Anderson, J.R., Skarecki, E.: The automated tutoring of introductory programming. Communications of the ACM 29(9), 842–849 (September 1986)
4. Azhar, M.Q., Goldman, R., Sklar, E.I.: An agent-oriented behavior-based interface framework for educational robotics. In: Agent-Based Systems for Human Learning (ABSHL) Workshop at Autonomous Agents and MultiAgent Systems (AAMAS) (2006)
5. Beck, J., Stern, M., Haugsjaa, E.: Applications of AI in Education. Crossroads 3(1), 11–15 (1996)
6. Beck, J.E., m. Chang, K., Mostow, J., Corbett, A.: Does help help? introducing the bayesian evaluation and assessment methodology. In: 9th International Conference on Intelligent Tutoring Systems. pp. 383–394 (June 2008)
7. Conati, C., Gertner, A., VanLehn, K.: Using Bayesian Networks to Manage Uncertainty in Student Modeling. User Modeling and User-Adapted Interaction 12(4) (2002)
8. Corbett, A.T., Anderson, J.R.: The LISP Intelligent Tutoring System: Research in skill acquisition, pp. 73–109. Lawrence Erlbaum, Hillsdale, NJ, USA (1992)
9. Erwin, B., Cyr, M., Rogers, C.B.: LEGO Engineer and ROBOLAB: Teaching Engineering with LabVIEW from Kindergarten to Graduate School. International Journal of Engineering Education 16(3) (2000)
10. Gertner, A.S., Conati, C., VanLehn, K.: Procedural help in Andes: Generating hints using a Bayesian network student model. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). pp. 106–111. AAAI Press (1998)
11. Girle, R.: Commands in Dialogue Logic. In: Practical Reasoning: Proceedings of the First International Conference on Formal and Applied Practical Reasoning (FAPR). pp. 246–260. Lecture Notes in Artificial Intelligence 1085, Springer, Berlin, Germany (1996)

12. Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated pedagogical agents: Face-to-face interaction in interactive learning environments. International Journal of Artificial Intelligence in Education 11 (2000)
13. Kasurinen, J., Nikula, U.: Estimating programming knowledge with bayesian knowledge tracing. In: Proceedings of the 14th annual ACM SIGCSE conference on Innovation and Technology in Computer Science Education (ITiCSE). pp. 313–317. ACM, New York, NY, USA (2009)
14. Lane, H.C., VanLehn, K.: A dialogue-based tutoring system for beginning programming. In: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS). pp. 449–454. American Association for Artificial Intelligence Press (2004)
15. Lane, H., VanLehn, K.: Coached program planning: dialogue-based support for novice program design. In: Proceedings of the 34th SIGCSE technical symposium on Computer science education. pp. 148–152. ACM (2003)
16. Lazarus, R.S.: Cognition and Motivation in Emotion. American Psychologist 45(4), 352–367 (1991)
17. LEGO Mindstorms. http://www.legomindstorms.com/
18. Martin, J., Vanlehn, K.: Student assessment using bayesian nets. International Journal of Human-Computer Studies 42, 575–591 (1995)
19. McBurney, P., Parsons, S.: Agent ludens: Games for agent dialogues. In: Proceedings of the AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents. Stanford, CA, USA (2001)
20. McBurney, P., Parsons, S.: Chance discovery using dialectical argumentation. In: Proceedings of the Workshop on Chance Discovery, Fifteenth Annual Conference of the Japanese Society for Artificial Intelligence. Matsue, Japan (2001)
21. McCalla, G.I., Greer, J.E.: Granularity-based reasoning and belief revision in student models. In: Student Models: The Key to Individualized Educational Systems. pp. 39–62. Springer Verlag, New York (1994)
22. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
23. Parsons, S., Sklar, E.: How agents alter their beliefs after an argumentation-based dialogue. In: Proceedings of the Workshop on Argumentation in Multiagent Systems (ArgMAS) at Autonomous Agents and MultiAgent Systems (AAMAS) (2005)
24. Robolab. http://www.ceeo.tufts.edu/robolabatceeo/
25. Sklar, E.I., Richards, D.: Agent-based systems for human learners. Knowledge Engineering Review 25(2), 111–135 (June 2010)
26. Sklar, E.: CEL: A Framework for Enabling an Internet Learning Community. Ph.D. thesis, Department of Computer Science, Brandeis University, Waltham, MA, USA (2000)
27. Sklar, E., Davies, M.: Multiagent simulation of learning environments. In: Fourth International Conference on Autonomous Agents and Multi Agent Systems (AAMAS) (2005)
28. Sklar, E., Parsons, S.: Towards the Application of Argumentation-based Dialogues for Education. In: Proceedings of the Third International Conference of Autonomous Agents and Multi Agent Systems (AAMAS). pp. 1420–1421 (2004)
29. Sklar, E., Richards, D.: The use of agents in human learning systems. In: Fifth International Conference on Autonomous Agents and Multi Agent Systems (AAMAS) (2006)
30. Spoelstra, M., Sklar, E.: Using simulation to model and understand group learning. Agent Based Systems for Human Learning, International Transactions on Systems Science and Applications 4(1) (2008)

31. VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M.: The Andes Physics Tutoring System: Lessons Learned. International Journal of Artificial Intelligence and Education 15(3) (2005)
32. VanLehn, K., Niu, Z., Slier, S., Gertner, A.: Student modeling from conventional test data: A bayesian approach without priors. In: Proceedings of the 4th Intelligent Tutoring Systems Conference (ITS). pp. 434–443 (1998)
33. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY, USA (1995)
34. Wei, F., Moritz, S.H., Parvez, S.M., Blank, G.D.: A student model for object-oriented design and programming. Journal of Computing Sciences in Colleges 20(5), 260–273 (2005)

A Semantics for Dynamic Argumentation Frameworks

Kazuko TAKAHASHI[1] and Yu NAMBU[1]

School of Science&Technology, Kwansei Gakuin University,
2-1, Gakuen, Sanda, 669-1337, JAPAN
ktaka@kwansei.ac.jp cdh67289@kwansei.ac.jp

Abstract. This paper presents a semantics for dynamic argumentation frameworks. A dynamic argumentation system involves the concept of execution of an argumentation affecting subsequent arguments. Although such dynamic treatment is necessary to grasp the behavior of actual argumentation, semantics proposed so far can only handle the static aspects of argumentation. Here, we present a new semantics that fits dynamic argumentation. We discuss what properties hold and explain how to compute changes in the set of acceptable arguments, depending on the presenting order of arguments.

1 Introduction

Argumentation is a powerful tool that enables the formal treatment of interactions, such as negotiation and agreement, among agents. There have been lots of studies on argumentation systems [4, 21].

An argumentation framework is usually defined as a pair $\langle Args, Atts \rangle$, where $Args$ is a set of arguments, and $Atts$ is a binary relation over $Args$ that indicates an attack by one argument on another. Most argumentation systems developed to date analyze a given argumentation framework statically. They regard an argumentation theory as fixed or focus on the selection of a specific argumentation theory that will result in a particular proposals being accepted. These systems are based on the assumption that arguing agents have a common knowledge base and can survey all possible arguments. However, knowledge bases actually differ between agents, so as each argument is presented, new information is added to modify the subsequent argumentation. We developed a dynamic argumentation system, *"the Argumentation Procedure with Knowledge Change (APKC),"* in which argumentation theory changes depending on the execution [19], and its extended version, APKC2 [20]. Our goal was to capture more precisely the behavior of actual argumentation. The proposed system is based on the concept of "execution" of an argument. We investigated the phenomenon in which new information is added by a presented argument, and this generates a new attack.

In APKC2, an argumentation continues over multiple branches. We demonstrated that the results may differ depending on the order of execution. We also proposed a judgment algorithm, JC, which can determine which agent wins without actually simulating each execution individually [20]. Although this previous work investigated simulation and judgment in dynamic argumentation, it did not clarify the meaning of each execution and the relationships between executions. In this paper, we present a new semantics to fit the dynamic argumentation system.

A semantics for an argumentation system is usually given with the notion of extension [11], i.e., a set of arguments that can be accepted together within a given argumentation framework. However, in dynamic argumentation, arguments and attacks change. Therefore, a semantics in which acceptability is defined for a static argumentation is not suitable for dynamic argumentation. In this paper, we present a separate extension for each execution as an acceptable set of arguments for that execution. An extension for a dynamic argumentation system is defined as the set of these individual extensions. Additionally, we discuss how these extensions are changed as argumentation proceeds and investigate their interrelationships and properties.

This paper is organized as follows. In section 2, we explain the need for dynamic argumentation. In section 3, after presenting basic concepts such as argumentation frameworks, we present a dynamic argumentation system. In section 4, we define the semantics for a dynamic argumentation frameworks, and show the rules by which the revision of extensions is computed. In section 5, we compare our approach with those used in related studies and discuss the effectiveness of our semantics. Finally, in section 6, we present our conclusions.

2 Informal Description for Dynamic Argumentation

In general, argumentation involves two agents taking turns presenting arguments to attack their respective opponent's argument until one is no longer able to attack. Finally, the loser accepts the winner's proposal. This process is usually represented in the form of a tree [1, 13]. The root node is a proposal statement, and each branch corresponds to a single argumentation line, i.e., a sequence of arguments. In a dynamic argumentation system [20], an argumentation proceeds along each branch. Once an argument is presented, the corresponding node is marked as "executed" and never reappears in the series of argumentation. If there is no executable node in the current branch, another branch that has an executable node is then selected. Finally, the agent that cannot make a counterargument loses the argumentation. An important feature of this system is the concept of a "threat." This refers to a case in which the execution of an argument results in the creation of a new counterargument to another argument. Intuitively, a threat is an argument that may provide information advantageous to the opponent. It acts to change the argumentation and affects the win/loss outcome of the argumentation.

For example, consider the argumentation tree shown in Figure 1(1). If we execute the argumentation from the left branch, after P_0, C_1, P_1 are executed, C_2, P_2 are executed, and P wins. If we execute from the right branch, after P_0, C_2, P_2 are executed, C_1, P_1 are executed, and P also wins. Now, consider the argumentation tree shown in Figure 1(2), which has a threat from C_1 to C_2. It means that the execution of C_1 causes the creation of P_2, a new counterargument to C_2. If we execute an argumentation from the left branch, after P_0, C_1, P_1 are executed, P_2 is generated. Then, C_2, P_2 are executed and finally, the execution terminates with P winning. In contrast, if we execute the argumentation from the right branch, after P_0, C_2 are executed, the execution terminates because C has the next turn, but no branch is available that can start by C's argument. In this case, C wins. Note that P_2 does not occur until C_1's execution. This example illustrates two important issues that need to be addressed: (i) the winner of an argumentation differs depending on the order of execution of the branches, and (ii) it is not appropriate to handle a revised tree in the same way as one that consists of the same nodes and edges without a threat.

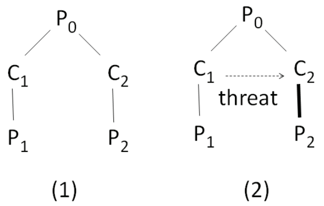


Fig. 1. Example of argumentation trees

3 Basic Concepts

3.1 Dynamic argumentation framework

In a dynamic argumentation agents of a proposer (P) and a defeater (C) have their own knowledge bases, which may have common elements. We construct a dynamic argumentation framework from given knowledge bases of agents and preference [19]. Preference is defined in advance for each formula in the knowledge base. The preference of each argument can be computed so that attack is possible only from an argument with a high preference to an argument with a lower preference. In this paper, we do not explain the construction process from knowledge bases and preference, but we assume that an argumentation framework with threats is given, and we treat a dynamic argumentation at an abstract level.

Definition 1 (argumentation framework) *An argumentation framework is defined as a triple $AF = \langle Arg_P, Arg_C, Atts\rangle$, where Arg_P and Arg_C are sets of P's arguments and C's arguments, respectively, Atts is a binary relation called* attack *over $Arg_P \cup Arg_C$, where for each $(A, B) \in Atts$, either $A \in Arg_P, B \in Arg_C$, or $A \in Arg_C, B \in Arg_P$ holds. For each pair of arguments A, B, both (A, B) and (B, A) are never contained in Atts at the same time.*

Definition 2 (argumentation tree) *Let φ be a proposal statement, and let P and C be a proposer and a defeater of φ, respectively. Let AF be an argumentation framework $\langle Arg_P, Arg_C, Atts\rangle$. Then, an argumentation tree for AF on φ is defined as follows [1].*

- *This is a finite, directed tree whose root node corresponds to a pro-argument to φ* [1].
- *Every node corresponds to an argument in $Arg_P \cup Arg_C$.*
- *Every edge from node N to M corresponds to an attack from an argument corresponding to N to that corresponding to M.*

Here, we call a path from the root node to a leaf node *a branch*. P's argument and C's argument appear in turn in each branch. The same arguments may be present in different branches, which follows that each node has a unique parent node. There is no loop in each branch due to the constraint of preference.

Definition 3 (win of a branch) *If the leaf of a branch D is P's argument, then it is said that P* wins *D; otherwise, P* loses *D.*

Definition 4 (candidate subtree) A candidate subtree *is a subtree of an argumentation tree that selects only one child node for each node corresponding to C's argument in the original tree and selects all child nodes for each node corresponding to P's argument.*

Definition 5 (solution subtree) A solution subtree *is a candidate subtree in which P wins all of the branches in the tree.*

In most argumentation systems, the win/loss of an argumentation is defined by handling each branch independently. But in a dynamic argumentation system, another branch may continue to be executed after all arguments of one branch are executed. In this case, arguments disclosed so far in one line affect arguments in another line. This may create a new argument and change the winner of the argumentation. This is the most characteristic feature of dynamic argumentation system.

[1] In general, there may exist multiple arguments whose sentence is φ with different grounds in Arg_P. Therefore, precisely speaking, the root is regarded as an empty argument, and the arguments to support φ should be regarded as its child nodes [19]. However, to simplify, we consider a simple version by assuming that there exists only one such argument and taking it as the root node.

3.2 Execution of an argumentation

Here, we present a dynamic argumentation system.

We first introduce a concept of the "execution" of an argumentation.

Both agents have their own knowledge bases. A set of all the formulas contained in all arguments given so far is stored in a commitment store [15]. We also prepare histories for each agent P and C, respectively, to preserve the coherence of each agent's arguments. First, for a given argumentation framework, we construct an initial argumentation tree. An argumentation starts by selecting a branch of an initial argumentation tree. It proceeds along the branch, and when the execution reaches the leaf node, the branch is suspended. At that time, the commitment store is updated, and agents can make new arguments using the commitment store in addition to their own knowledge bases. Therefore, the number of a set of arguments and that of a set of attacks increase in accordance with the execution of each branch. New nodes are added to the argumentation tree if new arguments are generated. Next, another branch is selected. In the execution procedure, the executed node is marked, and the branch containing unmarked nodes can be selected. The suspended branch may be resumed if a new unmarked node is added to it. Upon the selection of a branch, the utterance turns should be kept. This means that if one branch is suspended at the node that corresponds to one agent's argument, then the next branch should start with the node that corresponds to the other agent's argument.

Definition 6 (executable node) *For a node M_i ($1 \leq i \leq n$) in a branch $D = [M_1, \ldots, M_n]$ and a current turn t, if $M_1, \ldots, M_{i-1}$ are marked, $M_i, \ldots, M_n$ are unmarked, and M_i is t's argument, then the node M_i is said to be* executable*.*

Let $D = [M_1, \ldots, M_n]$ be a branch, $\mathbf{H}_P$ and $\mathbf{H}_C$ be histories for P and C, respectively, and $\mathbf{K}$ be the commitment store. Figure 2 shows an execution of D from M_i ($1 \leq i \leq n$).

```
Execution of a branch $D$ from a node $M_i$

1. Mark $M_i, \ldots, M_n$.
2. Update $\mathbf{K}$ by adding all the formulas contained in arguments $M_i, \ldots, M_n$.
3. if $M_n$ is P's argument,
     then set the current turn to C and update $\mathbf{H}_P$ by adding all the formulas
     contained in P's arguments in $D$.
   if $M_n$ is C's argument,
     then set the current turn to P and update $\mathbf{H}_C$ by adding all the formulas
     contained in C's arguments in $D$.
```

Fig. 2. Execution of a branch

Definition 7 (suspend/resume) *After the execution of all nodes in a branch, D is said to be* suspended*. For a suspended branch D, if an executable node is* *added to its leaf on the modification of a tree and D is selected, then D is said to be* resumed*.*

Definition 8 (threat) *Let A and A' both be arguments in Arg_P or in Arg_C. If A generates more than one new argument that attacks A', then it is said that A is* a threat to A'*, and that Arg_P/Arg_C* contains a threat*. A and A' are said to be* a threat resource *and* a threat destination*, respectively, and this is denoted by $threat(A, A')$.*

Intuitively, a threat is an argument that may provide information advantageous to the opponent. An argument may be a threat to another argument in the same branch.

We present a formal definition of the execution of an argumentation in Figure 3.

Argumentation Procedure with Knowledge Change (*APKC2*)

Let $AF = \langle Arg_P, Arg_C, Atts \rangle$ be an argumentation framework, and φ be a proposed statement.

```
[STEP 1 (initialization)]
Set $\mathbf{K} = \emptyset$, $\mathbf{H}_P = \emptyset$, $\mathbf{H}_C = \emptyset$. Construct an initial argumentation tree for $AF$ on $\varphi$
with all nodes unmarked.

[STEP 2 (execution of an argumentation)]
if no branch has an executable node,
  if $turn$=P, then terminate with P's loss.
  else $turn$=C, then terminate with P's win.
else select a branch and execute it from the executable node to the leaf node.

[STEP 3 (modification of a tree)]
For a pair of arguments $A, A' \in Arg_P/Arg_C$ such that $threat(A, A')$ holds,
if $A$ is marked,
  then add a new argument $B$ to $Arg_C/Arg_P$, respectively,
    and add a new attack $(B, A')$ to $Atts$[a].
if the nodes $N$ and $M$ are identical, and $N$ is marked while $M$ is unmarked,
  then mark $M$.
go to STEP 2.
```

[a] In fact, threats are derived from a set of formulas contained in the arguments in the marked nodes [20].

Fig. 3. Argumentation Procedure with Knowledge Base (APKC2)

In APKC2, both agents present arguments in turn, and the agent that cannot give a counterargument loses the argumentation. An execution based on a certain order of selecting branches corresponds to an argumentation pattern.

Proposition 1 *[19] (1) Any execution of APKC2 terminates in a finite time, and its winner is decidable.*
(2) The number of executions for an argumentation tree is finite.

Definition 9 (win/loss execution) *If APKC2 along an execution terminates with P's win/loss, then it is said that* P wins/loses the execution.

Definition 10 (execution tree) *For an argumentation framework AF, a subtree of a tree finally obtained as a result of APKC2 along an execution exec, which consists of the marked nodes and the edges between them, is said to be* an execution tree for *exec. It is denoted by T_{exec}.*

Definition 11 (continuous candidate subtree) *For a candidate subtree CT, if more than one candidate subtree is generated by the addition of nodes, then these subtrees are said to be continuous candidate subtrees of CT.*

Definition 12 (dynamic solution subtree) *Let CT be a candidate subtree of an initial argumentation tree. For any execution order of branches of CT, if APKC2 terminates with P's win or CT has a continuous candidate subtree such that P wins, then CT is said to be* a dynamic solution subtree.

Definition 13 (dynamic win of an argumentation) *If an argumentation tree has a dynamic solution subtree, then* P dynamically wins the argumentation *tree; otherwise,* P dynamically loses it.

Let T_{init} and T_{final} be the initial argumentation tree and the final argumentation tree appeared in *APKC2* for $AF = \langle Arg_P, Arg_C, Atts\rangle$, respectively. If there is no threat in Arg_P and Arg_C, then for any execution *exec*, $T_{exec} \subseteq T_{init}$ and $T_{final} = T_{init}$ hold.

4 Semantics

4.1 Extensions

Following the definition set out by Dung [11], we can define the following concepts related to arguments.

Definition 14 (conflict-free, admissible) *For an argumentation framework $AF = \langle Arg_P, Arg_C, Atts\rangle$, let $A \in Arg_P \cup Arg_C$ and $\mathcal{S} \subseteq Arg_P \cup Arg_C$.*
(1) $\mathcal{S}$ is conflict-free *iff there are no elements $A, B \in \mathcal{S}$ such that A attacks B.*
(2) $\mathcal{S}$ defends *A iff $\mathcal{S}$ attacks each argument that attacks A. The set of arguments that $\mathcal{S}$ defends is denoted by $\mathcal{F}(\mathcal{S})$. $\mathcal{F}$ is called* the characteristic function of an argumentation framework *$\langle Arg_P, Arg_C, Atts\rangle$.*
(3) $\mathcal{S}$ is admissible *iff $\mathcal{S}$ is conflict-free and defends all the elements.*

There are several definitions of acceptability, and different extensions exist for each acceptability.

Definition 15 (extensions) *Let $\mathcal{E} \subseteq Arg_P \cup Arg_C$.*
(1) $\mathcal{E}$ is a preferred extension *iff $\mathcal{E}$ is maximal w.r.t. $\subseteq$ admissible set.*
(2) $\mathcal{E}$ is a grounded extension *iff $\mathcal{E}$ is the least fixed point w.r.t. $\subseteq$ of the characteristic function $\mathcal{F}$.*
(3) $\mathcal{E}$ is a stable extension *iff $\mathcal{E}$ is conflict-free and attacks each argument that is not included in $\mathcal{E}$.*

The following relations hold among these extensions.

Proposition 2 *[11, 10] (1) There is at least one preferred extension, always a unique extension, and there may be zero, one, or many stable extensions.*
(2) If there is no cyclic structure in an argumentation framework, then there is a unique stable extension, and the three extensions coincide.

4.2 Dynamic extension

Here, for simplicity, we assume that there is no threat whose resource and destination belong to different candidate subtrees.

For an argumentation framework $AF = \langle Arg_P, Arg_C, Atts \rangle$, let T_{exec} be an execution tree for an execution *exec*. Let Arg'_P and Arg'_C be a set of P's and C's arguments in T_{exec}, respectively, and $Atts'$ be a set of attacks between these arguments. Then, T_{exec} is an argumentation tree for an argumentation framework $AF_{exec} = \langle Arg'_P, Arg'_C, Atts' \rangle$. We call such AF_{exec} *an argumentation framework for exec.*

Definition 16 (dynamic extension) *For an argumentation framework AF and its execution exec, let AF_{exec} be an argumentation framework for exec. Then, the preferred extension for AF_{exec} is said to be* dynamic extension for *exec* of AF*, and a set of all the dynamic extensions for executions of AF is said to be* the dynamic extension for AF.

Example 1 *In Figure 1(2), let $exec_1$ be an execution in which the left branch is executed first and $exec_2$ be an execution in which the right branch is executed first. Then, the argumentation framework for $exec_1$ is $AF_{exec_1} = \langle \{P_0, C_1, P_1, C_2, P_2\}, \{(C_1, P_0), (P_1, C_1), (C_2, P_0), (P_2, C_2)\} \rangle$, and the dynamic extension for $exec_1$ is $\mathcal{E}_{exec_1} = \{P_0, P_1, P_2\}$. Those for $exec_2$ are $AF_{exec_2} = \langle \{P_0, C_2\}, \{(C_2, P_0)\} \rangle$ and $\mathcal{E}_{exec_2} = \{C_2\}$, respectively. The dynamic extension for AF is $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2\}$.*

Definition 17 (minimal dynamic extension) *Let $\mathcal{E}_1 \ldots, \mathcal{E}_n$ be dynamic extensions for executions of AF. If $\mathcal{E}_i$ such that $\mathcal{E}_i \subset \mathcal{E}_j$ $(i \neq j)$ does not exist, then $\mathcal{E}_j$ is said to be* a minimal dynamic extension for AF.

The following subsections present dynamic extensions for each pattern of an initial argumentation tree.

4.3 Case in which no threat exists

First, we explain the case in which both Arg_P, Arg_C contain no threats.

Let $AF = \langle Arg_P, Arg_C, Atts \rangle$ be an argumentation framework and T be a candidate subtree of an initial argumentation tree for AF.

Let $\mathcal{D}_P$ and $\mathcal{D}_C$ be sets of branches in which the leaf nodes are P's nodes and C's nodes, respectively. Let $|\mathcal{D}_P| = n$ and $|\mathcal{D}_C| = m$. APKC2 proceeds by selecting a branch with an executable node from $\mathcal{D}_P \cup \mathcal{D}_C$ in an arbitrary order.

Considering that APKC2 proceeds by turn of P and C, we can classify argumentation trees into three types by focusing on the leaf nodes.

(1) All leaf nodes are P's nodes.

In this case, all branches $D_P^1, \ldots, D_P^n$ $(1 \leq j \leq n)$ in $\mathcal{D}_P$ can be executed in an arbitrary order. Then, dynamic extensions for all executions consist of all of P's nodes appearing in T, and they coincide with each other. Therefore, a dynamic extension for AF is a singleton that is a set including the root node.

(2) All leaf nodes are C's nodes.

In this case, only one branch D_C^i $(1 \leq i \leq m)$ in $\mathcal{D}_C$ can be executed. Then, a dynamic extension for each execution $\mathcal{E}_i$ consists of all of C's nodes in D_C^i. Therefore, a dynamic extension for AF is $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_m\}$ Each $\mathcal{E}_i$ contains only C's nodes and is a minimal dynamic extension. Moreover, their intersection is an empty set.

(3) Leaf nodes consists of both P's nodes and C's nodes.

In this case, after executing several branches $D_P^1, \ldots, D_P^k$ $(1 \leq k \leq n)$ in $\mathcal{D}_P$, a branch D_C^i $(1 \leq i \leq m)$ in $\mathcal{D}_C$ is executed. Then, a dynamic extension for each execution $\mathcal{E}_{ik}$ consists of all of C's nodes in $\mathcal{D}_C$ and all of P's nodes in $D_P^1 \cup \ldots \cup D_P^k$ that are not in $\mathcal{D}_C$, irrespective of the execution order of $D_P^1, \ldots, D_P^k$. Therefore, a dynamic extension for AF is $\mathcal{E} = \mathcal{E}_1 \cup \ldots \cup \mathcal{E}_m$ where each $\mathcal{E}_i$ is a set of extentions for all possible combinations of selecting k elements from $\mathcal{D}_P$.

Proposition 3 *For the above three cases, the number of minimal dynamic extensions can be defined as follows.*
(1) There exists a unique minimal dynamic extension.
(2) There exist $|\mathcal{D}_C|$ number of minimal dynamic extensions.
(3) There exist $|\mathcal{D}_C|$ number of minimal dynamic extensions.

Moreover, since an argumentation tree P wins is only the case (1), the following property holds.

Proposition 4 *If there is no threat, there is no case in which P wins in one execution and loses in another execution.*

4.4 Computing update of dynamic extension for an execution

For an argumentation framework $AF = \langle Arg_P, Arg_C, Atts\rangle$, if at least one of Arg_P and Arg_C contains a threat, the threat affects the outcome of an argumentation. We can explore the effect in detail by investigating how the dynamic extension of an argumentation with a threat and the dynamic extension of an argumentation without a threat differ in each pattern of the initial argumentation tree.

First, we will set out the rules for computing a dynamic extension for an execution tree and the properties it satisfies.

To simplify the problem, we can assume that an initial argumentation tree has only two branches: D_1, which includes a threat resource, and D_2, which includes a threat destination. The procedure shown here is applicable to an arbitrary argumentation tree insofar as it has no threat over multiple candidate subtrees.

The notations used are presented below.

T_0: a candidate subtree of initial argumentation tree for AF
$exec_1$: execution along the order $D_1 D_2$
$exec_2$: execution along the order $D_2 D_1$
T_i: execution tree for $exec_i$
$\mathcal{E}_i$: dynamic extension for $exec_i$
$\mathcal{E}$: the dynamic extension for AF

For a given execution tree for an execution *exec*, we can construct a dynamic extension $\mathcal{E}_{exec}$ for *exec*. For each node, we determine whether it is included in a dynamic extension by exploring the execution tree from the leaf nodes in a bottom-up manner using the following rule.

Judgment for inclusion of a dynamic extension for each node
(1) A leaf node is in $\mathcal{E}_{exec}$.
(2) The node whose all child nodes are not in $\mathcal{E}_{exec}$ is in $\mathcal{E}_{exec}$.
(3) The node whose child nodes include at least one node that is in $\mathcal{E}_{exec}$ is not in $\mathcal{E}_{exec}$.

Proposition 5 *Let T_1 and T_2 be execution trees for executions $exec_1$ and $exec_2$ in AF, respectively, and $\mathcal{E}_1$ and $\mathcal{E}_2$ be dynamic extensions for $exec_1$ and $exec_2$, respectively. If T_1 is a subtree of T_2 such that $T_1 \neq T_2$, then $\mathcal{E}_1 \subset \mathcal{E}_2$.*

Proof) Let D_1 and D_2 be branches in an argumentation tree for AF. Also, let $exec_1$ be an execution in which branches are executed in the order of D_1, D_2, and let $exec_2$ be an execution in the order of D_2, D_1. Assume that the number of nodes included in D_1 except for the root node is even. Then, the leaf node of D_1 is P's node. Therefore, after D_1 is executed, D_2 should be executed. In this case, T_1 should not be a subtree of T_2. Then, the number of nodes included in D_1 is odd. Therefore, $\mathcal{E}_2$ does not include the root node, but includes its child nodes. If at least one child node is judged to be included in $\mathcal{E}$, then its parent node is judged to be not included in $\mathcal{E}$. Therefore, $\mathcal{E}_1$ does not include the root node. Moreover, for any node N in D_1 other than the root node, it is obvious that if $N \in \mathcal{E}_1$, then $N \in \mathcal{E}_2$ holds. Thus, $\mathcal{E}_1 \subset \mathcal{E}_2$.

Next, we examine how extensions are changed by the effect of a threat and investigate their relationships and properties.

A new node N is added either to the leaf node or a mid-node of a branch D. by a threat, and a maximal admissible set for D is changed. All upper nodes in D including N and the root node are judged using the above rule of *judgment for inclusion of a dynamic extension for each node*, which results in the outcome of the revised maximal admissible set. This outcome is denoted by $UPDATE(D)$.

4.5 The effect of a threat from P to P

Next, we focus on the case in which a threat from P to P is contained in AF.

Let P_r and P_d be a threat resource and a threat destination, respectively, and let C' be a new node generated by this threat.

We can apply the following notation.

T_0: a candidate subtree of the initial argumentation tree for AF
D_1, D_2: branches in T_0
$exec_1$: execution along the order $D_1 D_2$
$exec_2$: execution along the order $D_2 D_1$
T_i: execution tree for $exec_i$ without a threat
$\mathcal{E}$: the dynamic extension for AF without a threat
$\mathcal{A}_i$: maximal admissible set of which each element corresponds to a node in D_i
T'_i: execution tree for $exec_i$
$\mathcal{E}'_i$: dynamic extension for $exec_i$
$\mathcal{E}'$: the dynamic extension for AF

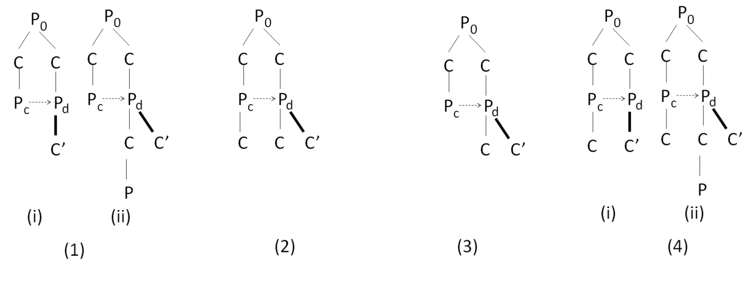


Fig. 4. The effect of P's threat

We can derive $\mathcal{E}'$ from T_0 and $threat(P_r, P_d)$. First, we compare execution trees with and without a threat.

(P1) All leaf nodes in $\boldsymbol{T_0}$ are P (Figure 4(1))

$T_1' = T_1 \cup \{C'\}$. $T_2' = T_2 \cup \{C'\}$.

Note that $T_1 = T_2$ holds. T_2' is the same as T_1' due to the suspend/resume mechanism.

Let P_l be the lowest node that belongs to both D_1 and D_2, and $uppereq(P_l)$ denote the nodes upper than or equivalent to P_l.

There may be two cases of extensions, depending on the position of P_d.
(i) P_d is a leaf node.
C' is added as a leaf of D_2

$\mathcal{E}_1' = \mathcal{A}_1 \backslash uppereq(P_l) \cup UPDATE(D_2)$. $\mathcal{E}_2' = \mathcal{A}_1 \backslash uppereq(P_l) \cup UPDATE(D_2)$.

$\mathcal{E}' = \{\mathcal{E}_1'\}$.
(ii) P_d is a mid-node.
C' is added as a child node of P_d to generate a new branch D_3.

$\mathcal{E}_1' = \mathcal{A}_1 \backslash uppereq(P_l) \cup UPDATE(D_3)$. $\mathcal{E}_2' = \mathcal{A}_1 \backslash uppereq(P_l) \cup UPDATE(D_3)$.

$\mathcal{E}' = \{\mathcal{E}_1'\}$.

(P2) All leaf nodes in $\boldsymbol{T_0}$ are C (Figure 4(2))

C' is added as a child node of P_d, and a new branch D_3 is added.

As for the dynamic extension, $\mathcal{E}' = \mathcal{E}$.

(P3) $\boldsymbol{D_1}$'s leaf node is P, $\boldsymbol{D_2}$'s leaf node is C (Figure 4(3))

C' is added as a child node of P_d, and a new branch D_3 is added. Let $lower(P_d)$ denote the nodes lower than P_d in D_2. A new execution $exec_3$ is generated. Let P_l be the lowest node that belongs to both D_1 and D_2, and $uppereq(P_l)$ denote the nodes upper than or equivalent to P_l.

$T_1' = T_1$. $T_2' = T_2$. $T_3' = T_1 \setminus lower(P_d) \cup \{C'\}$.

In this case, the dynamic extensions are as follows.

$\mathcal{E}_1' = \mathcal{A}_1 \cup \mathcal{A}_2$. $\mathcal{E}_2' = \mathcal{A}_2$. $\mathcal{E}_3' = \mathcal{A}_1 \setminus uppereq(P_l) \cup UPDATE(D_3)$.

$\mathcal{E}' = \{\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3'\}$.

Note that the selected branch must be executed as far as possible, and a node in the other branch cannot be executed at any time.

(P4) $\boldsymbol{D_1}$'s leaf node is P, $\boldsymbol{D_2}$'s leaf node is C (Figure 4(4))

There are two possible cases, depending on the position of P_d: (i) P_d is a leaf node and (ii) P_d is a mid-node.

With regard to the dynamic extension, $\mathcal{E}' = \mathcal{E}$ in either case.

4.6 The effect of a threat from C to C

Next, we focus on the case in which a threat from C to C is contained in AF.

Let C_r and C_d be a threat resource and threat destination, respectively, and let P' be a new node generated by this threat.

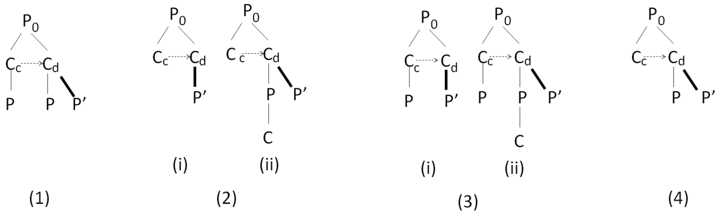


Fig. 5. The effect of C's threat

(C1) All the leaf nodes in $\boldsymbol{T_0}$ are P (Figure 5(1))

P' is added as a child node of C_d to generate a new branch D_3. Let $lower(C_d)$ denote the nodes lower than C_d in D_2. Let P_l be the lowest node that belongs to both D_1 and D_2, and $uppereq(P_l)$ denote the nodes upper than or equivalent to P_l.

$T_1' = T_1$. $T_2' = T_2$. $T_3' = T_1 \setminus lower(P_d) \cup \{C'\}$.

In this case, a new execution $exec_3$ is generated, and the dynamic extensions are as follows.

$\mathcal{E}_1' = \mathcal{E}_2' = \mathcal{A}_1 \cup \mathcal{A}_2$. $\mathcal{E}_3' = \mathcal{A}_1 \setminus uppereq(P_l) \cup UPDATE(D_3)$.

$\mathcal{E}' = \{\mathcal{E}_1', \mathcal{E}_3'\}$

(C2) All leaf nodes in $\boldsymbol{T_0}$ are C (Figure 5(2))

There are two possible cases, depending on the position of P_d: (i) C_d is a leaf node, and (ii) C_d is a mid-node.

With regard to the dynamic extension, $\mathcal{E}' = \mathcal{E}$ in either case.

(C3) $\boldsymbol{D_1}$'s leaf node is P, $\boldsymbol{D_2}$'s leaf node is C (Figure 5(3))

There are two possible cases, depending on the position of P_d.

Let P_l be the lowest node that belongs to both D_1 and D_2, and $uppereq(P_l)$ denote the nodes upper than or equivalent to P_l.

(i) P_d is a leaf node.

C' is added as a leaf of D_2

$T_1' = T_1$. $T_2' = T_2$. $T_3' = T_1 \setminus lower(C_d) \cup \{P'\}$.

With regard to the dynamic extensions,

$\mathcal{E}_1' = \mathcal{A}_1 \setminus uppereq(P_l) \cup UPDATE(D_2)$. $\mathcal{E}_2' = \mathcal{A}_2$.

$\mathcal{E}' = \{\mathcal{E}_1', \mathcal{E}_2'\}$.

(ii) P_d is a mid-node.

C' is added as a child node of P_d to generate a new branch D_3.

$T_1' = T_1 \cup \{P'\}$. $T_2' = T_2$. $T_3' = T_1 \setminus lower(C_d) \cup \{P'\}$.

With regard to the dynamic extensions,
$\mathcal{E}'_1 = \mathcal{A}_1 \setminus uppereq(P_l) \cup UPDATE(D_3)$. $\mathcal{E}'_2 = \mathcal{A}_2$.
$\mathcal{E}'_3 = \mathcal{A}_1 \setminus uppereq(P_l) \cup \mathcal{A}_2 \cup UPDATE(D_3)$.
$\mathcal{E}' = \{\mathcal{E}'_1, \mathcal{E}'_2, \mathcal{E}'_3\}$.

(C4) D_1's leaf node is C, D_2's leaf node is P (Figure 5(4))
P' is added as a child node of C_d, and a new branch D_3 is added.
With regard to the dynamic extension, $\mathcal{E}' = \mathcal{E}$.

Example 2 *Consider the example shown in Figure 6.*

Figure 6(1) shows an argumentation tree T_0 without a threat. The maximal admissible sets of each branch are $\mathcal{A}_1 = \{P_1\}$ and $\mathcal{A}_2 = \{C_2, C_3\}$, respectively. T_1 and T_2 show execution trees for executions without a threat (Figure 6(3)). The former corresponds to an execution in which the left branch is executed first, while the latter corresponds to an execution in which the right branch is executed first.

In contrast, Figure 6(2) shows an argumentation tree with a threat from C_1 to C_3 to generate a new node P'. This is an example of case (C3)(i). T'_1 and T'_2 show execution trees (Figure 6(4)). $\mathcal{E}'_1$ is obtained by updating D_2. $UPDATE(D_2) = \{P_0, P_2, P'\}$.

Therefore, the dynamic extension is $\mathcal{E}' = \{\mathcal{E}'_1, \mathcal{E}'_2\}$, where $\mathcal{E}'_1 = \{P_0, P_1, P_2, P'\}$, $\mathcal{E}'_2 = \{C_2, C_3\}$.

4.7 Properties

It is not sufficient simply to consider updating each branch when changes in extensions are considered. The interesting point is that even if a new node is added by a threat, it does not always affect the extension. This is due to the constraint of turn keeping and the fact that a new branch is not executed until all the executable nodes in the current branch are executed.

The following relation holds between a dynamic extension and the win/loss of an argumentation.

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be dynamic extensions for executions for an argumentation framework AF and $\mathcal{E}$ be a dynamic extension for AF.

1. If each $\mathcal{E}_i$ consists of only P's arguments, P dynamically wins. In this case, $\mathcal{E}_1, \ldots, \mathcal{E}_n$ coincide and include the root node.
2. If each $\mathcal{E}_i$ consists of only C's arguments, every one of P's arguments in an argumentation framework is attacked in any execution.
3. If each $\mathcal{E}_i$ consists of both P's and C's arguments, P loses the argumentation. In this case, $\mathcal{E}_i$ does not contain the root node, and a minimal dynamic extension that consists of all of C's nodes exists.

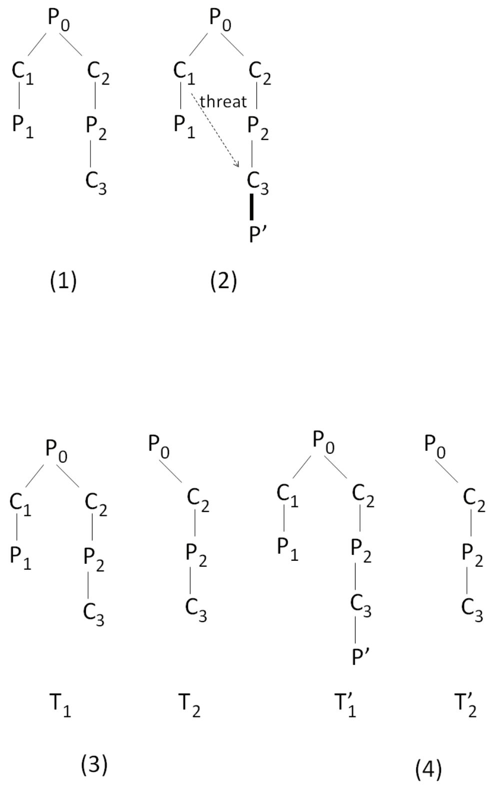


Fig. 6. Example of changing extensions

5 Related Works

The abstract argumentation framework proposed by Dung does not put orders of arguments and not include the idea of win/loss of an argumentation. It is represented as a graph structure in which nodes and edges correspond to arguments and attacks, respectively. On the other hand, in several works on dialogue or dialect, argumentation was represented in a tree form that identified the proposal statement as the root node, gave an order to arguments, and defined a concept of win/loss of an argumentation. Amgoud et. al regarded an argumentation as a dialogue game that could be represented as an AND/OR tree and gave a semantics to indicate whether the argument corresponding to the root node was accepted [1]. They defined a win as that situation where a solution subtree exists in which all the leaves are P's nodes. Dunne proposed a "dispute tree" on which subsequent execution of all branches is considered [10]. However, the revision of an agent's knowledge base was not considered there, allowing presented arguments to add new information to the opponent's knowledge base. Garc a et al. also represented an argumentation framework as a tree, called a dialectical tree [13]. An argumentation formalism was given based on defeasible logic programming (DeLP) to decide between contradictory goals. They presented an algorithm to judge whether an argument corresponding to the root node is self-defendable. Such an argument is called "warranted." The win in argumentation in APKC2 is identical to the concept of "warranted." Later, Modgil proposed the Extended Argumentation Framework, an extension of an argumentation framework that introduced the concept of a meta-attack, that is, an attack to an attack, and discussed its semantics [16].

Moguillansky et al. considered the treatment of DeLP by an argumentation framework [17]. Their treatment made belief change theory suitable for an argumentation system based on DeLP. They gave an algorithm for judging which rules are selected from a given set of defeasible rules such that an argument corresponding to the root node is warranted. Their work can be considered as one handling argument theory change because an argumentation framework is changed depending on the set of rules that are selected. However, the aim of their work was to construct an argumentation framework that makes the root node warranted, not to consider the effect(s) of the execution of an algorithm. For this reason, they did not consider the timing of applying addition/deletion of rules. In contrast, in our dynamic argumentation framework, we introduce the concept of an execution tree and insist that the execution does create a new argument.

While in the approaches based on DeLP new arguments and attacks are determined by formulas included in the rules, Cayrol et al. investigated argument theory change at a more abstract level by treating only the addition of nodes in an argumentation graph [5]. They investigated how acceptable arguments are changed when an argument is added. The aim of their research was a formal analysis of changes to argumentation; the contents of the additional arguments and the reasons for their addition were beyond its scope. Cobo et al. proposed an argumentation framework in which available arguments change depending on time intervals [8]. In their work, these intervals are given in advance, they did not consider the mechanism by which an argument causes to generate a new argument. In contrast, we focus specifically on the effect of knowledge gained from presented arguments, which is essential in actual argumentation.

Several studies have been conducted on argumentation semantics. Dung provided a semantics for a given abstract argumentation framework based on acceptability [11]. He defined several acceptable sets, depending on the range of strength against an attack. Coste-Morquis et al. argued that it is controversial to include both agents' arguments in an extension because this would indicate an indirect attack [9]. They defined a new semantics, called "prudent semantics," which does not allow such controversial cases, and compared this with Dung's semantics. Other semantics have also been proposed, such as ideal semantics [12], semi-stable semantics [6], and others. Boroni et al. compared these types of semantics from the viewpoint of skepticism [3].

All these semantics involved argumentation systems from a static viewpoint, whereas our proposed semantics is suitable for a dynamic argumentation system.

6 Conclusion

In this paper, we defined a new semantics that can fit a dynamic argumentation framework. We defined a dynamic extension for each execution of an argumentation and defined the dynamic extension for an argumentation framework as a set of these extensions. Additionally, we discussed how these extensions are changed by the effect of a threat and investigated their relationships and properties. Interestingly, a threat does not always affect the outcome of an extension it changes. Although we restricted our analysis to the case in which a threat exists in only a single candidate subtree, it should be straightforward to extend the semantics to include cases in which a threat occurs over multiple candidate subtrees. We are currently formalizing this extended version.

We are also investigating the relationship of this system to the JC algorithm that we proposed previously [20]; this is an algorithm for judging the win/loss of an argumentation.

References

1. L.Amgoud, S.Parsons, and N.Maudet: Arguments, dialogue, and negotiation, ECAI2000, pp.338-342, 2000.
2. L.Amgoud and S.Vesic: Repairing preference-based argumentation frameworks, IJCAI2009, pp.665-670, 2009.
3. P.Baroni and M.Giacomin: Comparing Argumentation Semantics with Respect to Skepticism. Symbolic and Quantitative Approaches to Reasoning with Uncertainty, pp.210-221, LNCS4724, 2007.
4. T.Bench-Capon and P.Dunne: Argumentation in artificial intelligence, Artificial Intelligence, 171, pp.619-641, 2007.

5. C.Cayrol, F.D.de St-Cyr, and M-C Lagasquie-Shiex: Change in Abstract Argumentation Frameworks: Adding an Argument. Journal of Artificial Intelligence Research, 38, pp.49-84, 2010.
6. M.Caminada: Semi-stable semantics. In COMMA2006, pp.121-130, 2006.
7. C.I.Chesnevar, A.Maguitman and R.Loui: Logical models of argument. ACM Computing Surveys, 32(4), pp.337-383, 2005.
8. M.L.Cobo, D.C.Martinez and G.R.Simari: An approach to timed abstract argumentation. NMR2010, Workshop on Argument, Dialog and Decision.
9. S.Coste-Marquis, C.Devred and P.Marquis: Prudent semantics for argumentation frameworks. In ICTAI2005, pp.568-572, 2005.
10. P.E.Dunne and T.J.M.Bench-Capon: Coherence in finite argument system. Artificial Intelligence, 141(1-2), pp.187-203, 2002.
11. P.M.Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77, pp.321-357, 1995.
12. P.M.Dung, P.Mancarella and F.Toni: A dialectic procedure for sceptical, assumption-based argumentation. In COMMA2006, pp.145-156, 2006.
13. A.García, and G.Simari: Defeasible logic programming: an argumentative approach. Theory and practice of logic programming, 4(1), pp.95-138, 2004.
14. A.García, C.Chesnevar, N.Rotstein, and G.Simari: An abstract presentation of dialectical explanations in defeasible argumentation, ArgNMR07, pp.17-32, 2007.
15. C.Hamblin: *Fallacies*, Methuen, 1970.
16. S.Modgil: Reasoning about preferences in argumentation frameworks, Artificial Intelligence, 173(9-10), pp.901-1040, 2009.
17. M.O.Moguillansky, et al.: Argument theory change applied to defeasible logic programming, AAAI2008, pp.132-137, 2008.
18. H.Prakken: Combining skeptical epistemic reasoning with credulous practical reasoning. COMMA 2006, pp.311-322, 2006.
19. K.Okuno and K.Takahashi: Argumentation system with changes of an agent's knowledge base, IJCAI2009, pp.226-232, 2009.
20. K.Okuno and K.Takahashi: Argumentation System Allowing Suspend/Resume of an Argumentation Line ArgMAS2010, pp.145-162, 2010.
21. I.Rahwan, and G.Simari (eds.): *Argumentation in Artificial Intelligence*, Springer, 2009.

Argumentation Patterns

Serena Villata[1] and Guido Boella[1] and Leendert van der Torre[2]

[1] Dipartimento di Informatica, University of Turin
{villata,boella}@di.unito.it
[2] Computer Science and Communication, University of Luxembourg
leendert@vandertorre.com

Abstract. Argumentation patterns are general reusable solutions to commonly occurring problems in the design of argumentation frameworks, such as the relation between claim and data in the Toulmin scheme. We introduce a formal approach for the semantics of argumentation patterns describing relationships and interactions among arguments, without instantiating the involved abstract arguments. Argumentation patterns are a multi-labeling of a set of arguments, together with constraints on this labeling. Constraints express the relations among the labels of the arguments of the pattern when they interact with other arguments. Moreover, we define argumentation patterns using a two sorted argumentation framework where focal arguments are distinguished from auxiliary arguments, and we show how to compute their semantics by reusing a semantics introduced by Dung. We show how patterns are applied to design conjunction and disjunction of arguments, accrual, proof standards, and second-order attacks.

1 Introduction

An argumentation framework [9] is composed by a set of elements called *arguments* and a binary relation over the arguments called *attack*. A core issue in argumentation theory is the relation between abstract arguments. In modelling argumentation frameworks, this relation has been investigated following different lines [3, 1, 5, 13, 8, 6]. In this paper, we propose to reuse software engineering ideas like patterns to investigate the relation between abstract arguments.

Our context deals with situations where argumentation frameworks are not generated from a knowledge base, but where the knowledge engineer has to directly design the arguments and attacks. In many cases, for the engineer is easier to reuse parts of existing frameworks, so a methodology for representing abstractions facilitating such reuse and for defining their meaning is needed. As methodology we introduce argumentation patterns. Argumentation patterns are visual descriptions for how to solve design problems of argumentation frameworks, that can be used in many situations.

Argumentation patterns are sets of arguments related to each other in such a way that they cannot be expressed directly with the basic attack relation. For example, assume that a modeler believes that the argument "Jones is not liable" is attacked if both "Jones has a contract" and "Jones has breached the contract" are acceptable. Then how to relate these three arguments such that this property holds? This is an instance of the conjunctive attack pattern: the former argument is attacked only if each of the latter is accepted. To express situations like this one the usual solution is to define extended argumentation frameworks, i.e., introducing a conjunctive attack relation with its own semantics. However, when more solutions must be put together it becomes problematic to unify everything into a single extended argumentation framework. This is why we propose patterns.

Our challenge to define argumentation patterns leads to the following research questions:

1. How to visualize argumentation patterns?
2. How to define argumentation patterns?
3. Which argumentation patterns can be identified in the literature?

First, the problem of finding a good visualization for argumentation patterns is not trivial. We mainly focus on the existing well-known visualizations such as boolean gates and transistors, and we provide the argumentative counterpart of such visualizations. In particular, we use the logic gate AND for visualizing the conjunctive pattern where each "input" argument needs to be acceptable to get the "output" argument unacceptable (or acceptable), and the OR gate for visualizing the disjunctive pattern, where at least one of the "input" arguments needs to be acceptable to get the "output" argument unacceptable (or acceptable). We introduce transistors to represent the second-order pattern where the collector is the attacking argument, the emitter is the attacked argument, and the base is the argument raising the second-order attack. Transistors can be composed to visualize the higher-order pattern. Transistors are used also to visualize part of the Toulmin scheme where the data is the collector, the claim is the emitter, and the warrant is the base.

Second, there are many ways to define argumentation patterns. Formal techniques are needed since the visualization may be ambiguous, and, in particular, not expressive enough to define how to combine argumentation patterns. Formal semantics is needed to define patterns and their use, and a formal syntax is needed to embed them in the overall argumentation framework. We consider two dimensions. First, the perspective of the designer, who knows the meaning of the pattern and how it behaves once inserted in a larger framework. We define an argumentation pattern as a set of arguments together with the specification of their behavior, which is not simply a set of attacks among the arguments of the pattern. We express the meaning of the pattern with a *multi-labeling* function and a set of propositional formulas called *constraints*. The multi-labeling shows the values assigned to the arguments in the pattern while the constraints express relations between these values. In particular, constraints allow to compute the labels of the arguments in the pattern, in case they are attacked by arguments outside the pattern. The multi-labeling, instead, restricts the possible labels of the arguments in the pattern, independently of attacks by arguments outside the pattern. The second dimension concerns the semantics of a framework which

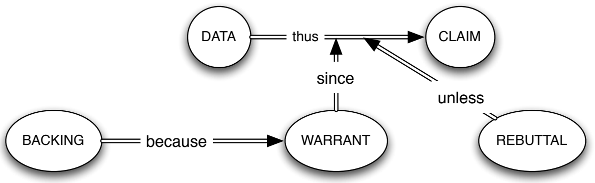


Fig. 1. The Toulmin scheme.

includes patterns. We could define an extended argumentation framework with an *ad hoc* semantics to cope with all the allowed patterns. Instead, we decide to flatten the patterns to abstract argumentation frameworks, by adding, when necessary, auxiliary arguments and suitable attacks. The flattening is driven by the definition of the pattern in terms of multi-labeling and constraints. The advantage of our solution is that it allows to reuse standard semantics, and to introduce further patterns without having to revise the semantics like in extended argumentation frameworks.

Third, the formal framework must be able to model argumentation patterns discussed in the literature. Fig. 1 visualizes the well-known Toulmin scheme [15]. The arrows represent unspecified relations between the elements, e.g., the warrant connects the data and the claim and it is supported with some backing, the rebuttal indicates circumstances in which the authority of the warrant has to be set aside. The framework has to be able to give a formal meaning to these relations – there may even be competing semantics of the Toulmin scheme, e.g., the claim is accepted only if the rebuttal is not accepted and if the warrant is supported by a backing.

Whereas most research in argumentation theory is driven by theoretical concerns, the work reported in this paper is driven by practical concerns. Even if ultimately arguments must be instantiated, in our experience of modeling argumentation [5, 3, 4], there is a need at the abstract level to define argumentation patterns. Our work raises also theoretical questions, but in this paper we restrict ourselves to concepts and examples.

This paper follows the research questions and it is organized as follows. Section 2 introduces the visualization, syntax and semantics of argumentation patterns, and how they are used. Section 3 defines patterns from the argumentation literature. Related work and conclusions end the paper.

2 Formal framework

2.1 Dung's abstract argumentation

We express Dung's [9] complete semantics of abstract argumentation using Jakobovits-Vermeir-Caminada's three valued labelings [11, 7], where an argument a can be labeled *in*, *out* or *undecided*. To define the meaning of patterns, we must be able to express whether arguments are *in*, *out* or *undecided*, and which is the label of an argument given the label of other arguments.

We write the fact that an argument can have one of the three labels by means of propositions $a^{\in}$, $a^{\notin}$, and $a^{?}$, meaning, respectively, that argument a is *in*, *out* or *undecided*. Given this notation we can express the relation between labelings and extensions in the following way. A labeling corresponds to the extension $\{a \mid a^{\in}\}$, and given an extension $E \subseteq A$ of argumentation framework $\langle A, \rightarrow \rangle$, the corresponding labeling is given by $a^{\in}$ iff $a \in E$, $a^{\notin}$ iff $a \notin E$ and $\exists b \in E$ such that $b \rightarrow a$, and $a^{?}$ otherwise. A simple example to start with is the equivalence between two arguments which can be expressed as $a^{\in} \equiv b^{\in} \wedge a^{\notin} \equiv b^{\notin}$. We write $\Rightarrow$ for material implication.

Definition 1 (Complete semantics). *Let U be a set of arguments called the universe of arguments. For any finite set of arguments $A \subseteq U$, a three valued labeling function $l : A \rightarrow \{\in, \notin, ?\}$ is a complete function that partitions a set of arguments into the* in *($\in$),* out *($\notin$) and* undecided *($?$) arguments. An acceptance function ϵ is a function that associates with every argumentation framework $\langle A, \rightarrow \rangle$ with $A \subseteq U$ and $\rightarrow \subseteq A \times A$, the set of three valued labelings of A satisfying the following conditions:*

- $\forall a, b \in A : a \rightarrow b \Rightarrow \neg(a^{\in} \wedge b^{\in})$*: an extension is conflict free.*
- $\forall b \in A : b^{\in} \Leftrightarrow \forall a : a \rightarrow b \Rightarrow a^{\notin}$*: an argument is in, if and only if all its attackers are out.*
- $\forall b \in A : b^{\notin} \Leftrightarrow \exists a : a \rightarrow b \wedge a^{\in}$*: an argument is out, if and only if at least one of its attackers is in.*

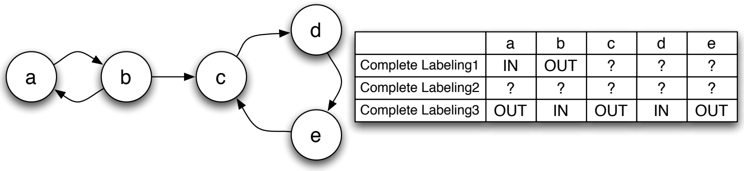
We call these labelings the complete labelings of argumentation framework $\langle A, \rightarrow \rangle$.

The following example due to Caminada [7] illustrates the complete semantics and our notation.

Example 1 (Two cycles). Fig. 2 visualizes the argumentation framework $\langle A, \rightarrow \rangle$ with $A = \{a, b, c, d, e\}$ and $\rightarrow = \{a \rightarrow b, b \rightarrow a, b \rightarrow c, c \rightarrow d, d \rightarrow e, e \rightarrow c\}$,where a="Jones is a spy", b="Jones is not a spy" c="Mary says that Jones lies", d="Jones says that Harry lies," and e="Harry says that Mary lies." This figure must be read as follows: a circle visualizes an argument, and an arrow visualizes an attack. The complete semantics is given by three labelings: $a^{\in} \wedge b^{\notin} \wedge c^{?} \wedge d^{?} \wedge e^{?}$, $a^{?} \wedge b^{?} \wedge c^{?} \wedge d^{?} \wedge e^{?}$, $a^{\notin} \wedge b^{\in} \wedge c^{\notin} \wedge d^{\in} \wedge e^{\notin}$. Other semantics return other labelings, for example the grounded semantics returns only $a^{?} \wedge b^{?} \wedge c^{?} \wedge d^{?} \wedge e^{?}$, the maximal number of undecided arguments, whereas the preferred or stable semantics only return $a^{\notin} \wedge b^{\in} \wedge c^{\notin} \wedge d^{\in} \wedge e^{\notin}$, the minimal number of undecided arguments.

2.2 Semantics of argumentation patterns

An argumentation pattern is a multi-labeling (i.e., a set of labels for each argument) of a set of arguments, together with propositional constraints on the labeling. Roughly, the multi-labeling contains the labelings of the arguments



	a	b	c	d	e
Complete Labeling1	IN	OUT	?	?	?
Complete Labeling2	?	?	?	?	?
Complete Labeling3	OUT	IN	OUT	IN	OUT

Fig. 2. Two cycles (Example 1)

when none of the arguments of the pattern is attacked by arguments not in the pattern, and the constraints represent an invariant expressing the properties which always hold between the labels of the arguments of the pattern, regardless whether the arguments are attacked by other arguments or not. The constraints are expressed in terms of propositions $x^{\in}$, $x^{\notin}$, and $x^{?}$ for all $x \in A$ which represent if an argument is *in*, *out* or *undecided*. Note that this is a possible choice, but other choices are possible too, as we discuss in the conclusion. One criterion to decide is the expressive power of the pattern language.

Definition 2 (Argumentation pattern). *An n-ary argumentation pattern is a triple $\langle A, M, C\rangle$ where $A \subseteq U$ is a sequence of n arguments, $M : A \rightarrow 2^{\{\in,\notin,?\}}$ a function from the arguments to the powerset of the labels (called a multi-labeling) and C is a propositional formula on signature $x^{\in}$, $x^{\notin}$, and $x^{?}$ for all $x \in A$. The labelings of an argumentation pattern are the labelings where the label of each argument is one of its multi-labels, and that satisfy the constraints of the pattern.*

At first sight it may seem that the multi-labeling is a constraint too, namely the constraint that the label of the argument contains one of the values of the multi-label. However, the multi-label expresses the values the arguments can have when the pattern is not attacked by other arguments, and the constraints express the relations between the values of the arguments, whether they are attacked by other arguments or not. Both the multi-labeling and the constraints are needed for the case in which patterns are used in a larger argumentation framework, and when they are combined with other argumentation patterns, as explained in Examples 7 and 8.

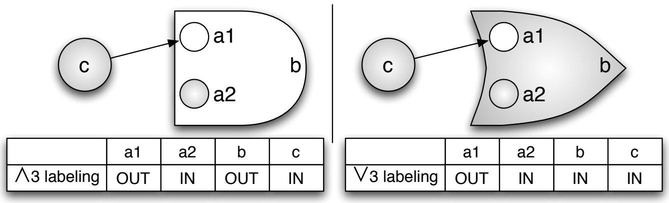
Example 2 illustrates the definition of semantics of argumentation patterns by maybe the simplest patterns, namely conjunction and disjunction. We express the patterns as *patternName: constraints*, e.g., $\wedge_{n+1}(a_1, \ldots, a_n, b)$ is the name of the conjunction pattern.

Example 2 (Conjunction and disjunction). Both patterns are defined by multi-labeling $M(a_1) = \ldots = M(a_n) = M(b) = \{\in\}$, together with respectively:

$$\wedge_{n+1}(a_1, \ldots, a_n, b) : (a_1^{\in} \wedge \ldots \wedge a_n^{\in} \Leftarrow b^{\in}) \wedge (a_1^{\notin} \vee \ldots \vee a_n^{\notin} \Rightarrow b^{\notin})$$

$$\vee_{n+1}(a_1, \ldots, a_n, b) : (a_1^{\in} \vee \ldots \vee a_n^{\in} \Leftarrow b^{\in}) \wedge (a_1^{\notin} \wedge \ldots \wedge a_n^{\notin} \Rightarrow b^{\notin})$$

Fig. 3 visualizes a_1="Jones has a contract", a_2="Jones has breached the contract" and b="Jones is liable" with $\wedge_3$ and $\vee_3$, together with an additional



	a1	a2	b	c
⋀3 labeling	OUT	IN	OUT	IN

	a1	a2	b	c
⋁3 labeling	OUT	IN	IN	IN

Fig. 3. Conjunction and disjunction (Example 2)

argument c="Jones did not sign the contract" attacking a_1. In the figures, we visualize in grey accepted arguments. In Fig. 3, the whole component is called b, and the two incoming ports are called a_1 and a_2. Attacking a_1 is attacking the port of the component. In the former case we have that Jones is not liable, because $a_1^{\notin} \wedge a_2^{\in} \wedge b^{\notin} \wedge c^{\in}$ is the unique labeling, in the latter case we have that Jones is liable, because $a_1^{\notin} \wedge a_2^{\in} \wedge b^{\in} \wedge c^{\in}$ is the unique labeling. Notice that the labels of the pattern are different from the multi-labeling defined above. The multi-labeling assigns to each argument the label $\in$, but the existence of argument c attacking the argument of the pattern a_1, leads to a change in the labels. Given the presence of this external argument c we cannot assign the label defined by the multi-labeling to the arguments of the pattern, thus we have to satisfy the constraints posed by the patterns.

We have to underline that a multi-label is assigned to an argument if there is the presence of a cycle in the pattern, otherwise, as in the example above, only a single label is assigned. Example 3 illustrates that some extended argumentation frameworks can also be represented by argumentation patterns, by defining second-order attacks as an argumentation pattern. Roughly, second-order attacks "disconnect" the attack relations among the arguments.

Example 3 (Second-order attack). The pattern is given by the multi-labeling $M(a) = M(b) = M(c) = \{\in\}$, because the attack relation among a and c is attacked by the second order attack, together with the constraint

$$\#_3(a, b, c) : (a^{\notin} \vee b^{\in} \Leftarrow c^{\in}) \wedge (a^{\in} \wedge b^{\notin} \Rightarrow c^{\notin})$$

Fig. 4 visualizes the second-order attack as a transistor where the collector is the attacking argument, the emitter is the attacked argument, and the base is the argument raising the second-order attack. The arguments can be read as a="Jones was honored at a special ceremony", b="Intelligence wants to study Jones's behaviour" and c="Jones is a spy".

We now have to consider the behavior of the pattern when it belongs to a wider argumentation framework. For example, consider what happens when argument d attacks the different arguments composing the pattern. We have the following possible situations $a^{\in} \wedge b^{\in} \wedge c^{\in} \wedge d^{\in}$, and respectively $d \rightarrow a$, $a^{\notin} \wedge b^{\in} \wedge c^{\in} \wedge d^{\in}$, $d \rightarrow b$, $a^{\in} \wedge b^{\notin} \wedge c^{\notin} \wedge d^{\in}$, $d \rightarrow a \wedge d \rightarrow b$, $a^{\notin} \wedge b^{\notin} \wedge c^{\in} \wedge d^{\in}$. So we have $c^{\notin}$ only if $a^{\in}$ and thus d does not attack a, together with $b^{\notin}$ and thus d attacks b.

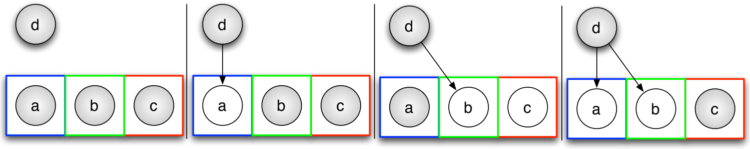


Fig. 4. Second-order attack (Example 3)

2.3 Syntax of argumentation patterns

In the previous section we described how to express the meaning of an argumentation pattern for the designer. It remains to define the structure of the patterns when they appear in an argumentation framework. Rather than proposing an extended argumentation framework with an *ad hoc* semantics to cope with all the allowed patterns, we decide to flatten the argumentation patterns to abstract argumentation frameworks, by adding auxiliary arguments and suitable attacks. In this paper, we are interested in argumentation patterns that can be expressed as a two sorted argumentation framework, distinguishing between auxiliary and focal arguments.

Definition 3 (Two sorted AF). *A two sorted argumentation framework is a triple $\langle A, B, \rightarrow \rangle$ with $A \subseteq B \subseteq U$ and $\rightarrow \subseteq B \times B$, where A are called the focal arguments, and $B \backslash A$ the auxiliary arguments.*

We have to consider two directions. First, an argumentation pattern can be flattened into a two sorted AF by respecting the multi-labeling and constraints. Second, a two sorted argumentation framework induces an argumentation pattern. This direction is more complicated, since we have to abstract away the auxiliary arguments. Moreover, given the constraints on the two sorted AF corresponding to the attack relations, we have to abstract away the propositions concerning the labeling of auxiliary arguments. This abstraction process means that we have to forget the variables referring to arguments which we abstract away, in the technical sense of forgetting defined by Lang and Marquis [12]. Generally, it is sometimes the case that ignoring a small set of propositional atoms of the formulas from an inconsistent set renders it consistent. Lang and Marquis [12] define a framework for reasoning from inconsistent propositional bases, using forgetting as a basic operation for weakening formulas. Belief bases are viewed as finite vectors of propositional formulas, conjunctively interpreted. Forgetting a set X of atoms in a formula consists in replacing it by its logically strongest consequence which is independent of X, in the sense that it is equivalent to a formula in which no atom from X occurs. The key notion is that of recoveries, which are sets of atoms whose forgetting enables restoring consistency. Forgetting is defined by Lang and Marquis [12] as follows. For more details about forgetting, see [12].

Definition 4 (Forgetting [12]). *Let ϕ be a formula from $PROP_{PS}$ and $V \subseteq PS$. The forgetting of V in ϕ, noted $\exists V.\phi$, is a formula from $PROP_{PS}$ that is inductively defined up to logical equivalence as follows:*

- $\exists \emptyset.\phi \equiv \phi$;
- $\exists \{x\}.\phi \equiv \phi_{x \leftarrow 0} \vee \phi_{x \leftarrow 1}$;
- $\exists (\{x\} \cup V).\phi \equiv \exists V.(\exists \{x\}.\phi)$;

where $PROP_{PS}$ denotes the propositional language built up from a finite set PS of atoms, the Boolean constants $\top$ and $\bot$, and the standard connectives and $\phi_{x \leftarrow 0}$ (resp. $\phi_{x \leftarrow 1}$) denotes the formula obtained by replacing in ϕ every occurrence of symbol x by $\bot$ (resp. $\top$).

Definition 5. *The argumentation pattern $\langle A, M, C\rangle$ induced by the two sorted argumentation framework $\langle A, B, \rightarrow\rangle$ is given by the constraint that takes the conjunction of the constraints given in Definition 1 for the auxiliary arguments, i.e.:*

$$\forall b \in B \setminus A : b^{\in} \Leftrightarrow \forall a : a \rightarrow b \Rightarrow a^{\notin}$$

$$\forall b \in B \setminus A : b^{\notin} \Leftrightarrow \exists a : a \rightarrow b \wedge a^{\in}$$

and then forgetting the variables referring to the arguments from which we have abstracted away.

The following two examples illustrate how the argumentation patterns for conjunction, disjunction and second-order attacks are induced by the two sorted argumentation framework.

Example 4 (Conjunction and disjunction).
AND^{n+1}=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a_1, \ldots, a_n, b\}, B = A \cup \{x_1, \ldots, x_n\}$$

$$a_1 \rightarrow x_1, \ldots, a_n \rightarrow x_n, x_1 \rightarrow b, \ldots, x_n \rightarrow b$$

OR^{n+1}=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a_1, \ldots, a_n, b\}, B = A \cup \{x\}$$

$$a_1 \rightarrow x, \ldots, a_n \rightarrow x, x \rightarrow b$$

The patterns AND^{n+1} and OR^{n+1} are visualized in Fig. 5. It can be verified that AND^{n+1} induces $\wedge_{n+1}(a_1, \ldots, a_n, b)$ and that OR^{n+1} induces $\vee_{n+1}(a_1, \ldots, a_n, b)$ by:

1. The multi-labeling of the pattern is that a_1 to a_n are not attacked so they are in, and therefore the auxiliary arguments are out, and therefore b is in;

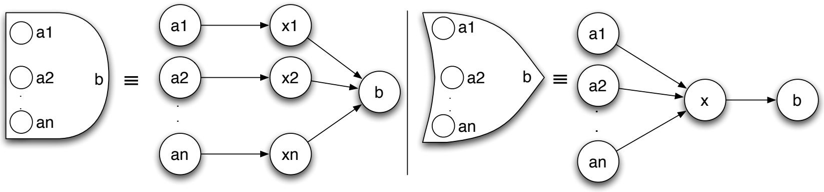


Fig. 5. Conjunction and disjunction (Example 4).

2. The constraint of conjunction is that if either one of the a_i is out, then x_i is in. The reason is that x_i is an auxiliary argument, and therefore, if it is not attacked by a_i, it is not attacked at all. If x_i is in, then b is out. vice versa, if b is in, then the x_i are out, and thus the a_i are in. The constraint of disjunction is that if all of the a_i are out, then x is in. If x is in, then b is out. Vice versa, if b is in, then x is out, and thus one of the a_i is in. By trying out all possibilities, it can be checked that these are the only constraints that hold.

Example 5 (Second-order attack).
ATTACK^3=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a, b, c\}, B = A \cup \{x, y\}$$

$$a \rightarrow x, x \rightarrow y, y \rightarrow c, b \rightarrow y$$

The pattern ATTACK^3 is visualized in Fig. 6. It can be verified that ATTACK^3 induces $\#_3(a, b, c)$ by

1. The multi-labeling of the pattern is that a and b are not attacked so they are in, and therefore the auxiliary arguments are out, and therefore c is in;
2. The constraint of second-order attacks is that if a is in, then x is out. Moreover, if b and x are out, then y is in. If y is in, then c is out. The converse can be checked in the same way. By trying out all possibilities, it can be checked that this is the only constraint that holds.

Example 6 (Equivalence).
EQUIV^2=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a, b\}, B = A \cup \{z_1, z_2\}$$

$$a \rightarrow z_1, z_1 \rightarrow b, b \rightarrow z_2, z_2 \rightarrow a$$

. It can be verified that EQUIV^2 induces $\equiv_2 (a, b) : a^{\in} \equiv b^{\in} \wedge a^{\notin} \equiv b^{\notin}$.

Examples 7 and 8 illustrate the difference between the multi-label and the constraints.

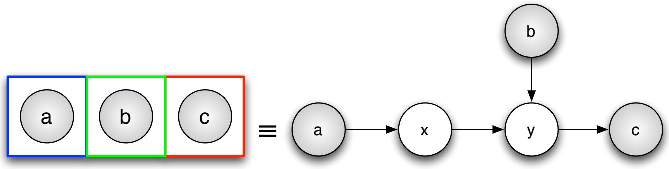


Fig. 6. Second-order attack pattern (Example 5).

Example 7. Consider the argumentation pattern $\langle A, M, C\rangle$, visualized in Fig. 7.a, where $A = \{b, d\}$. The pattern is given by multi-labeling $M(b) = M(d) = \{\in, \notin, ?\}$ together with an empty set of constraints. The two sorted argumentation framework is $\langle A, B, \rightarrow\rangle$ with

$$A = \{b, d\}, B = A \cup \{a, c\}$$

$$a \rightarrow b, b \rightarrow a, c \rightarrow d, d \rightarrow c$$

Consider now another pattern, represented in Fig. 7.b, where $A = \{b, d\}$. The pattern is given by multi-labeling $M(b) = M(d) = \{\in, \notin, ?\}$ together with the following constraint:

$$(d^{\in} \Leftarrow b^{\notin}) \wedge (b^{\in} \Leftarrow d^{\notin})$$

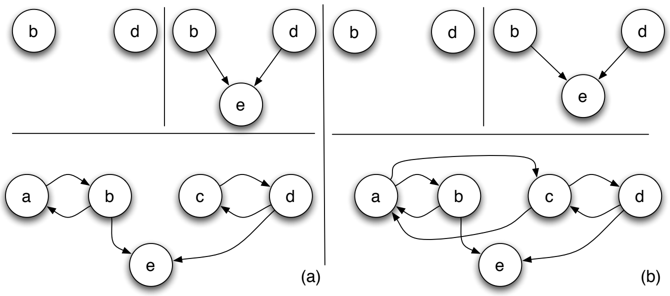
Consider now the introduction of argument e which is attacked by the two arguments b, d of the pattern. In the first case, argument e can have any label $\{\in, \notin, ?\}$ while in the second case, it cannot be $\in$, since b and d cannot both be $\notin$, as given by the constraint of the pattern. Fig. 7 shows in the tables the labelings allowed for each pattern. The two patterns have the same set of arguments and the same multi-labeling but distinct constraints. Notice that only a subset of the labelings satisfying the constraints of the first pattern satisfies the constraints of the second pattern.

Example 8. Consider the two two-sorted AF:

1. a single focal argument a, no attacks,
2. a single focal argument a and an auxiliary argument b which attack each other.

Moreover, consider the use of this pattern. The first should say that a is *in*, the second that a is either *in*, *out* or *undecided*. The constraints induced by the two multi-sorted AFs are the same (empty constraint), but the difference is represented by the multi-label.

In the context of flattening, Gabbay [10] discusses the notion of *critical subsets*. Given two argumentation frameworks where the set of arguments S_1 of the first AF is a subset of the set S_2 of the second AF, Gabbay [10] claims that S_2 is a *critical subset* of S_1 if and only if every Caminada labeling on S_2 can be extended uniquely to a labeling on S_1. This means that the additional arguments



	a	b	c	d	e
labeling1	IN	OUT	IN	OUT	IN
labeling2	IN	OUT	OUT	IN	OUT
labeling3	OUT	IN	OUT	IN	OUT
labeling4	OUT	IN	IN	OUT	OUT
labeling5	?	?	?	?	?
labeling6	?	?	IN	OUT	?
labeling7	?	?	OUT	IN	OUT
labeling8	IN	OUT	?	?	?
labeling9	OUT	IN	?	?	OUT

	a	b	c	d	e
labeling1	IN	OUT	OUT	IN	OUT
labeling2	OUT	IN	IN	OUT	OUT
labeling3	?	?	?	?	?

Fig. 7. Two patterns with the same multi-labeling and different constraints.

of S_1 only help in clarifying what is going on in S_2 and do not add any additional information. Critical subsets may recall the notion of actual arguments, whose labels are assigned, and depending on them, the labels of the auxiliary arguments are assessed.

2.4 Combining patterns

Patterns can be combined, just like boolean operators. For example, we can combine $\wedge^2$ to $\wedge^3$ and $\wedge^4$. Since attack works as a negation, we can form all kind of propositional combinations. For example, we can combine conjunction and attack to a combined conjunctive attack, known as accrual.

Example 9 (Accrual). Consider the following accrual attack pattern:

$$\#_{n+1}(a_1, \ldots, a_n, b) : (a_1^{\not\in} \vee \ldots \vee a_1^{\not\in} \Leftarrow b^{\in}) \wedge (a_1^{\in} \wedge \ldots \wedge a_1^{\in} \Rightarrow b^{\not\in})$$

This is $\wedge_{n+1}(a_1, \ldots, a_n, c)$ extended with an attack $b \rightarrow c$. Here, the latter attack acts as a kind of negation.

When combining two patterns, we can identify some of their arguments, and then abstract these arguments away. The definition of patterns' combination is left for further research.

3 Patterns

In this section, we present how to define the argumentation patterns of well-known extended argumentation frameworks and structures.

3.1 The Toulmin scheme

Dung's argumentation framework introduces a unique binary relation among arguments, called attack relation. The notion of support is rather controversial in argumentation theory. Here, without taking a position in the debate about the representation of this notion, we present an argumentation pattern for modeling support which we adopt in the Toulmin pattern. Our support pattern idea is driven by structured argumentation where argument a supports argument b if a attacks those arguments contradicting b's conclusion, i.e., undercutting b. We move this case to abstract argumentation and the two-sorted argumentation framework in Figure 8 models support with the auxiliary argument $\neg b$ with the meaning that $a^{\not\in}$ implies $b^{\not\in}$. This interpretation of support in abstract argumentation has been proposed by Cayrol and Lagasquie-Schiex [8] and we represent it by means of patterns.

Example 10 (Support). The support pattern is defined by multi-labeling $M(c_1) = \ldots = M(c_n) = \{\in\}$ and $M(a) = M(b) = \{\not\in\}$, together with:

$$\rhd_{n+2}(a, b, c_1, \ldots, c_n) : ((c_1^{\in} \wedge \ldots \wedge c_n^{\in}) \Rightarrow b^{\not\in} \wedge a^{\not\in}) \wedge$$

$$((a^{\in} \wedge (c_1^{\not\in} \wedge \ldots \wedge c_n^{\not\in})) \Leftarrow b^{\in})$$

Consider $\rhd_4$ a="Jones was born in England", b="Jones is a British citizen", c_1="Jones does not have a British passport" and c_2="Jones has a dutch accent". Now consider the pattern together with argument d. We have the following situation: d = "Mary says she saw Jones' British passport and he has no dutch accent", so $d \rightarrow c_1 \wedge c_2$, leading to the labeling $d^{\in} \wedge c_1^{\not\in} \wedge c_2^{\not\in} \wedge b^{\in} \wedge a^{\in}$. If d = "Jones' birth certificate is Bermudian", so $d \rightarrow a$, with the labeling $d^{\in} \wedge c_1^{\in} \wedge c_2^{\in} \wedge b^{\not\in} \wedge a^{\not\in}$. SUPPORT^{n+2}=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a, c_1, \ldots, c_n\}, B = A \cup \{b, \neg b\}$$

$$a \rightarrow \neg b, \neg b \rightarrow b, c_1 \rightarrow b, \ldots, c_n \rightarrow b, c_1 \rightarrow a, \ldots, c_n \rightarrow a$$

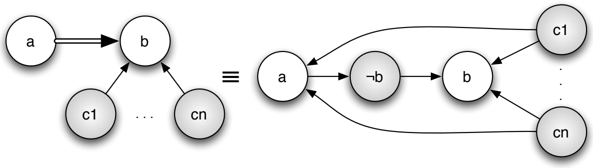


Fig. 8. The support pattern (Example 10).

Notice that the support pattern includes all attackers c_i of b. This means that we embed them in the pattern and argument b cannot be attacked by any argument external to the pattern. Thus b is an auxiliary argument which cannot be attacked, but it still can attack other arguments. Argument b is an "output" node of the pattern. Another approach to support has been introduced by Boella et al. [4], but in this case a deductive model of support is provided where the label *out* of argument a does not imply the same label for argument b.

The Toulmin scheme, in Fig. 1, is one of the most famous patterns in argumentation theory. There is not a unique model for representing the Toulmin scheme, there are many versions in which the warrant and the rebuttal support and attack different elements of the scheme. We provide a possible pattern but many other patterns are suitable for this scheme. Consider the following well-known example about the British citizenship.

Example 11 (Toulmin scheme). Jones tries to convince Mary that he is a British citizen. The claim is "I am a British citizen". Then Jones can support his claim with the data "I was born in Bermuda". In order to move from the data to the claim, Jones has to supply a warrant to bridge the gap between them with the rule "A man born in Bermuda will legally be a British citizen". If Mary does not deem the warrant as credible, Jones should supply the legal provisions as backing statement to show that it is true that the rule holds. Finally, the rebuttal of Mary is exemplified as follows "A man born in Bermuda will legally be a British citizen, unless he has betrayed Britain and has become a spy of another country."

In Example 11, the warrant, which can be modeled as a strict rule in structured argumentation, connects the data and the claim and it is supported by the backing. Moreover, the warrant is attacked by the rebuttal. We model the rules, i.e., the warrant, in the Toulmin pattern following the example of Wyner et al. [16] for the strict rule where $z \rightarrow c$. Moreover we have to model the support given by the backing to the warrant and finally, the attack from the rebuttal to the warrant and the claim. Note that the Toulmin scheme is the combination of patterns we defined thus far, as shown in Fig. 9. It combines a transistor where the collector is the data, the emitter is the claim, and the base is the warrant, a support pattern, and a conjunctive pattern.

Example 12 (Continued). The Toulmin pattern is defined by multi-labeling $M(d) = \{\in\}$, $M(r) = \{\in\}$ and $M(b) = M(w) = M(c) = \{\not\in\}$, together with:

$$TS(d, c, w, b, r) : (r^{\not\in} \wedge (w^{\in} \wedge b^{\in}) \Leftarrow c^{\in}) \wedge (d^{\not\in} \Rightarrow w^{\not\in} \wedge c^{\not\in})$$

The pattern is visualized in Fig. 9. Now consider the pattern together with argument e. We have the following situation: e = "Mary lies asserting that Jones is a spy", so $e \rightarrow r$, the labeling is $e^{\in} \wedge r^{\not\in} \wedge b^{\in} \wedge d^{\in} \wedge w^{\in} \wedge c^{\in}$. The labeling satisfies the invariant expressed by the constraints. TS=$\langle A, B, \rightarrow \rangle$ with

$$A = \{d, c, w, b, r\}, B = A \cup \{\neg z, z, \neg c, \neg w, x_1, x_2, y_1, y_2\}$$

$$z \rightarrow \neg z, \neg z \rightarrow z, \neg z \rightarrow w, w \rightarrow \neg c, d \rightarrow \neg z, \neg c \rightarrow c, c \rightarrow \neg c,$$

$$r \rightarrow x_1, r \rightarrow x_2, x_1 \rightarrow y_1, x_2 \rightarrow y_2, y_1 \rightarrow c, y_2 \rightarrow w, y_2 \rightarrow b, b \rightarrow \neg w, \neg w \rightarrow w$$

Notice that the relation between b and w is a support relation as modeled above where the attacker c_i is identified by auxiliary argument y_2.

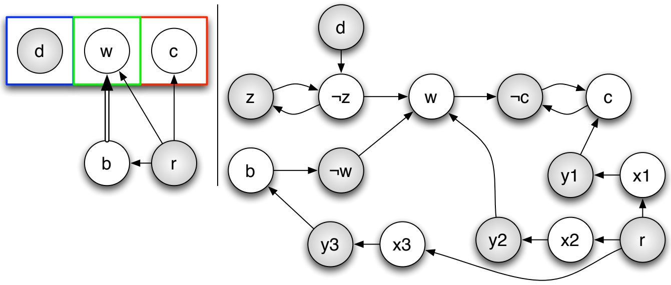


Fig. 9. The Toulmin pattern (Example 12).

3.2 Patterns for higher-order attacks

In Section 2, we introduced the pattern for second-order attacks where we follow the model of Boella et al. [3] for the multi-sorted argumentation framework. However, Modgil and Bench-Capon [13] and Baroni et al. [1] propose another way to model second-order attacks. Using patterns, we can shows that the three models are equivalent from the point of view of multi-labeling and constraints while they differ for the two sorted argumentation frameworks which induce the pattern.

Example 13 (Second-order patterns). The two patterns of Modgil and Bench-Capon [13] and Baroni et al. [1] are given by the same multi-labeling of Example 3 $M(a) = M(b) = M(c) = \{\in\}$, together with the same constraints:

$$\#_3(a, b, c) : (a^{\notin} \vee b^{\in} \Leftarrow c^{\in}) \wedge (a^{\in} \wedge b^{\notin} \Rightarrow c^{\notin})$$

Fig. 4 visualizes the multi-sorted AFs proposed by [13], ATTACK$^3{}_1$, and [1], ATTACK$^3{}_2$, which are formalized as follows: ATTACK$^3{}_1$=$\langle A, B, \rightarrow\rangle$ with

$$A = \{a, b, c\}, B = A \cup \{r\text{-}c, r\text{-}a, a\text{-def-}c, b\text{-def-}(a\text{-def-}c), r\text{-}b\}$$

$$a \rightarrow r\text{-}a, r\text{-}a \rightarrow a\text{-def-}c, a\text{-def-}c \rightarrow c, c \rightarrow r\text{-}c, b \rightarrow r\text{-}b,$$

$$r\text{-}b \rightarrow b\text{-def-}(a\text{-def-}c), b\text{-def-}(a\text{-def-}c) \rightarrow a\text{-def-}c$$

ATTACK$^3{}_2$=$\langle A, B, \rightarrow\rangle$ with $A = \{a, b, c\}, B = A \cup \{\alpha, \beta\}$

$$\alpha \rightarrow \beta, \beta \rightarrow c$$

Now consider the pattern, where arguments have the same meaning as in Example 3, together with argument d = "In the Intelligence's documents there is nothing about controlling Jones", such that $d \rightarrow b$, as visualized in Fig. 10. We have the following situation for the first pattern [13]: $a^{\in} \wedge d^{\in} \wedge b^{\notin} \wedge c^{\notin}$, and the same holds for the second pattern [1]. Notice that the two patterns are the same pattern as the one of Example 3, and only the two-sorted argumentation framework which induces the pattern differs. This means that they differ only in the choice of the auxiliary arguments and the constraints which hold for these auxiliary arguments.

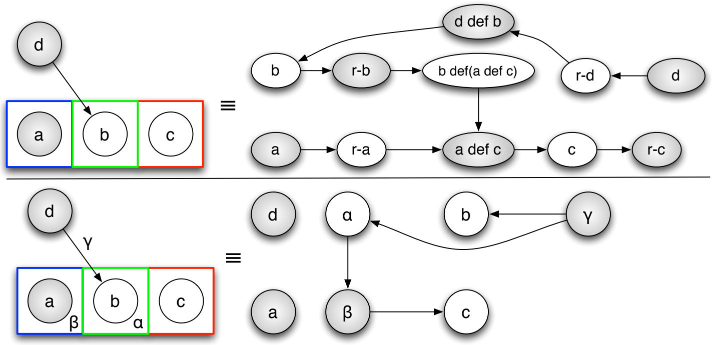


Fig. 10. Second-order attack pattern (Example 13).

3.3 Patterns for Proof Standards

In everyday reasoning and in legal reasoning, proof standards play a relevant role in those situations in which, involving risk, we apply higher standards rather than in cases where there is not much to loose. Two standards of proof have been recently analyzed by Brewka and Woltran [6] using the acceptability conditions of the abstract dialectical frameworks. The proof standards we consider are: (i) argument s is labeled $\in$ if the set of arguments R contains no node attacking s and at least one node supporting s and, (ii) s is labeled $\in$ if R contains all nodes supporting s and no node attacking s.

Example 14 (Proof standards). The patterns for proof standards are given by the same multi-labeling $M(t_1) = \ldots = M(t_n) = M(s) = \{\in\}$ and $M(r_1) = \ldots = M(r_m) = \{\notin\}$, together with different constraints:

$$PS1_{n+m+1}(t_1, \ldots, t_n, r_1, \ldots, r_m, s):$$

$$(t_i^{\in} \wedge (r_1^{\notin} \wedge \ldots \wedge r_m^{\notin}) \Leftarrow s^{\in}) \wedge ((t_1^{\notin} \wedge \ldots \wedge t_n^{\notin}) \vee r_i^{\in} \Rightarrow s^{\notin})$$

$$PS2_{n+m+1}(t_1, \ldots, t_n, r_1, \ldots, r_m, s):$$

$$((t_1^{\in} \wedge \ldots \wedge t_n^{\in}) \wedge (r_1^{\notin} \wedge \ldots \wedge r_m^{\notin}) \Leftarrow s^{\in}) \wedge (t_i^{\notin} \vee r_i^{\in} \Rightarrow s^{\notin})$$

Fig. 11 visualizes the two-sorted AFs which induce these patterns.

PS1^{n+m+1}=$\langle A, B, \rightarrow\rangle$ with

$$A = \{t_1, \ldots, t_n, r_1, \ldots, r_m\}, B = A \cup \{s, \neg s, \neg r_1, \ldots, \neg r_m\}$$

$$t_1 \rightarrow \neg s, \ldots, t_n \rightarrow \neg s, \neg s \rightarrow s,$$

$$r_1 \rightarrow s, \ldots, r_m \rightarrow s, \neg r_1 \rightarrow r_1, \ldots, \neg r_m \rightarrow r_m$$

PS2^{n+m+1}=$\langle A, B, \rightarrow\rangle$ with

$$A = \{t_1, \ldots, t_n, r_1, \ldots, r_m\}, B = A \cup \{s, x_1, \ldots, x_n\}$$

$$t_1 \rightarrow x_1, \ldots, t_n \rightarrow x_n, x_1 \rightarrow s, \ldots, x_n \rightarrow s, r_1 \rightarrow s, \ldots, r_m \rightarrow s$$

$$\neg r_1 \rightarrow r_1, \ldots, \neg r_m \rightarrow r_m$$

Notice that, in the two sorted argumentation framework, we avoid to have argument s attacked by other arguments external to the pattern because we consider every argument r_i attacking s.

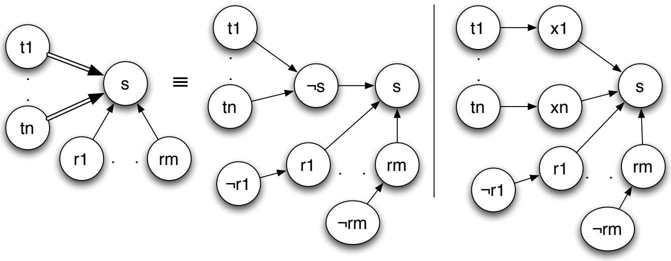


Fig. 11. Proof standards (Example 14).

4 Related work

Argumentation patterns may recall to mind the structure of syllogisms, rules as modus ponens or argumentation schemes. Reed et al. [14] explain argumentation schemes as argument forms that represent inferential structures of arguments used in everyday discourse, and in special contexts like legal argumentation and artificial intelligence. Besides forms of reasoning like modus ponens, some of the most common schemes are neither deductive nor inductive, but defeasible and presumptive. One of the issues which brings argumentation theory and computer science closer together is the need to diagram such arguments [14]. Diagramming is of interest both to those in argumentation as a tool in the analytical toolbox, and to computer scientists as a precursor to implementable formalization. We agree about the relevance of diagrams in representing the relationships of the arguments but, as we have shown in the paper, it is not precise enough to define all the relations among the arguments.

Our patterns together with their multi-labeling and constraints can be compared to the abstract dialectical frameworks, defined by Brewka and Woltran [6], and their acceptance functions. They provide a generalization of Dung's argumentation framework. Such a framework is defined as a tuple $D = (S, L, C)$ where S is a set of nodes, $L \subseteq S \times S$ is a set of links and C is an acceptance condition associated to each node. C_s specifies the exact conditions under which argument s is accepted. Summarizing, if for some $R \subseteq par(s)$, where $par(s)$ are the parents of node s, we have $C_s(R) = in$ then s will be accepted provided the nodes in R are accepted. We can express the acceptability condition with a conjunction pattern where the set R contains the arguments $a_1, \ldots, a_n$ and argument s corresponds to our argument b. An advantage in using patterns is that we can compose them together to form a larger pattern while Brewka and Woltran [6] need to define the acceptance function from scratch.

5 Conclusions

The success criteria of argumentation patterns lies for the 90% in the proposed visualization. The contribution of this paper with respect to visualization is to use transistors for second-order attack patterns, and introduce visualizations for the conjunction and disjunction patterns inspired by visualizations of AND and OR gates in boolean circuits. Moreover, we show how these visualizations can be combined, as in the case of the Toulmin scheme.

Argumentation patterns are reusable solutions to common problems in argumentation theory, and are driven by practical rather than theoretical concerns. We define argumentation patterns by a multi-labeling, i.e., the labels of the arguments inside the pattern, together with a set of constraints showing the relations among the arguments, even if some of them are attacked by arguments external to the pattern.

We identify, among others, the following patterns in the argumentation literature, and formalize them in our framework: the support relation, the Toulmin scheme, second-order attacks, accrual, and the standards of proof. Patterns avoid us to define extended argumentation frameworks *ad hoc* for particular application domains.

Two main points emerge from this initial exploration of how to visualize and formalize argumentation patterns. First, the language has to distinguish the description of the behaviour of the pattern as standalone framework, and it has to contain a description of how the behaviour of the pattern changes when it is attacked from outside the framework. In this paper, we use multi-labelling for the former, and constraints for the latter. The general point is that a pattern definition has to provide the definition of part of an argumentation framework, or an argumentation framework in an environment. The SCC recursive scheme [2] can bring some inspiration since here also Dung's semantics are associated to a context to define the base function. The second technical issue which is emerged is the soundness and completeness proofs needed for patterns. We define the semantics of patterns in terms of multi-labelling and constraints, then the syntax in terms of flattening. We need to show now that they are equivalent. All this is left as future work.

References

1. Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *Int. J. Approx. Reasoning*, 52(1):19–37, 2011.

2. Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artif. Intell.*, 168(1-2):162–210, 2005.
3. 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.
4. Guido Boella, Dov M. Gabbay, Leendert van der Torre, and Serena Villata. Support in abstract argumentation. In *Procs. of the 3rd Int. Conf. on Computational Models of Argument*, pages 40–51. Frontiers in Artificial Intelligence, IOS Press, 2010.
5. Guido Boella, Leendert van der Torre, and Serena Villata. Analyzing cooperation in iterative social network design. *Journal of Universal Computer Science*, 15(13):2676–2700, 2009.
6. Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *KR*. AAAI Press, 2010.
7. 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.
8. Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *Int. J. Intell. Syst.*, 25(1):83–109, 2010.
9. 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):321–358, 1995.
10. Dov M. Gabbay. Fibring argumentation frames. *Studia Logica*, 93(2-3):231–295, 2009.
11. Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9(2):215–261, 1999.
12. Jérôme Lang and Pierre Marquis. Reasoning under inconsistency: A forgetting-based approach. *Artif. Intell.*, 174(12-13):799–823, 2010.
13. S. Modgil and T.J.M Bench-Capon. Metalevel argumentation. Technical report, page www.csc.liv.ac.uk/research/ techreports/techreports.html, 2009.
14. Chris Reed, Douglas Walton, and Fabrizio Macagno. Argument diagramming in logic, law and artificial intelligence. *Knowledge Eng. Review*, 22(1):87–109, 2007.
15. Stephen Toulmin. *The Uses of Argument.* Cambridge University Press, 1958.
16. Adam Wyner, Trevor Bench-capon, and Paul Dunne. Instantiating knowledge bases in abstract argumentation frameworks. In *The Uses of Computational Argumentation: Papers from the AAAI Fall Symposium*, pages 76–83, 2009.

Reasoning about and Discussing Preferences between Arguments[*]

T.L. van der Weide and F. Dignum

Universiteit Utrecht
{tweide,dignum}@cs.uu.nl

Abstract. Agents that have different knowledge bases and preferences over arguments can use dialogues to exchange information and explanations. In order for the dialogue to be useful, agents need to utilize the other participants' knowledge fully while being resistant against manipulation. Furthermore, the information they exchange can be objective but also subjective such as what goals an agent wants to achieve. To understand why another agent draws a certain conclusion it is necessary to understand and communicate preferences over arguments. This paper proposes an ASPIC-based meta-level argumentation logic for reasoning about preferences over arguments. Extended argumentation frameworks are used to determine what arguments are justified. Prakken's dialogue framework is then adapted for meta-level arguments and a protocol is proposed that explicitly distinguishes between objective and subjective topics. Several mechanisms for using other agents' knowledge have been proposed in the literature. This paper proposes to use different acceptance attitudes with respect to claims made in a dialogue and to store the source of those claims on a meta-level. In the meta-level, agents can then reason about the effect of other agents' claims on the conclusive force of arguments. This makes agents more robust against manipulation and able to handle new information better.

1 Introduction

The following dialogue illustrates what motivated this paper. In this example, agent β tries to persuade agent α to eat a healthy salad rather than a pizza.

1. α claims: I want to eat pizza quattro formaggi because I like gorgonzola
2. β questions a premise: why do you like gorgonzola?
3. α answers: I don't know
4. β claims: you should eat salad because salad is healthier than pizza
5. β claims: health is more important than taste
6. α questions a premise: why is salad healthier?
7. β claims: salad contains less calories than pizza.

[*] The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

In move 2, β asks α to justify a subjective statement. Asking why a subjective statement is true is different than asking why an objective statement is true, because only the agent himself can determine whether the subjective statement is true. In contrast to an objective statement, questioning or giving a counterargument for a subjective statement should not attack the statement. Furthermore, in move 5, β claims that move 4's claim is stronger than move 1's claim, which only makes the attack of move 4's argument on move 1's argument successful.

Preferences between arguments describe what argument has more conclusive force and determine what attacks are successful [12, 6, 10]. Therefore it is important to be able to discuss preferences over arguments. Preferences between arguments may differ per agent and are therefore subjective, but disputable. Because preferences may have a significant effect on what arguments are acceptable, it is important to discuss them in a dialogue. Extended Argumentation Frameworks (EAFs) have been proposed to argue about preferences between arguments [6]. However, in dialogue frameworks such as [8] it is not possible for agents to discuss their preferences between arguments. In this paper, a dialogue framework is proposed that enables participants to discuss preferences between arguments.

In dialogues with argumentation, participants make claims about the truth of statements and justify those claims with a supporting argument. We distinguish between claims whose truth can be established objectively and claims whose truth can only be established subjectively. For example, whether salad is healthier than pizza can be established objectively whereas whether agent α likes gorgonzola only α himself can establish. Other examples of subjective statements are the values and goals of an agent, but also the conclusive force of arguments for an agent because the trustworthiness of others may be used to establish this. Existing dialogue frameworks such as [8] do not distinguish between objective and subjective statements. This means that α's claim that he likes gorgonzola is attacked by β's question why he likes it. This paper distinguishes between subjective and objective statements and introduces a protocol where the burden of proof and production of each participant depends on whether a certain statement is subjective or objective.

If agent α makes a claim whose truth β cannot establish (e.g. that α likes gorgonzola), then β should reason about whether β should accept α's claim to be true. Several existing approaches introduce agent types that treat incoming arguments differently. For example, in [11], one agent type simply puts every argument in his knowledge base, whereas another agent type only puts an argument in his knowledge base if he has no attacking arguments. In [7], three different so-called acceptance attitudes are proposed that treat incoming arguments differently. This paper introduces a general argumentation-based approach in which different 'acceptance strategies' can be implemented.

Section 2 describes the ASPIC argumentation framework of [10] and extends it by adding meta-level argumentation and showing its relation to extended argumentation frameworks. Next, Section 3 adapts the dialogue framework of [8] to allow discussing preferences between arguments. Furthermore, a protocol is proposed in which the burden of production and the burden of proof depends on whether a statement is objective or subjective. Section 4 describes a general argumentation-based approach that agents can use to reason about what to do with arguments they receive from other agents. This paper is ended with some conclusions and discussion in Section 5.

2 Argumentation

To represent arguments, the ASPIC+ abstract framework for structured argumentation is used, which provides an abstract account of the structure of arguments, the nature of attack and the effect of preferences between arguments on what attacks are successful [10]. The conclusive force of arguments (also called preferences between arguments) determines what attacks are successful. In the introduction example, agent β claims that health is more important than taste, which means that the argument to eat salad has more conclusive force than the argument to eat pizza. Consequently, only the salad argument's attack on the pizza argument is successful.

ASPIC+ does not provide means to reason about the conclusive force of arguments. In [5], ASPIC+ is further developed to define attacks on attacks using an abstract function, which defines when an argument or a set of arguments attacks an attack. Because this function is abstract, Section 2.2 proposes a more specific approach to argue about conclusive force by using meta-argumentation. Also, Section 4 describes a meta-argumentation system to reason about an agent's commitments in a dialogue, his beliefs and how that influences the conclusive force of object-level arguments.

2.1 ASPIC+: Structured Argumentation

The ASPIC abstract framework for structured argumentation integrates work on rule-based argumentation with Dung's abstract approach [2]. The notion of an argumentation system extends the familiar notion of a proof system by distinguishing between strict and defeasible inference rules. The informal reading of a strict inference rule is that if its antecedent holds, then its conclusion holds without exception. The informal reading of a defeasible inference rule is that if its antecedent holds, then its conclusion tends to hold. A strict rule is an expression of the form $\phi_1, \ldots, \phi_m \rightarrow \phi$ and a defeasible rule is an expression of the form $\phi_1, \ldots, \phi_m \Rightarrow \phi$, with $m \geq 0$.

Definition 1 (Argumentation System). *An argumentation system is a tuple $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}, \mathsf{cf} \rangle$ with*

- *$\mathcal{L}$ the language of predicate logic,*
- *$\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ such that $\mathcal{R}_s$ is a set of strict and $\mathcal{R}_d$ is a set defeasible inference rules, and*
- *cf a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$.*

For $\phi \in \mathcal{L}$, it is always the case that $\neg\phi \in \mathsf{cf}(\phi)$ and $\phi \in \mathsf{cf}(\neg\phi)$. Also for $\phi, \psi \in \mathcal{L}$, if $\phi \in \mathsf{cf}(\psi)$ and $\psi \notin \mathsf{cf}(\phi)$, then ϕ is called the *contrary* of ψ. If $\phi \in \mathsf{cf}(\psi)$ and $\psi \in \mathsf{cf}(\phi)$, then ϕ and ψ are called *contradictory*.

Arguments are defined following [12] and can be thought of as inference trees.

Definition 2 (Argument). *An argument A in an argumentation system $\mathcal{AS} = \langle \mathcal{L}, \mathcal{R}_s \cup \mathcal{R}_d, \mathsf{cf} \rangle$ is:*

- *ϕ if $\phi \in \mathcal{L}$ with* $\mathsf{premises}(A) = \phi$*;* $\mathsf{conc}(A) = \phi$*;* $\mathsf{sub}(A) = \{A\}$*;* $\mathsf{lastRule}(A) =$ *undefined.*
- *$A_1, \ldots, A_n \rightarrow / \Rightarrow \phi$ if $A_1, \ldots, A_n$ are arguments in $\mathcal{AS}$ such that there is a strict / defeasible inference rule* $\mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \rightarrow / \Rightarrow \phi$ *in $\mathcal{R}_s/\mathcal{R}_d$. Furthermore,*
  - $\mathsf{premises}(A) = \bigcup_{i=1}^{n} \mathsf{premises}(A_i)$
  - $\mathsf{conc}(A) = \phi$
  - $\mathsf{sub}(A) = \{A\} \cup \bigcup_{i=1}^{n} \mathsf{sub}(A_i)$
  - $\mathsf{lastRule}(A) = \mathsf{conc}(A_1), \ldots, \mathsf{conc}(A_n) \rightarrow / \Rightarrow \phi$

The set of all arguments in an argumentation system $\mathcal{AS}$ is denoted as $\mathsf{Args}(\mathcal{AS})$. Arguments are constructed by applying inference rules to some knowledge base in an argumentation system. A *knowledge base* is a set of formulae consisting of a set of axioms and a set of ordinary premises. An argument A can be constructed from a knowledge base $\mathcal{K}$ if all A's premises are contained in $\mathcal{K}$.

Following [10], the following kinds of attack can be distinguished.

Definition 3 (Attack). *Let $A, B \in \mathsf{Args}(\mathcal{AS})$ be two arguments. Argument A* attacks *B iff A rebuts, undermines or undercuts B, where:*

- *A* rebuts *B if A's conclusion is the contrary of the conclusion of some defeasible inference rule that was applied in B,*
- *A* undermines *B if A's conclusion is the contrary of one of B's premises,*
- *A* undercuts *B if A concludes an exception to a defeasible inference rule that was applied in B.*

Following [8], we will look at when an argument extends another argument, because this is useful in dialogues where agents may first give an argument A and later an argument B that justifies a premise of A. If the conclusion of argument A is the premise of argument B, then we say that A extends B.

Definition 4 (Extended Argument). *Let $A, B \in \mathsf{Args}(\mathcal{AS})$. We say that A* extends *B* on *B' if and only if B has an atomic argument B' such that* $\mathsf{conc}(A) = \mathsf{conc}(B')$*. Furthermore, if A extends B on B', then $A \oplus_{B'} B$ is an argument with*

- $\mathsf{conc}(A \oplus_{B'} B) = \mathsf{conc}(B)$*,*
- $\mathsf{premises}(A \oplus_{B'} B) = \mathsf{premises}(A) \cup (\mathsf{premises}(B) \setminus \{\mathsf{conc}(A)\})$*,*
- $\mathsf{lastRule}(A \oplus_{B'} B) = \mathsf{lastRule}(B)$*,*

– $\mathsf{sub}(A \oplus_{B'} B) = \mathsf{sub}(A) \cup (\mathsf{sub}(B) \setminus \{B'\})$

Please note that extending an argument has nothing to do with extended argumentation frameworks. Further note that if $A, B \in \mathsf{Args}(\mathcal{AS})$, then the extended argument $A \oplus_{B'} B$ is also in $\mathsf{Args}(\mathcal{AS})$.

2.2 Meta Argumentation

In a meta-argumentation system, arguments are constructed with respect to an (object-level) argumentation system. To reason about the conclusive force of (object-level) arguments, meta-argumentation systems are required to have a special predicate $\preceq$ that compares the conclusive force of object-level arguments. For example, if A is the argument that α should eat pizza because it is tasty and B that α should eat salad because it is healthy, then $A \preceq B$ denotes that B has as much or more conclusive force as A. Extended argumentation frameworks will be constructed using the predicate $\preceq$.

Definition 5 (Meta-Argumentation System). *A* Meta-Argumentation System *(MAS) on the basis of argumentation system $\mathcal{AS} = (\mathcal{L}, \mathcal{R}, \mathsf{cf})$ is an argumentation system $\mathcal{AS}' = (\mathcal{L}', \mathcal{R}', \mathsf{cf}')$ such that*

- *each formula in $\mathcal{L}$, rule in $\mathcal{R}$ and argument in $\mathsf{Args}(\mathcal{AS})$ is a constant in $\mathcal{L}'$,*
- *the functions on arguments (see Definition 2) are function symbols in $\mathcal{L}'$,*
- *$\preceq$ is a binary predicate in $\mathcal{L}'$.*

The predicate $\preceq$ denotes conclusive force. The predicate $\prec$ denotes strictly more conclusive force and is defined in the usual way.

A number of meta-argumentation systems can be stacked upon an argumentation system. This results in what we call a 'tower of argumentation systems'. In [15], logical languages are stacked in a similar way resulting in a tower or hierarchy of languages. Our approach is similar except that a meta-argumentation system $\mathcal{AS}'$ can only refer to its object argumentation system $\mathcal{AS}$ and not to argumentation systems that are below $\mathcal{AS}$.

Definition 6 (Tower Of Argumentation Systems). *A* tower of argumentation systems of level *$1 \leq n$ is a set $\{\mathcal{AS}_1, \ldots, \mathcal{AS}_n\}$ such that:*

- *$\mathcal{AS}_1$ is an argumentation system and*
- *for each $2 \leq i \leq n$: $\mathcal{AS}_i$ is a meta-argumentation system based on $\mathcal{AS}_{i-1}$.*

Given a tower of argumentation systems and a knowledge base for each argumentation system in the tower, we consider meta-argumentation theories.

Definition 7 (Meta-Argumentation Theory). *A* Meta-Argumentation Theory *(MAT) is a tuple $\langle \mathcal{T}_{\mathcal{AS}}, \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}\rangle$ such that*

- *$\mathcal{T}_{\mathcal{AS}} = \{\mathcal{AS}_1, \ldots, \mathcal{AS}_n\}$ is a tower of argumentation systems of level n, and*
- *for each $1 \leq i \leq n$: $\mathcal{K}_i$ is a knowledge base in argumentation system $\mathcal{AS}_i$*

– *for $2 \leq i \leq n$: $\mathcal{K}_i$ contains axioms for reflexivity and transitivity of the predicate $\preceq$.*

If we say that MAT is a meta-argumentation theory of level n, then $\mathsf{MAT} = \langle \mathcal{T}_{\mathcal{AS}}, \{\mathcal{K}_1, \dots, \mathcal{K}_n\}\rangle$ with $\mathcal{T}_{\mathcal{AS}} = \{\mathcal{AS}_1, \dots, \mathcal{AS}_n\}$ a tower of argumentation systems of level n.

Attack between arguments in a meta-argumentation system can be defined according to Definition 3. However, we also want that arguments in a MAS can attack the attacks between arguments in the AS on which the MAS is based. For this, the notion of *meta-attack* is introduced.

Definition 8 (Meta-Attack). *Let $\mathcal{AS}$ be an argumentation system, $\mathcal{AS}'$ be a meta-argumentation system on the basis of $\mathcal{AS}$, and $A_1, A_2 \in \mathsf{Args}(\mathcal{AS})$ and $B \in \mathsf{Args}(\mathcal{AS}')$. Argument B* meta-attacks *that A_1 attacks A_2 if and only if $\mathsf{conc}(B) = A_1 \prec A_2$ and A_1 attacks A_2 according to Definition 3.*

2.3 Argumentation Frameworks

An *Argumentation Framework* (AF) is a tuple $\langle \mathsf{Args}, \mathcal{R}\rangle$ where Args is a set of arguments and $\mathcal{R}$ a binary attack relation between those arguments [2]. A dialectical calculus can be used to evaluate what arguments are justified and rejected under different semantics.

Extended Argumentation Frameworks (EAFs) extend AFs with an attack relation between an argument and an attack between two arguments [6], a so-called pref-attack. An EAF is a tuple $\langle \mathsf{Args}, \mathcal{R}, \mathcal{D}\rangle$ with $\langle \mathsf{Args}, \mathcal{R}\rangle$ an AF and $\mathcal{D} \subseteq \mathsf{Args} \times \mathcal{R}$ the pref-attack relation. *Bounded hierarchical EAFs* (bhEAFs) is a class of EAFs that are stratified so that attacks at some level only are only pref-attacked by arguments in the next level up. [5] uses bhEAFs to link ASPIC+ with EAFs.

In meta-argumentation systems as defined in Definition 5, there is a binary predicate $\preceq$ to express preference between arguments in the object argumentation system. Because a tower of meta-argumentation system stratifies arguments neatly into different levels, meta-attack as defined in Definition 8 can be used to initialize the pref-attack relation in a bounded hierarchical EAF.

Definition 9 (Structured EAF). *Let MAT be a meta-argumentation system of level n. A* Structured EAF *on the basis of MAT is a bounded hierarchical EAF $\{(\mathsf{Args}_1, \mathcal{R}_1, \mathcal{D}_1), \dots, (\mathsf{Args}_{n-1}, \mathcal{R}_{n-1}, \mathcal{D}_{n-1}), (\mathsf{Args}_n, \mathcal{R}_n, \emptyset)\}$ such that*

– *Args_i a set of arguments on the basis of AS_i such that each argument can be constructed from $\mathcal{K}_i$, i.e. for each $A \in \mathsf{Args}_i$: $\mathsf{premises}(A) \subseteq \mathcal{K}_i$,*
– *for each $A, B \in \mathsf{Args}_i$: $(A, B) \in \mathcal{R}_i$ if A attacks B according to Definition 3,*
– *for each $A, B \in \mathsf{Args}_i$ and $C \in \mathsf{Args}_{i+1}$: if $(A, B) \in \mathcal{R}_i$ and C meta-attacks that A attacks B according to Definition 8, then $(C, (A, B)) \in \mathcal{D}_i$.*

In [6], the definitions can be found for when an argument is acceptable with respect to complete, preferred, stable, and grounded semantics. If S is a semantics, then we say that a formula ϕ is (1) *justified under* S if in each S-extension, there is an argument concluding ϕ; (2) *defensible under* S if ϕ is not justified under S, but there is an S-extension with an argument concluding ϕ; and, (3) *overruled under* S if there is no S-extension with an argument concluding ϕ.

3 Dialogue Framework

Because how an agent prefers arguments has a significant effect on what arguments he finds acceptable, it is important that agents can give and discuss their preferences between arguments in a dialogue. Furthermore, when discussing topics like what to do, the distinction between subjective and objective information has an effect on the participants' burden of persuasion.

The previous section proposed how to argue about the conclusive force of arguments. Section 3.1 adapts the dialogue framework in [8] such that preferences between arguments can be expressed and discussed in a dialogue. Section 3.2 describes and formalizes the distinction between objective and subjective statements. Finally, Section 3.3 proposes a protocol that is tailored for discussing objective and subjective statements.

3.1 Communication Language, Dialogue Moves and Dialogues

The participants of a dialogue use a communication language to communicate. The communication language depends on the topic language, which in this paper consists of a tower of argumentation systems so that the conclusive force of arguments can be discussed. For convenience, the communication language is split into a communication language for each argumentation system on each level.

Definition 10 (Communication Language). *Let* $\mathcal{T}_{\mathcal{AS}} = \{\mathcal{AS}_1, \ldots, \mathcal{AS}_n\}$ *be a tower of argumentation systems of level* n *with* $\mathcal{AS}_i = \langle \mathcal{L}_i, \mathcal{R}_i, \mathsf{cf} \rangle$ *for each* $\mathcal{AS}_i \in \mathcal{T}_{\mathcal{AS}}$. *A* communication language *for* $\mathcal{T}_{\mathcal{AS}}$ *is a set* $\mathcal{L}_{\mathcal{C}} = \mathcal{L}_{\mathcal{C}1} \cup \ldots \cup \mathcal{L}_{\mathcal{C}n}$ *such that for* $1 \leq i \leq n$*:*

- *for all* $A \in \mathsf{Args}(\mathcal{AS}_i)$*:* $\mathsf{claim}_i(A) \in \mathcal{L}_{\mathcal{C}i}$
- *for all* $\phi \in \mathcal{L}_i$*:* $\mathsf{why}_i(\phi), \mathsf{concede}_i(\phi), \mathsf{retract}_i(\phi) \in \mathcal{L}_{\mathcal{C}i}$

Because every argument on every level of a tower of ASs can be communicated, this communication language can be used to express preferences between arguments. Note that [8] distinguishes between claiming a formula and claiming an argument. In contrast, this definition does not distinguish between these two claims. Rather, if a participant just wants to claim a formula, then he should claim an atomic argument concluding that formula.

In a dialogue, agents can make *dialogue moves*. A dialogue move is made by an agent and can target previously made dialogue moves. Each dialogue move has an identifier.

Definition 11 (Dialogue Move). *Let $\mathcal{L}_\mathcal{C}$ be a communication language and $\mathcal{P}$ a set of agents. The set of* dialogue moves w.r.t. $\mathcal{L}_\mathcal{C}$ and $\mathcal{P}$ *is defined as* $\mathbb{N} \times \mathcal{P} \times \mathcal{L}_\mathcal{C} \times 2^{\mathbb{N}}$.

If $m = \langle i, \alpha, l, X\rangle$ is a dialogue move, then (a) $\mathsf{id}(m) = i$ denotes the identifier of move m; (b) $\mathsf{pl}(m) = \alpha$ denotes the agent that made move m; (c) $\mathsf{loc}(m) = l$ denotes the locution of move m; and, (d) $\mathsf{target}(m) = X$ denotes the set of move identifiers at which m is targeted.

In contrast to [8], a dialogue move targets a set of dialogue moves. This is necessary because if move m_1 claims argument A, move m_2 replies to m_1 by claiming argument B and move m_3 claims that A is preferred to B, then m_3 is targeted at both m_1 and m_2. If a dialogue move m's target is $\emptyset$, then m does not reply to any dialogue move. Also, if m and m' are dialogue moves in a dialogue such that $\mathsf{id}(m') \in \mathsf{target}(m)$, then we say that *move m replies to move m'*.

Definition 12 (Dialogue). *A* dialogue *is a tuple $\langle \mathcal{L}_\mathcal{C}, \mathcal{P}, M\rangle$ such that $\mathcal{L}_\mathcal{C}$ is a communication language, $\mathcal{P}$ a set of participants and M is a finite non-empty set $\{m_1, \ldots, m_n\}$ of dialogue moves w.r.t. $\mathcal{L}_\mathcal{C}$ and $\mathcal{P}$ such that for each $m_i \in M$: (1) $\mathsf{id}(m_i) = i$, and (2) for each $j \in \mathsf{target}(m_i)$: $0 < j < \mathsf{id}(m_i)$.*

The first condition ensures that every dialogue move in a dialogue has a unique identifier. The second condition ensures that every dialogue move must reply to 0 or to a dialogue move that has been made earlier in that dialogue, i.e. one with a lower identifier. Note that the second condition also ensures that there the first dialogue move always has target $\emptyset$.

If there is only a single dialogue move in a dialogue d that does not reply to any dialogue move, i.e. there is only one dialogue move with $\emptyset$ as target, then we say that dialogue d is a *single-topic dialogue*. Otherwise, the dialogue is also called a *multi-topic dialogue*.

In a dialogue, the participants can claim arguments. If a premise of a claimed argument is questioned, then an argument that extends the original argument can be given to answer that question. The following definition collects all arguments that have been uttered in a dialogue taking into account that arguments might extend other arguments.

Definition 13 (Arguments In A Dialogue). *Let $d = \langle \delta, M\rangle$ be a dialogue. The* arguments of level i in d *is the set $\mathsf{Args}_i(d)$ such that for all $m \in M$ such that $\mathsf{loc}(m) = \mathsf{claim}_i(A)$:*

- *if there is no $m' \in M$ such that $\mathsf{loc}(m') = \mathsf{claim}_i(B)$ with B extending A on some A', then $A \in \mathsf{Args}_i(d)$,*
- *if there is a $B \in \mathsf{Args}_i(d)$ that extends A on A', then $B \oplus_{A'} A$ in $\mathsf{Args}_i(d)$*

The first condition ensures that all arguments that have been claimed but not extended are contained in $\mathsf{Args}_i(d)$. The second condition ensures that if an argument is extended, then the extended argument is contained in $\mathsf{Args}_i(d)$ by using the possibly extended argument that is already in $\mathsf{Args}_i(d)$.

3.2 Subjective and Objective Statements

We say that a statement ϕ is *subjective to agent* α if only agent α can determine whether ϕ is true. Otherwise it is called *objective*. In the introduction example, 'α likes gorgonzola' and 'α finds health more important than taste' are subjective statements, whereas 'salad is healthier than pizza' is an objective statement. It is important to distinguish between subjective and objective statements because objective statements can be attacked by everyone whereas subjective statements cannot. A subjective statement can merely be challenged. This is important for the protocol and for the conclusive force of arguments. If a formula is subjective to an agent, then the negation of that formula is also subjective to that agent.

Definition 14 (Subjectivity Mapping). *Let* Agents *be a set of agents and* $\mathcal{L}$ *a logical language. A* subjectivity mapping *for* $\mathcal{L}$ *is a function* $s : \mathsf{Agents} \rightarrow 2^{\mathcal{L}}$ *that maps an agent to the set of formulae that are subjective to that agent such that* $\phi \in s(\alpha)$ *if and only if* $\neg\phi \in s(\alpha)$*.*

If $\phi \in s(\alpha)$, then we say that formula ϕ is subjective for agent α. If a formula $\phi \in \mathcal{L}$ is not subjective for any agent, i.e. ϕ not in $s(\alpha)$ for all α in Agents, then ϕ is called *objective*. Note that a formula is subjective to multiple agents if there are multiple agents α such $\phi \in s(\alpha)$. For example, the formula 'α likes gorgonzola and β likes gorgonzola' is subjective to both α and β.

3.3 Protocol

Protocols regulate dialogues by specifying what dialogue moves are legal. Some protocols distinguish between subjective and objective statements [1], but others do not. Statements like an agent's goals but also an agent's preferences between arguments are subjective because they are internal to that agent. Questioning or giving a counterargument for a subjective statement like that you like gorgonzola is different than questioning or giving a counterargument for an objective statement like that salad is healthier than pizza because the truth of a subjective statement can only be determined by the agent himself. In this section, we adapt [8]'s set of protocol rules to treat subjective and objective statements differently. The most important adaptation is that dialogues moves cannot be attacked on subjective claims they make.

[8] proposes the following five rules in order to capture the lower bound on coherent dialogues. Let $d = \langle \mathcal{L}_{\mathcal{C}}, \mathcal{P}, M \rangle$ be a dialogue. Dialogue move m is *legal in d* if it obeys the following rules:

- R_1: $\mathsf{pl}(m) \in \mathcal{P}$ (only d's participants are allowed to make dialogue moves)
- R_2: d must be single-topic
- R_3: if m replies to $m' \in M$, then $\mathsf{pl}(m) \neq \mathsf{pl}(m')$
- R_4: there is no $m' \in M$ with the same target and content (i.e. no repetition)
- R_5: for any $m' \in M$ that surrenders to a dialogue move in $\mathsf{target}(m)$, m is not an attacking counterpart of m'.

We will use these rules as a basis, except for rule R_3. Rule R_3 states that a participant is never allowed to reply to one of his own dialogue moves. Suppose that during a dialogue, a participant α claims argument A. After a while, the other participants have not attacked α's claim, but α has learned new facts and now also has constructed argument B which successfully attacks A. Because of the rule that participants cannot reply to their own dialogue moves, participant α cannot attack nor retract his own claim.

To determine the outcome of a dialogue, [8] considers two dialogical statuses of dialogue moves: warranted and unwarranted (this is called 'in' and 'out'). Furthermore, it is defined when a dialogue move *attacks* another dialogue move and when a dialogue move *surrenders* to another dialogue move. A move m's dialogical status is then determined using the dialogue statuses of the dialogue moves that attack and surrender to m. Because intuitively a statement subjective to some agent cannot be attacked by other agents, the definitions of when a dialogue move attacks another dialogue move need to be adapted.

If argument A undercuts B, then A concludes that there is an exception such that a defeasible inference rule in B cannot be applied. Because the application of an inference rule cannot be subjective, a dialogue move m_i claiming argument A attacks another dialogue move m_j claiming argument B if A undercuts B. Rebutting and undermining attacks do concern statements and therefore depend on whether the statement in question is subjective or objective. Therefore, if a dialogue move m_i questions or attacks a statement of m_j that is not subjective to the speaker of m_j, then m_i attacks m_j. On the other hand, if participant α first makes dialogue move m_i claiming argument A and later finds out that A is not justified, then α's dialogue move m_j of retracting his claim A attacks m_i. Finally, answering a why-question attacks the why-question.

Definition 15 (Attacking Dialogue Moves). *Let m and m' be two dialogue moves.* Dialogue move m' attacks m *if and only if m' replies to m and*

- $\mathsf{loc}(m) = \mathsf{claim}_i(A)$ *and* $\mathsf{loc}(m') = \mathsf{claim}_i(B)$ *such that either*
  - *B undercuts A,*
  - *B rebuts A on $A' \in \mathsf{sub}(A)$ s.t.* $\mathsf{conc}(A')$ *is not subjective to* $\mathsf{pl}(m)$*, or*
  - *B undermines A on a premise that is not subjective to* $\mathsf{pl}(m)$
- $\mathsf{loc}(m) = \mathsf{claim}_i(A)$ *and* $\mathsf{loc}(m') = \mathsf{why}_i(\phi)$ *such that ϕ is a premise of A and is not subjective to* $\mathsf{pl}(m)$
- $\mathsf{loc}(m) = \mathsf{claim}_i(A)$*,* $\mathsf{pl}(m) = \mathsf{pl}(m')$ *and* $\mathsf{loc}(m') = \mathsf{retract}_i(\mathsf{conc}(A))$
- $\mathsf{loc}(m) = \mathsf{why}_i(\phi)$ *and* $\mathsf{loc}(m') = \mathsf{claim}_i(B)$ *with* $\mathsf{conc}(B) = \phi$

Note that subjective statements cannot be attacked. In contrast to [8], because participants can reply to their own dialogue moves, they can retract a claim without the necessity of another participating agent having to question the claim first.

If a participant does not agree with a claim, then attacking that claim makes clear why he does not agree. This furthers the dialogue because now the other participants have more information and can respond appropriately. On the other hand, if participant α agrees with a claim of another participant, then α can concede that claim, which sets the claim's dialogical status to 'warranted'.

Definition 16 (Meta-Attacking Dialogue Moves). *Let m_1, m_2, m_3 be dialogue moves.* Dialogue move m_3 meta-attacks m_2 *if and only if*

- $\mathsf{loc}(m_1) = \mathsf{claim}_i(A)$*,* $\mathsf{loc}(m_2) = \mathsf{claim}_i(B)$*, and* m_2 *attacks* m_1*, and*
- $\mathsf{target}(m_3) = \{m_1, m_2\}$ *and* $\mathsf{loc}(m_3) = \mathsf{claim}_{i+1}(C)$ *with* $\mathsf{conc}(C) = B \prec A$

Definition 17 (Dialogical Status). *Let $d = \langle \mathcal{L_C}, \mathcal{P}, M \rangle$ be a dialogue. The* dialogical status of $m_i \in M$ *is* warranted *if and only if all attacking replies are not warranted or if there is a $m' \in M$ that replies to m such that* $\mathsf{pl}(m) \neq \mathsf{pl}(m')$ *and* $\mathsf{loc}(m') = \mathsf{concede}_i(\mathsf{conc}(A))$*.*

The notion of dialogical status is convenient to define rules in protocols. To keep the dialogue coherent, a notion of relevancy is required. First we will define when an argument is related to a dialogue, which depends on whether it is an object-level or meta-level argument. An object-level A is related only if A attacks or has the same conclusion an argument that has been uttered before. A meta-level argument B is related to d if the object-level arguments, formulae or inference rules to which B have been used before in the dialogue d or if B attacks or has the same conclusion as a meta-level argument that has been uttered before.

Definition 18 (Related Arguments). *Let d be a dialogue and* $\mathsf{Args}_i(d)$ *the arguments of level i in d.* Argument $A \in \mathsf{Args}(\mathcal{AS}_i)$ is related on level i to dialogue d *if*

- *either A attacks an argument in* $\mathsf{Args}_i(d)$ *or A has the same conclusion as some argument in* $\mathsf{Args}_i(d)$*, and*
- *if $i > 1$, then all terms in A that refer to elements in the argumentation system of level $i - 1$ must have been used previously in dialogue d*

To enforce the coherency of dialogues, a protocol could only allow claiming related arguments. Furthermore, a protocol could only allow dialogue moves that change the status of a previously uttered dialogue move. A result of this is that participants cannot give alternative arguments for the same conclusion because they do not change the status. This may stimulate that the participants give the most important argument first, which may promote the efficiency of the dialogue. However, there are also several disadvantages of such a protocol rule. Suppose agent α has been persuaded by agent β of ϕ being true in a dialogue. After the dialogue ended, α learns new information that overrules ϕ being true. However, if α would have gotten β's alternative arguments in favor of ϕ, then α may not have changed his belief w.r.t. ϕ. Furthermore, if more information is exchanged by allowing agents to give alternative arguments, then agents may discover new interesting arguments that could not have been constructed if agents were not allowed to give alternative arguments. Finally, in a deliberation or decision support dialogue, it is important that agents can describe all important aspects of their motivation so that other agents can find better joint actions or support their decision better. In a protocol that forbids alternative arguments these things are not possible.

4 Treating Incoming Arguments

Communicating is exchanging information, but if agents do not do anything with the information they get, communication is useless. A result of communication is that agents can get information from different sources, which may differ in reliability. In the introduction example, agent β may have learned that α likes gorgonzola and α may have learned from β that salad is healthier than pizza. However, α may have heard from another agent that pizza is just as healthy.

In existing approaches such as [11] and [7], the way how agents deal with incoming information is independent from its source. Furthermore, once some statement is added to the agent's knowledge base, it is impossible to trace back where it came from. This section proposes that the source information is stored on a meta-level where the agent can reason about the effect of the source on the conclusive force of arguments.

Section 4.1 proposes to represent the commitments and beliefs of agents in the meta-argumentation systems proposed in Section 2.2. Several argument schemes are proposed and formalized to infer what an agent believes from his commitments and to compare the conclusive force of arguments. This enables using epistemic approaches like the one in [3] for sophisticated reasoning about what other agents believe. Section 4.2 then proposes how an agent's knowledge base should be updated if he observes another agent making a dialogue move. Finally, Section 4.3 describes how an agent can select dialogue moves.

4.1 Meta Argumentation System

In this section, we will explain how the meta-argumentation systems we have proposed can be used by agents to reason about the conclusive force of arguments that they receive from other arguments. For this, several elements will be introduced. The binary predicates cm and b will be used in meta-argumentation systems to represent to what agents are committed to and what they believe. The predicate $\mathsf{cm}(\alpha, \phi)$ denotes that agent α is committed towards the object-level formula ϕ being true and the predicate $\mathsf{b}(\alpha, \phi)$ denotes that agent α believes that ϕ is true. The unary predicate $\mathsf{axiom}(\phi)$ denotes that ϕ is an axiom in the object-level argumentation system.

First, several inference rules are proposed to reason about beliefs and how the conclusive force of arguments compares. Then, a tower of argumentation systems is tailored for the envisioned dialogues by including these predicates and inference rules.

In general it will be the case that agents believe to what they commit themselves. The following argument scheme describes this intuition informally.

Argument Scheme 1: Commitment to Belief

Agent α is committed to that formula ϕ is true,

---

therefore, presumably, α believes that ϕ is true.

Critical questions for this argument scheme could question whether the agent has lied and whether the agent only has this commitment for the sake of the argument.

Several factors influence the conclusive force of an argument, e.g. the certainty of the premises, the strength of the inferences, or the reliability of the sources that are used. These different factors can be seen as criteria that contribute to the conclusive force of an argument. Therefore, how the conclusive force of arguments compares can be seen as a multi-criteria problem. The different criteria that contribute to an argument's conclusive force are typically incommensurable and therefore hard to combine. In [14, 13], an argumentation-based approach is proposed to combine incommensurable criteria.

Because each agent is the expert with respect to what is subjective to him, it should not be the case that the arguments of other agents that conflict with the agent's preferences have more conclusive force.

Argument Scheme 2: Subjectivity
Statement ϕ is subjective to agent α,
α believes ϕ is true,
β believes ψ is true which conflicts with ϕ,

---

therefore, presumably, the ϕ has more conclusive force than ψ.

Argument Scheme 1 is formalized with the defeasible inference rule d_{cm2b}. The constant me is used to denote the agent himself. Because an agent knows what he believes himself, this rule should only be used on other agents. The critical questions could be modeled by rules that undercut an application of this defeasible inference rule. Defeasible inference rule r_{sbj} formalizes Argument Scheme 2.

$$\begin{aligned} r_{\mathsf{cm2b}} &: \alpha \neq \mathsf{me}, \mathsf{cm}(\alpha, \phi) \Rightarrow \mathsf{b}(\alpha, \phi) \\ r_{sbj} &: \phi \in s(\alpha),\ \mathsf{b}(\alpha, \phi),\ \mathsf{b}(\beta, \psi),\ \psi \in \mathsf{cf}(\phi)\ \Rightarrow\ \psi \preceq \phi \end{aligned}$$

We will now introduce a tower of argumentation systems that is tailored for dialogues by including the proposed predicates and inference rules. The set Agents is used to denote the set of all agents and always contains the special element me which denotes the agent itself.

Definition 19 (Tower For Dialogues). *Let $\mathcal{T}_{\mathcal{AS}} = \{\mathcal{AS}_1, \ldots, \mathcal{AS}_n\}$ be a tower of argumentation systems and* Agents *the set containing all agents. We say that $\mathcal{T}_{\mathcal{AS}}$ is a* tower for dialogues *if for each $1 < i \leq n$ and $\mathcal{AS}_i = \langle \mathcal{L}_i, \mathcal{R}_i, \mathsf{cf} \rangle$:*

- *each agent in the set* Agents *is a constant in $\mathcal{L}_i$,*
- *$\mathcal{L}_i$ contains the unary predicate* axiom *and the binary predicates* cm *and* b*,*
- *$\mathcal{R}_i$ contains the defeasible inference rules r_{cm2b} and r_{sbj}.*

We want the ordering of arguments by conclusive force to be what is called 'admissible', i.e. arguments that are firm and strict have strictly more conclusive force than defeasible or plausible arguments, and a strict inference cannot increase the conclusive force of an argument.

Definition 20 (MAT For Dialogues). *Let $\mathcal{T}_{\mathcal{AS}}$ be a tower for dialogues of level n and* MAT $= \langle \mathcal{T}_{\mathcal{AS}}, \{\mathcal{K}_1, \ldots, \mathcal{K}_n\} \rangle$ *a meta-argumentation theory. We say that* MAT *is a* Meta-Argumentation Theory for dialogues *if for $1 < i \leq n$:*

– *$\mathcal{K}_i$ contains the axioms that ensure $\preceq$ is admissible,*
– *if* $\mathsf{b}(\mathsf{me}, \phi) \in \mathcal{K}_i$*, then* $\phi \in \mathcal{K}_{i-1}$*, and*
– *if ϕ is an axiom of level $i-1$, then* $\mathsf{axiom}(\phi) \in \mathcal{K}_i$*.*

The first constraint ensures that the axioms with respect to the conclusive force of arguments are in every MAT for dialogues. The second constraint ensures consistency between what the agent believes and what is in his knowledge base.

4.2 Observing Dialogue Moves

If an agent observes a dialogue move of another agent, then his knowledge base should be updated. This can be done in several ways. In [7], three so-called *acceptance attitudes* are proposed: (1) if α is *credulous*, then it accepts the conclusion of any sub-argument of previously asserted arguments; (2) if α is *cautious*, then it only accepts the conclusions of sub-arguments of previously asserted arguments if α has no attacking argument that is stronger; and, (3) if α is *skeptical*, then it only accepts conclusions of sub-arguments of previously asserted argument if that sub-argument would be acceptable.

If the agent observes another agent claiming an argument A of level i, then these different acceptance attitudes dictate whether the premises of A are added to the agent's knowledge base of level i. If the premises of argument A are added, then the agent can construct A for itself and possibly other new arguments. An EAF can then be built to determine what arguments are acceptable.

Regardless of whether the premises of the argument are added to the agent's knowledge base, the agent can update the speaker's commitments in the agent's meta-level knowledge base. Following [8], agents use the following commitment rules to update their knowledge bases when receiving a new dialogue move m. In contrast to [8], the commitments of agents are stored in the agent's knowledge base. For example, if the agent observes agent β claim argument A of level i, then the agent adds to his knowledge base of level $i+1$ that β is committed to A's premises and A's conclusion.

Definition 21 (Updating Commitments). *Let the agent's meta-argumentation theory be* $\mathsf{MAT} = \langle \mathcal{T}_{\mathcal{AS}}, \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}\rangle$*. If the agent observes dialogue move m on level $1 \leq i < n$, then* MAT *is updated to* $\langle \mathcal{T}_{\mathcal{AS}}, \{\mathcal{K}_1, \ldots, \mathcal{K}'_{i+1}, \ldots, \mathcal{K}_n\}\rangle$ *such that:*

– *if* $\mathsf{loc}(m) = \mathsf{claim}_i(A)$*, then* $\mathcal{K}'_{i+1} = \mathcal{K}_{i+1} \cup \{\mathsf{cm}(\mathsf{pl}(m), \phi) \mid \phi \in \mathsf{premises}(A)\} \cup \{\mathsf{cm}(\mathsf{pl}(m), \mathsf{conc}(A)\}$
– *if* $\mathsf{loc}(m) = \mathsf{why}_i(\phi)$*, then* $\mathcal{K}'_{i+1} = \mathcal{K}_{i+1}$*, i.e. nothing changes*
– *if* $\mathsf{loc}(m) = \mathsf{concede}(\phi)$*, then* $\mathcal{K}'_{i+1} = \mathcal{K}_{i+1} \cup \{\mathsf{cm}(\mathsf{pl}(m), \phi)\}$
– *if* $\mathsf{loc}(m) = \mathsf{retract}(\phi)$*, then* $\mathcal{K}'_{i+1} = \mathcal{K}_{i+1} \setminus \{\mathsf{cm}(\mathsf{pl}(m), \phi)\}$

Note that because we have a tower of finite 'height', commitments concerning formulae of the highest level argumentation system cannot be added because there is no argumentation system on top.

Using the defeasible inference rule from commitment to belief, the agent can construct arguments with respect to what the other agent believes. Furthermore, if the premises of the argument are added to the knowledge base, then the agent can reconstruct the received argument. Consequently, this argument will be in the updated argumentation framework of the agent. The arguments on the meta-level concerning the conclusive force of object-level arguments then have an effect on what arguments the agent accepts and rejects.

Example 1 (Pizza versus Salad). Consider the introduction example. The tower of dialogues is 3 high. We have the following statements on level 1: ϕ_1 denotes that agent α likes gorgonzola, ϕ_2 that α wants to eat pizza, ϕ_3 that salad is healthier than pizza, ϕ_4 that α wants to eat salad, and ϕ_5 that salad has less calories than pizza. Note that ϕ_1, ϕ_2 and ϕ_4 are subjective to α.

$$A_1 = \frac{\phi_1}{\phi_2} \qquad A_2 = \frac{\phi_3}{\phi_4} \qquad A_3 = \frac{\phi_5}{\phi_3}$$

Because α can only choose one action, ϕ_2 and ϕ_4 are contradictory. Consequently, A_1 and A_2 attack each other. Furthermore, meta-level statement $A_1 \prec A_2$ denotes that argument A_2 is stronger than A_1. Agent β starts with object-level knowledge base $\{\phi_3, \phi_5\}$ and meta-level knowledge base $\{A_1 \prec A_2\}$. The dialogue is as follows: (1) $m_1 = \langle 1, \alpha, \mathsf{claim}_1(A_1), \emptyset \rangle$, (2) $m_2 = \langle 2, \beta, \mathsf{why}_1(\phi_1), \{1\} \rangle$, (3) $m_3 = \langle 3, \alpha, \mathsf{claim}_1(\phi_1), \{2\} \rangle$, (4) $m_4 = \langle 4, \beta, \mathsf{claim}_1(A_2), \{1\} \rangle$, (5) $m_5 = \langle 5, \beta, \mathsf{claim}_2(A_1 \prec A_2), \{1, 4\} \rangle$, (6) $m_6 = \langle 6, \alpha, \mathsf{why}_1(\phi_3), \{4\} \rangle$, and (7) $m_7 = \langle 7, \beta, \mathsf{claim}_1(A_3), \{6\} \rangle$ Note that move m_5's claim is related to the dialogue, but it does not change the status of any move. Table 1 shows how agent β's knowledge base is updated during the dialogue, where $\mathcal{K}_i$ denotes the object-level knowledge base after dialogue move i, $\mathcal{K}'_i$ the meta-level knowledge base and $\mathcal{K}''_i$ the meta-meta-level knowledge base.

Table 1. Updating the Knowledge Base in a Dialogue

$\mathcal{K}$	$\mathcal{K}'$	$\mathcal{K}''$
$\mathcal{K}_0 = \{\phi_3, \phi_5\}$	$\mathcal{K}'_0 = \{A_1 \prec A_2, \mathsf{b}(\beta, \phi_3), \mathsf{b}(\beta, \phi_5)\}$	$\mathcal{K}''_0 = \mathsf{b}(\beta, A_1 \prec A_2)$
$\mathcal{K}_1 = \mathcal{K}_0 \cup \{\phi_1, \phi_2\}$	$\mathcal{K}'_1 = \mathcal{K}'_0 \cup \{\mathsf{cm}(\alpha, \phi_1), \mathsf{cm}(\alpha, \phi_2\}$	$\mathcal{K}''_1 = \mathcal{K}''_0$
$\mathcal{K}_2 = \mathcal{K}_1$	$\mathcal{K}'_2 = \mathcal{K}'_1$	$\mathcal{K}''_2 = \mathcal{K}''_1$
$\mathcal{K}_3 = \mathcal{K}_2$	$\mathcal{K}'_3 = \mathcal{K}'_2$	$\mathcal{K}''_3 = \mathcal{K}''_2$
$\mathcal{K}_4 = \mathcal{K}_3$	$\mathcal{K}'_4 = \mathcal{K}'_3 \cup \{\mathsf{cm}(\beta, \phi_3), \mathsf{cm}(\beta, \phi_4\}$	$\mathcal{K}''_4 = \mathcal{K}''_3$
$\mathcal{K}_5 = \mathcal{K}_4$	$\mathcal{K}'_5 = \mathcal{K}'_4$	$\mathcal{K}''_5 = \mathcal{K}''_4 \cup \{\mathsf{cm}(\beta, A_1 \prec A_2)\}$
$\mathcal{K}_6 = \mathcal{K}_5$	$\mathcal{K}'_6 = \mathcal{K}'_5$	$\mathcal{K}''_6 = \mathcal{K}''_5$
$\mathcal{K}_7 = \mathcal{K}_6$	$\mathcal{K}'_7 = \mathcal{K}'_6 \cup \{\mathsf{cm}(\beta, \phi_5)\}$	$\mathcal{K}''_7 = \mathcal{K}''_6$

After dialogue move 1, both arguments A_1 and A_2 can be constructed. However, because $A_1 \prec A_2$ is in the meta-knowledge base, A_1's attack on A_2 is unsuccessful. Because the meta-level knowledge base stores what β believes and other agents' commitments, β can use all this information to reason about the relative strength of object-level arguments. If later β finds out that α was lying about whether he likes gorgonzola, then the relative strength of arguments using this automatically changes.

4.3 Dialogue Move Selection

Given a dialogue, the protocol determines what dialogue moves are legal, but a participating agent should also determine what moves are interesting for him to make. If at a given point in the dialogue multiple dialogue moves are interesting, then the agent should be able to make a decision about what dialogue move to make. In the introduction example, after α made the initial claim, β had to decide between asking why α likes gorgonzola or immediately giving the counterargument of eating the salad. If the agent can select from multiple arguments that he could claim, then an argument selection mechanism like the one proposed in [13] could be used.

In [7], three different kinds of so-called *assertion attitudes* are proposed, which agents can use to determine whether they will assert a proposition in a dialogue. These attitudes can be adapted to the formalism in this paper w.r.t. a semantics S as follows: (1) if the agent is *confident*, then he can claim any argument he can construct; (2) if the agent is *careful*, then he can claim any argument that is defensible or justified under S; and, (3) if the agent is *thoughtful*, then he can claim any argument that is justified under S.

Suppose that the agent is participating in dialogue d and that the agent has updated its meta-argumentation theory and corresponding EAF with all the dialogue moves that have been made. Using Definition 13, the agent can extract the set of arguments in d. For each of those arguments, the agent can compare the argument's status in the dialogue to the argument's status in his own argumentation framework. If these statuses of an argument correspond, then the dialogue and the agent agree on A. If these statuses of an argument do not correspond, then there is a need for the agent to make a dialogue move. Table 2 shows the differences between the dialogical status of a dialogue move claiming an argument A and A's status in the agent's EAF.

Table 2. Dialogical Status versus the Status in an Agent's EAF

	Justified	**Defensible**	**Overruled**	**Invalid**
Warranted	Agree	Weakly agree	Disagree	Disagree
Not Warranted	Disagree	Weakly disagree	Agree	Agree

Depending on the agent's acceptance attitude, it is possible that an agent cannot reconstruct an argument that has been claimed in the dialogue because its premises are not in the agent's knowledge base. Such an argument is then *invalid* for the agent. For each premise ϕ of an invalid argument, if the agent has no argument concluding ϕ, then the dialogue move of asking why ϕ is *interesting*.

If another participant has claimed argument A and A's conclusion is justified under S in the agent's EAF, then the dialogue move of conceding with A's conclusion is *interesting*. If the agent himself is committed to ϕ, but ϕ is an overruled conclusion under S in the agent's EAF, then the dialogue move of retracting ϕ is *interesting*. If the agent is careful, then he should also retract a formula if it is a defensible conclusion under S in his EAF.

Let m be an uttered dialogue move that claims argument A. If the agent's EAF and the dialogical status of m disagree, then dialogue moves that attack m are *interesting*. If the agent's EAF and the dialogical status weakly agree, then a confident agent should be *interested* in moves attacking m and a thoughtful agent should not reply to m.

5 Conclusion

In this paper we have presented an abstract formalism for reasoning about preferences between arguments using the commitments that other agents make in dialogue moves. Furthermore, a dialogue framework is proposed in which preferences between arguments can be discussed.

In [5], ASPIC+ is further developed to define attacks on attacks using an abstract function. The meta-argumentation approach of Section 2.2 can be seen as an instantiation of this abstract function. Section 3 extends the dialogue framework of [8] such that agents can give meta-arguments and a protocol is proposed that treats subjective and objective statements differently. Agents can discuss their preferences between arguments using meta-arguments. Finally, Section 4 proposed to represent the sources of information on a meta-level such that agents can argue about the reliability of sources and its impact on the conclusive force of arguments. This approach is more robust against manipulation because the agent can reason about trustworthiness of sources and its effect on arguments. Also, if new information is obtained or if an agent retracts a claim, then it is straightforward to update the knowledge base.

In many domains it is common that there are multiple arguments with the same conclusion. For example, when making a decision, there may be multiple arguments in favor and against a decision, or when determining the conclusive force of an argument, there may be multiple sources that believe some premise. In such cases, arguments need to be accrued. Several approaches such as [9] and [4] address the accrual of arguments, which needs to be added to our framework.

By using towers of argumentation systems, there is a risk that a tower of infinite height is required because it is always possible to reason about preferences between arguments on a higher level. This issue should be addressed. In this paper we have only addressed comparing the conclusive force of arguments, but meta-argumentation systems could also be used to reason about whether an argument has an acceptable amount of conclusive force (a proof standard) to be even considered. Adding this requires that meta-arguments can attack object-arguments. The effect of this is that the resulting EAF is not hierarchical.

Bibliography

[1] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.

[2] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[3] J.-J. Ch. Meyer and W. Van Der Hoek. *Epistemic logic for AI and computer science*. Cambridge Univ Pr, 2004.

[4] S. Modgil and T. Bench-Capon. Integrating Dialectical and Accrual Modes of Argumentation. In *3rd International Conference on Computational Models of Argument (COMMA 2010)*, 2010.

[5] S. Modgil and H. Prakken. Reasoning about preferences in structured extended argumentation frameworks. In Giacomin & Simari Baroni, Cerutti, editor, *Computational Models of Argument. Proc. of COMMA 2010*, pages 347–358. IOS Press, 2010.

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

[7] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[8] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009, 2005.

[9] H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th International Conference on A.I. and Law*, pages 85–94. ACM NY, USA, 2005.

[10] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[11] C. Sierra, N. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pages 177–192, 1998.

[12] G.A.W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1-2):225–279, 1997.

[13] T. L. van der Weide, F. Dignum, J.-J. Ch. Meyer, H. Prakken, and G. Vreeswijk. Multi-criteria argument selection in persuasion dialogues. In Stone Yolum, Turner and Sonenberg, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, to appear.

[14] T.L. van der Weide, F. Dignum, J.-J. Ch. Meyer, H. Prakken, and G. A. W. Vreeswijk. Arguing about preferences and decisions. In *Proc. of the 7th Int. Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*, 2010.

[15] M. Wooldridge, P. McBurney, and S. Parsons. On the meta-logic of arguments. In *Argumentation in Multi-Agent Systems 2005*, volume 4049/2006 of *LNCS*, pages 42–56. Springer Berlin / Heidelberg, 2005.