AAMAS 2010 TORONTO

The 9th International Conference on Autonomous Agents and Multiagent Systems
May 10-14, 2010
Toronto, Canada

Workshop 10

The Seventh International Workshop on Argumentation in Multi-Agent Systems

ArgMAS 2010

Editors:
Wiebe van der Hoek
Gal A. Kaminka
Yves Lespérance
Michael Luck
Sandip Sen

ArgMAS 2010

Seventh International Workshop

on

Argumentation in Multi-Agent Systems

Toronto, Canada, May 2010

In conjunction with AAMAS 2010

Workshop Proceedings

Editors:

Peter McBurney, Iyad Rahwan and Simon Parsons

ArgMAS 2010 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
Frank Dignum, Utrecht University, The Netherlands
Yannis Dimopoulos, University of Cyprus, Cyprus
Sylvie Doutre, University of Toulouse 1, France
Rogier van Eijk, Utrecht University, The Netherlands
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, University of Liverpool, UK
Jarred McGinnis, London, UK
Sanjay Modgil, Imperial College London, UK
Pavlos Moraitis, Paris Descartes University, France
Tim Norman, University of Aberdeen, Scotland, UK
Nir Oren, King's College London, UK
Fabio Paglieri, ISTC-CNR, Roma IT
Simon Parsons, Brooklyn College, City University of New York, USA
Enric Plaza, Spanish Scientific Research Council, Spain
Henri Prade, IRIT, Toulouse, France
Henry Prakken, Utrecht University, & University of Groningen, The Netherlands
Iyad Rahwan, Masdar Institute, UAE, & University of Edinburgh, Scotland, UK
Chris Reed, University of Dundee, Scotland, UK
Michael Rovatsos, University of Edinburgh, 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, Scotland, UK
Michael Wooldridge, University of Liverpool, UK.

ArgMAS STEERING COMMITTEE

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

Contents

Preface

Welcome to the seventh edition of the *International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*, being held in Toronto, Canada, in association with the Ninth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010). Previous ArgMAS workshops have been held in New York City (2004), Utrecht (2005), Hakodate (2006), Honolulu (2007), Estoril (2008), and Budapest (2009), 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 14 papers and position statements selected following a peer-review process. We thank all authors who made submissions to ArgMAS 2010, and we thank the members of the Programme Committee for their efforts in reviewing the papers submitted. The papers presented at the workshop are assembled in this document in alphabetical order of the surname of the first author.

Our invited speaker this year is well-known argumentation-theorist and philosopher Professor David Hitchcock of McMaster University, Hamilton, Ontario, Canada, who will talk on instrumental rationality. We are honoured by Professor Hitchcock's participation, and we thank him for giving the keynote address.

Following a successful trial in 2009, we plan again this year to have selected official respondents offer short critiques to several of the papers presented at ArgMAS 2010. We have adopted this innovation from conferences in Philosophy, where it is standard, and we found it worked will in Budapest 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 Toronto.

Peter McBurney, Iyad Rahwan and Simon Parsons

Programme Co-Chairs

May 2010

Agreeing what to do

Elizabeth Black[1] and Katie Atkinson[2]

[1] Department of Engineering Science, University of Oxford, UK
lizblack@robots.ox.ac.uk
[2] Department of Computer Science, University of Liverpool, UK
katie@liverpool.ac.uk

Abstract. When deliberating about what to do, an autonomous agent must generate and consider the relative pros and cons of the different options. The situation becomes even more complicated when an agent is involved in a joint deliberation, as each agent will have its own preferred outcome which may change as new information is received from the other agents involved in the deliberation. We present an argumentation-based dialogue system that allows agents to come to an agreement on how to act in order to achieve a joint goal. The dialogue strategy that we define ensures that any agreement reached is acceptable to each agent, but does not necessarily demand that the agents resolve or share their differing preferences. We give properties of our system and discuss possible extensions.

ACM Category: I.2.11 Multiagent systems. **General terms:** Theory.
Keywords: dialogue, argumentation, agreement, strategy, deliberation, action.

1 Introduction

When agents engage in dialogues their behaviour is influenced by a number of factors including the type of dialogue taking place (e.g. negotiation or inquiry), the agents' own interests within the dialogue, and the other parties participating in the dialogue. Some of these aspects have been recognised in Walton and Krabbe's characterisation of dialogue types [1]. Some types of dialogue are more adversarial than others. For example, in a persuasion dialogue an agent may try to force its opponent to contradict itself, thus weakening the opponent's position. In a deliberation dialogue, however, the agents are more co-operative as they each share the same goal to establish agreement, although individually they may wish to influence the outcome in their own favour.

We present a dialogue system for deliberation that allows agents to reason and argue about what to do to achieve some joint goal but does not require them to pool their knowledge, nor does it require them to aggregate their preferences. Few existing dialogue systems address the problem of deliberation ([2, 3] are notable exceptions). Ours is the first system for deliberation that provides a dialogue strategy that allows agents to come to an agreement about how to act that each is happy with, despite the fact that they may have different preferences and thus may each be agreeing for different reasons; it couples a dialectical setting with formal methods for argument evaluation and allows strategic manoeuvring in order to influence the dialogue outcome. We present an analysis of when agreement can and cannot be reached with our system; this provides an essential foundation to allow us to explore mechanisms that allow agents to come to an agreement in situations where the system presented here may fail.

We assume that agents are co-operative in that they do not mislead one another and will come to an agreement wherever possible; however, each agent aims to satisfy its own preferences. For the sake of simplicity, here we present a two party dialogue; however, the assumed co-operative setting means that many of the difficult issues which normally arise with multi party dialogues (e.g. [4]) are avoided here. We believe it to be straightforward to extend the system to allow multiple participants, for example following the approach taken in [5].

We describe the setting envisaged through a characteristic scenario. Consider a situation where a group of colleagues is attending a conference and they would all like to go out for dinner together. Inevitably, a deliberation takes place where options are proposed and critiqued and each individual will have his own preferences that he wishes to be satisfied by the group's decision. It is likely that there will be a range of different options proposed that are based on criteria such as: the type of cuisine desired; the proximity of the restaurant; the expense involved; the restaurant's capacity; etc.

To start the dialogue one party may put forward a particular proposal, reflecting his own preferences, say going to a French restaurant in the town centre. Such an argument may be attacked on numerous grounds, such as it being a taxi ride away, or it being expensive. If expense is a particular consideration for some members of the party, then alternative options would have to be proposed, each of which may have its own merits and disadvantages, and may need to consider the preferences already expressed. We can see that in such a scenario the agents , whilst each having their own preferred options, are committed to finding an outcome that everyone can agree to.

We present a formal argumentation-based dialogue system to handle joint deliberation. In section 2 we present the reasoning mechanism through which agents can construct and propose arguments about action. In section 3 we define the dialogue system and give an example dialogue. In section 4 we present an analysis of our system and in section 5 we discuss important extensions. In section 6 we discuss related work, and we conclude the paper in section 7.

2 Practical arguments

We now describe the model of argumentation that we use to allow agents to reason about how to act. Our account is based upon a popular approach to argument characterisation, whereby argumentation schemes and critical questions are used as presumptive justification for generating arguments and attacks between them [6]. Arguments are generated by an agent instantiating a *scheme for practical reasoning* which makes explicit the following elements: the initial circumstances where action is required; the action to be taken; the new circumstances that arise through acting; the goal to be achieved; and the social value promoted by realising the goal in this way. The scheme is associated with a set of characteristic critical questions (CQs) that can be used to identify challenges to proposals for action that instantiate the scheme. An unfavourable answer to a CQ will identify a potential flaw in the argument. Since the scheme makes use of what are termed as 'values', this caters for arguments based on subjective preferences as well as more objective facts. Such values represent qualitative social interests that an agent wishes (or does not wish) to uphold by realising the goal stated [7].

To enable the practical argument scheme and critical questions approach to be precisely formalised for use in automated systems, in [8] it was defined in terms of an Action-based Alternating Transition System (AATS) [9], which is a structure for modelling game-like multi-agent systems where the agents can perform actions in order to attempt to control the system in some way. Whilst the formalisms given in [8, 9] are intended to represent the overall behaviour of a multi-agent system and the effects of joint actions performed by the agents, we are interested in representing the knowledge of individual agents within a system. Hence, we use an adaptation of their formalisms (first presented in [5]) to define a *Value-based Transition System* (VATS) as follows.

Definition 1: *A* **Value-based Transition System** *(***VATS***), for an agent x, denoted S^x, is a 9-tuple $\langle Q^x, q_0^x, Ac^x, Av^x, \rho^x, \tau^x, \Phi^x, \pi^x, \delta^x \rangle$ s.t.:*

Q^x is a finite set of states*;*

$q_0^x \in Q^x$ is the designated initial state*;*

Ac^x is a finite set of actions*;*

Av^x is a finite set of values*;*

$\rho^x : Ac^x \mapsto 2^{Q^x}$ is an action precondition function*, which for each action $a \in Ac^x$ defines the set of states $\rho(a)$ from which a may be executed;*

$\tau^x : Q^x \times Ac^x \mapsto Q^x$ is a partial system transition function*, which defines the state $\tau^x(q, a)$ that would result by the performance of a from state q—n.b. as this function is partial, not all actions are possible in all states (cf. the precondition function above);*

Φ^x is a finite set of atomic propositions*;*

$\pi^x : Q^x \mapsto 2^{\Phi^x}$ is an interpretation function*, which gives the set of primitive propositions satisfied in each state: if $p \in \pi^x(q)$, then this means that the propositional variable p is satisfied (equivalently, true) in state q; and*

$\delta^x : Q^x \times Q^x \times Av^x \mapsto \{+, -, =\}$ is a valuation function*, which defines the* status *(promoted (+), demoted (−), or neutral (=)) of a value $v \in Av^x$ ascribed by the agent to the transition between two states: $\delta^x(q, q', v)$ labels the transition between q and q' with respect to the value $v \in Av^x$.*

Note, $Q^x = \emptyset \leftrightarrow Ac^x = \emptyset \leftrightarrow Av^x = \emptyset \leftrightarrow \Phi^x = \emptyset$.

Given its VATS, an agent can now instantiate the practical reasoning argument scheme in order to construct arguments for (or against) actions to achieve a particular goal because they promote (or demote) a particular value.

Definition 2: *An* **argument** *constructed by an agent x from its VATS S^x is a 4-tuple $A = \langle a, p, v, s \rangle$ s.t.: $q_x = q_0^x$; $a \in Ac^x$; $\tau^x(q_x, a) = q_y$; $p \in \pi^x(q_y)$; $v \in Av^x$; $\delta^x(q_x, q_y, v) = s$ where $s \in \{+, -\}$.*
We define the functions: $\mathsf{Act}(A) = a$; $\mathsf{Goal}(A) = p$; $\mathsf{Val}(A) = v$; $\mathsf{Sign}(A) = s$.
If $\mathsf{Sign}(A) = +(-resp.)$, then we say A is an argument **for** *(***against** *resp.) action a.*
We denote the **set of all arguments an agent** *x* **can construct from** *S^x as $Args^x$; we let $Args_p^x = \{A \in Args^x \mid \mathsf{Goal}(A) = p\}$.*
The set of **values** *for a set of arguments $\mathcal{X}$ is defined as $\mathsf{Vals}(\mathcal{X}) = \{v \mid A \in \mathcal{X}$ and $\mathsf{Val}(A) = v\}$.*

If we take a particular argument for an action, it is possible to generate attacks on that argument by posing the various CQs related to the practical reasoning argument scheme. In [8], details are given of how the reasoning with the argument scheme and posing CQs is split into three stages: *problem formulation*, where the agents decide on the facts and values relevant to the particular situation under consideration; *epistemic reasoning*, where the agents determine the current situation with respect to the structure formed at the previous stage; and *action selection*, where the agents develop, and evaluate, arguments and counter arguments about what to do. Here, we assume that the agents' problem formulation and epistemic reasoning are sound and that there is no dispute between them relating to these stages; hence, we do not consider the CQs that arise in these stages. That leaves CQ5-CQ11 for consideration (as numbered in [8]):

CQ5: Are there alternative ways of realising the same consequences?

CQ6: Are there alternative ways of realising the same goal?

CQ7: Are there alternative ways of promoting the same value?

CQ8: Does doing the action have a side effect which demotes the value?

CQ9: Does doing the action have a side effect which demotes some other value?

CQ10: Does doing the action promote some other value?

CQ11: Does doing the action preclude some other action which would promote some other value?

We do not consider CQ5 or CQ11 further, as the focus of the dialogue is to agree to an action that achieves the *goal*; hence, the incidental consequences (CQ5) and other potentially precluded actions (CQ11) are of no interest. We focus instead on CQ6-CQ10; agents participating in a deliberation dialogue use these CQs to identify attacks on proposed arguments for action. These CQs generate a set of arguments for and against different actions to achieve a particular goal, where each argument is associated with a motivating value. To evaluate the status of these arguments we use a Value Based Argumentation Framework (VAF), introduced in [7]. A VAF is an extension of the argumentation frameworks (AF) of Dung [10]. In an AF an argument is admissible with respect to a set of arguments S if all of its attackers are attacked by some argument in S, and no argument in S attacks an argument in S. In a VAF an argument succeeds in defeating an argument it attacks only if its value is ranked as high, or higher, than the value of the argument attacked; a particular ordering of the values is characterised as an *audience*. Arguments in a VAF are admissible with respect to an audience A and a set of arguments S if they are admissible with respect to S in the AF which results from removing all the attacks which are unsuccessful given the audience A. A maximal admissible set of a VAF is known as a *preferred extension*.

Although VAFs are commonly defined abstractly, here we give an instantiation in which we define the attack relation between the arguments. Condition 1 of the following attack relation allows for CQ8 and CQ9; condition 2 allows for CQ10; condition 3 allows for CQ6 and CQ7. Note that attacks generated by condition 1 are not symmetrical, whilst those generated by conditions 2 and 3 are.

Definition 3: *An* **instantiated value-based argumentation framework** *(***iVAF***) is defined by a tuple $\langle \mathcal{X}, \mathcal{A} \rangle$ s.t. $\mathcal{X}$ is a finite set of arguments and $\mathcal{A} \subset \mathcal{X} \times \mathcal{X}$ is the* **attack relation***. A pair $(A_i, A_j) \in \mathcal{A}$ is referred to as "A_i attacks A_j" or "A_j is attacked by*

A_i". For two arguments $A_i = \langle a, p, v, s\rangle$, $A_j = \langle a', p', v', s'\rangle \in \mathcal{X}$, $(A_i, A_j) \in \mathcal{A}$ iff $p = p'$ and either:

1. $a = a'$, $s = -$ and $s' = +$; or
2. $a = a'$, $v \neq v'$ and $s = s' = +$; or
3. $a \neq a'$ and $s = s' = +$.

An **audience** *for an agent x over the values V is a binary relation $\mathcal{R}^x \subset V \times V$ that defines a* total order *over V. We say that an argument A_i is* **preferred to** *the argument A_j in the audience $\mathcal{R}^x$, denoted $A_i \succ_x A_j$, iff $(\mathsf{Val}(A_i), (\mathsf{Val}(A_j)) \in \mathcal{R}^x$. If R^x is an audience over the values V for the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$, then $\mathsf{Vals}(\mathcal{X}) \subseteq V$.*

We use the term audience here to be consistent with the literature, it does not refer to the preference of a *set* of agents; rather, we define it to represent a particular agent's preference over a set of values.

Given an iVAF and a particular agent's audience, we can determine acceptability of an argument as follows. Note that if an attack is symmetric, then an attack only succeeds in defeat if the attacker is more preferred than the argument being attacked; however, as in [7], if an attack is asymmetric, then an attack succeeds in defeat if the attacker is at least as preferred at the argument being attacked.

Definition 4: *Let $\mathcal{R}^x$ be an audience and let $\langle \mathcal{X}, \mathcal{A}\rangle$ be an iVAF.*

For $(A_i, A_j) \in \mathcal{A}$ s.t. $(A_j, A_i) \notin \mathcal{A}$, A_i **defeats** *A_j under $\mathcal{R}^x$ if $A_j \not\succ_x A_i$.*

For $(A_i, A_j) \in \mathcal{A}$ s.t. $(A_j, A_i) \in \mathcal{A}$, A_i **defeats** *A_j under $\mathcal{R}^x$ if $A_i \succ_x A_j$.*

An argument $A_i \in \mathcal{X}$ is **acceptable w.r.t** *S under $\mathcal{R}^x$ ($S \subseteq \mathcal{X}$) if: for every $A_j \in \mathcal{X}$ that defeats A_i under $\mathcal{R}^x$, there is some $A_k \in S$ that defeats A_j under $\mathcal{R}^x$.*

A subset S of $\mathcal{X}$ is **conflict-free** *under $\mathcal{R}^x$ if no argument $A_i \in S$ defeats another argument $A_j \in S$ under $\mathcal{R}^x$.*

A subset S of $\mathcal{X}$ is **admissible** *under $\mathcal{R}^x$ if: S is conflict-free in $\mathcal{R}^x$ and every $A \in S$ is acceptable w.r.t S under $\mathcal{R}^x$.*

A subset S of $\mathcal{X}$ is a **preferred extension** *under $\mathcal{R}^x$ if it is a maximal admissible set under $\mathcal{R}^x$.*

An argument A is **acceptable** *in the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$ under audience $\mathcal{R}^x$ if there is* some *preferred extension containing it.*

We have now defined a mechanism with which an agent can determine attacks between arguments for and against actions, and can then use an ordering over the values that motivate such arguments (its audience) in order to determine their acceptability. In the next section we define our dialogue system.

3 Dialogue system

The communicative acts in a dialogue are called *moves*. We assume that there are always exactly two agents (*participants*) taking part in a dialogue, each with its own identifier taken from the set $\mathcal{I} = \{1, 2\}$. Each participant takes it in turn to make a move to the other participant. We refer to participants using the variables x and $\overline{x}$ such that: x is 1 if and only if $\overline{x}$ is 2; x is 2 if and only if $\overline{x}$ is 1.

Move	**Format**
open	$\langle x, \mathsf{open}, \gamma\rangle$
assert	$\langle x, \mathsf{assert}, A\rangle$
agree	$\langle x, \mathsf{agree}, a\rangle$
close	$\langle x, \mathsf{close}, \gamma\rangle$

Table 1. Format for moves used in deliberation dialogues: γ is a goal; a is an action; A is an argument; $x \in \{1, 2\}$ is an agent identifier.

A move in our system is of the form $\langle Agent, Act, Content\rangle$. $Agent$ is the identifier of the agent generating the move, Act is the type of move, and the $Content$ gives the details of the move. The format for moves used in deliberation dialogues is shown in Table 1, and the set of all moves meeting the format defined in Table 1 is denoted $\mathcal{M}$. Note that the system allows for other types of dialogues to be generated and these might require the addition of extra moves. Also, $\mathsf{Sender} : \mathcal{M} \mapsto \mathcal{I}$ is a function such that $\mathsf{Sender}(\langle Agent, Act, Content\rangle) = Agent$.

We now informally explain the different types of move: an *open* move $\langle x, \mathsf{open}, \gamma\rangle$ opens a dialogue to agree on an action to achieve the goal γ; an *assert* move $\langle x, \mathsf{assert}, A\rangle$ asserts an argument A for or against an action to achieve a goal that is the topic of the dialogue; an *agree* move $\langle x, \mathsf{agree}, a\rangle$ indicates that x agrees to performing action a to achieve the topic; a *close* move $\langle x, \mathsf{close}, \gamma\rangle$ indicates that x wishes to end the dialogue.

A dialogue is simply a sequence of moves, each of which is made from one participant to the other. As a dialogue progresses over time, we denote each timepoint by a natural number. Each move is indexed by the timepoint when the move was made. Exactly one move is made at each timepoint.

Definition 5: *A* **dialogue**, *denoted D^t, is a sequence of moves $[m_1, \ldots, m_t]$ involving two participants in $\mathcal{I} = \{1, 2\}$, where $t \in \mathbb{N}$ and the following conditions hold:*

1. *m_1 is a move of the form $\langle x, \mathsf{open}, \gamma\rangle$ where $x \in \mathcal{I}$*
2. *$\mathsf{Sender}(m_s) \in \mathcal{I}$ for $1 \leq s \leq t$*
3. *$\mathsf{Sender}(m_s) \neq \mathsf{Sender}(m_{s+1})$ for $1 \leq s < t$*

The **topic** *of the dialogue D^t is returned by $\mathsf{Topic}(D^t) = \gamma$. The set of all dialogues is denoted $\mathcal{D}$.*

The first move of a dialogue D^t must always be an open move (condition 1 of the previous definition), every move of the dialogue must be made by a participant (condition 2), and the agents take it in turns to send moves (condition 3). In order to terminate a dialogue, either: two close moves must appear one immediately after the other in the sequence (a *matched-close*); or two moves agreeing to the same action must appear one immediately after the other in the sequence (an *agreed-close*).

Definition 6: *Let D^t be a dialogue s.t. $\mathsf{Topic}(D^t) = \gamma$. We say that m_s ($1 < s \leq t$), is*

- *a* **matched-close for** *D^t iff $m_{s-1} = \langle x, \mathsf{close}, \gamma\rangle$ and $m_s = \langle \overline{x}, \mathsf{close}, \gamma\rangle$.*
- *an* **agreed-close for** *D^t iff $m_{s-1} = \langle x, \mathsf{agree}, a\rangle$ and $m_s = \langle \overline{x}, \mathsf{agree}, a\rangle$.*

We say D^t has a **failed outcome** *iff m_t is a matched-close, whereas we say D^t has a* **successful outcome** *of a iff $m_t = \langle x, \mathsf{agree}, a\rangle$ is an agreed-close.*

So a matched-close or an agreed-close will terminate a dialogue D^t but only if D^t has not already terminated.

Definition 7: *Let D^t be a dialogue. D^t* **terminates at** *t iff m_t is a matched-close or an agreed-close for D^t and $\neg\exists s$ s.t. $s < t$, D^t* **extends** *D^s (i.e. the first s moves of D^t are the same as the sequence D^s) and D^s terminates at s.*

We shortly give the particular protocol and strategy functions that allow agents to generate deliberation dialogues. First, we introduce some subsidiary definitions. At any point in a dialogue, an agent x can construct an iVAF from the union of the arguments it can construct from its VATS and the arguments that have been asserted by the other agent; we call this x's *dialogue iVAF*.

Definition 8: *A* **dialogue iVAF** *for an agent x participating in a dialogue D^t is denoted* dVAF(x, D^t). *If D^t is the sequence of moves $= [m_1, \ldots, m_t]$, then* dVAF(x, D^t) *is the iVAF $\langle \mathcal{X}, \mathcal{A} \rangle$ where $\mathcal{X} = Args^x_{\mathsf{Topic}(D^t)} \cup \{A \mid \exists m_k = \langle \overline{x}, assert, A \rangle (1 \leq k \leq t)\}$.*

An action is *agreeable* to an agent x if and only if there is some argument *for* that action that is acceptable in x's dialogue iVAF under the audience that represents x's preference over values. Note that the set of actions that are agreeable to an agent may change over the course of the dialogue.

Definition 9: *An action a is* **agreeable** *in the iVAF $\langle \mathcal{X}, \mathcal{A} \rangle$ under the audience $\mathcal{R}^x$ iff $\exists A = \langle a, \gamma, v, + \rangle \in \mathcal{X}$ s.t. A is acceptable in $\langle \mathcal{X}, \mathcal{A} \rangle$ under $\mathcal{R}^x$. We denote the* **set of all actions that are agreeable to an agent** *x* **participating in a dialogue** *D^t as* $\mathsf{AgActs}(x, D^t)$, *s.t. $a \in \mathsf{AgActs}(x, D^t)$ iff a is agreeable in* dVAF(x, D^t) *under $\mathcal{R}^x$.*

A protocol is a function that returns the set of moves that are permissible for an agent to make at each point in a particular type of dialogue. Here we give a deliberation protocol. It takes the dialogue that the agents are participating in and the identifier of the agent whose turn it is to move, and returns the set of permissible moves.

Definition 10: *The* **deliberation protocol** *for agent x is a function $\mathsf{Protocol}_x : \mathcal{D} \mapsto \wp(\mathcal{M})$. Let D^t be a dialogue ($1 \leq t$) with participants $\{1, 2\}$ s.t. $\mathsf{Sender}(m_t) = \overline{x}$ and $\mathsf{Topic}(D^t) = \gamma$.*

$$\mathsf{Protocol}_x(D^t) = P^{\mathsf{ass}}_x(D^t) \cup P^{\mathsf{ag}}_x(D^t) \cup \{\langle x, \mathsf{close}, \gamma \rangle\}$$

where the following are sets of moves and $x' \in \{1, 2\}$.

$$\begin{aligned} P^{\mathsf{ass}}_x(D^t) = \{ & \langle x, \mathsf{assert}, A \rangle \mid \mathsf{Goal}(A) = \gamma \\ & \textbf{and} \\ & \neg\exists m_{t'} = \langle x', assert, A \rangle (1 < t' \leq t) \} \end{aligned}$$

$$\begin{aligned} P^{\mathsf{ag}}_x(D^t) = \{ & \langle x, \mathsf{agree}, a \rangle \mid \textbf{ either} \\ & (1) m_t = \langle \overline{x}, agree, a \rangle \} \\ & \textbf{else} \\ & (2)(\exists m_{t'} = \langle \overline{x}, assert, \langle a, \gamma, v, + \rangle \rangle (1 < t' \leq t) \\ & \quad \textbf{and} \\ & \quad (\textbf{ if } \exists m_{t''} = \langle x, agree, a \rangle) \\ & \quad \textbf{then } \exists A, m_{t'''} = \langle x, assert, A \rangle \\ & \qquad (t'' < t''' \leq t))) \} \end{aligned}$$

The protocol states that it is permissible to assert an argument as long as that argument has not previously been asserted in the dialogue. An agent can agree to an action

$$\mathsf{Strategy}_x(D^t) = \begin{cases} \mathsf{Pick}(S_x^{\mathsf{ag}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) \neq \emptyset \\ \mathsf{Pick}(S_x^{\mathsf{prop}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) = \emptyset \text{ and } S_x^{\mathsf{prop}}(D^t) \neq \emptyset \\ \mathsf{Pick}(S_x^{\mathsf{att}})(D^t) & \text{iff } S_x^{\mathsf{ag}}(D^t) = S_x^{\mathsf{prop}}(D^t) = \emptyset \text{ and } S_x^{\mathsf{att}}(D^t) \neq \emptyset \\ \langle x, \mathsf{close}, \mathsf{Topic}(D^t)\rangle & \text{iff } S_x^{\mathsf{ag}}(D^t) = S_x^{\mathsf{prop}}(D^t) = S_x^{\mathsf{att}}(D^t) = \emptyset \end{cases}$$

where the choices for the moves are given by the following subsidiary functions ($x' \in \{x, \overline{x}\}$, $\mathsf{Topic}(D^t) = \gamma$):

$$\begin{aligned} S_x^{\mathsf{ag}}(D^t) &= \{\langle x, \mathsf{agree}, a\rangle \in P_x^{\mathsf{ag}}(D^t) \mid a \in \mathsf{AgActs}(x, D^t)\} \\ S_x^{\mathsf{prop}}(D^t) &= \{\langle x, \mathsf{assert}, A\rangle \in P_x^{\mathsf{ass}}(D^t) \mid A \in Args_\gamma^x, \mathsf{Act}(A) = a, \mathsf{Sign}(A) = + \text{ and} \\ &\qquad a \in \mathsf{AgActs}(x, D^t)\} \\ S_x^{\mathsf{att}}(D^t) &= \{\langle x, \mathsf{assert}, A\rangle \in P_x^{\mathsf{ass}}(D^t) \mid A \in Args_\gamma^x, \mathsf{Act}(A) = a, \mathsf{Sign}(A) = -, \\ &\qquad a \notin \mathsf{AgActs}(x, D^t) \text{ and } \exists m_{t'} = \langle x', \mathsf{assert}, A'\rangle \\ &\qquad (1 \leq t' \leq t) \text{ s.t. } \mathsf{Act}(A') = a \text{ and } \mathsf{Sign}(A') = +\} \end{aligned}$$

Fig. 1. The **strategy** function uniquely selects a move according to the following preference ordering (starting with the most preferred): an agree move (ag), a proposing assert move (prop), an attacking assert move (att), a close move (close).

that has been agreed to by the other agent in the preceding move (condition 1 of P_x^{ag}); otherwise an agent x can agree to an action that has been proposed by the other participant (condition 2 of P_x^{ag}) as long as if x has previously agreed to that action, then x has since then asserted some new argument. This is because we want to avoid the situation where an agent keeps repeatedly agreeing to an action that the other agent will not agree to: if an agent makes a move agreeing to an action and the other agent does not wish to also agree to that action, then the first agent must introduce some new argument that may convince the second agent to agree before being able to repeat its agree move. Agents may always make a close move. Note, it is straightforward to check conformance with the protocol as it only refers to public elements of the dialogue.

We now define a *basic deliberation strategy*. It takes the dialogue D^t and returns exactly one of the permissible moves. Note, this strategy makes use of a function $\mathsf{Pick} : \wp(\mathcal{M}) \mapsto \mathcal{M}$. We do not define Pick here but leave it as a parameter of our strategy (in its simplest form Pick may return an arbitrary move from the input set); hence our system could generate more than one dialogue depending on the definition of the Pick function. In future work, we plan to design particular Pick functions; for example, taking into account an agent's perception of the other participant (more in section 5).

Definition 11: *The* **basic strategy** *for an agent x is a function* $\mathsf{Strategy}_x : \mathcal{D} \mapsto \mathcal{M}$ *given in Figure 1.*

A *well-formed deliberation dialogue* is a dialogue that has been generated by two agents each following the basic strategy.

Definition 12: *A* **well-formed deliberation dialogue** *is a dialogue D^t s.t. $\forall t'$ ($1 \leq t' \leq t$), $\mathsf{Sender}(m^{t'}) = x$ iff $\mathsf{Strategy}_x(D^{t'-1}) = m_{t'}$*

We now present a simple example. There are two participating agents ($\{1, 2\}$) who have the joint goal to go out for dinner together (din). $Ac^1 \cup Ac^2 = \{it, ch\}$ (it: go to an Italian restaurant; ch: go to a Chinese restaurant) and $Av^1 \cup Av^2 = \{d, e1, e2, c\}$ (d: distance to travel; $e1$: agent 1's enjoyment; $e2$: agent 2's enjoyment; c: cost). The agents' audiences are as follows.

$$d \succ_1 e1 \succ_1 c \succ_1 e2$$
$$c \succ_2 e2 \succ_2 e1 \succ_2 d$$

Agent 1 starts the dialogue.

$$m_1 = \langle 1, open, din \rangle$$

The agents' dialogue iVAFs at this opening stage in the dialogue can be seen in Figs. 2 and 3, where the nodes represent arguments and are labelled with the action that they are for (or the negation of the action that they are against) and the value that they are motivated by. The arcs represent the attack relation between arguments, and a double circle round a node means that the argument it represents is acceptable to that agent.

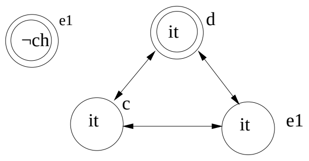


Fig. 2. Agent 1's dialogue iVAF at t = 1, *dVAF*$(1, D^1)$.

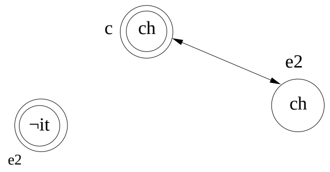


Fig. 3. Agent 2's dialogue iVAF at t = 1, *dVAF*$(2, D^1)$.

At this point in the dialogue, there is only one argument *for* an action that is acceptable to 2 ($\langle ch, din, c, + \rangle$), hence ch is the only action that is agreeable to 2. 2 must therefore assert an argument that it can construct for going to the Chinese restaurant. There are two such arguments that the Pick function could select ($\langle ch, din, c, + \rangle$, $\langle ch, din, e2, + \rangle$). Let us assume that $\langle ch, din, c, + \rangle$ is selected.

$$m_2 = \langle 2, assert, \langle ch, din, c, + \rangle \rangle$$

This new argument is added to 1's dialogue iVAF, to give *dVAF*$(1, D^2)$ (Fig. 4).

Although agent 2 has proposed going to the Chinese restaurant, this action is not agreeable to agent 1 at this point in the dialogue (as there is no argument for this action that is acceptable in Fig. 4). There is, however, an argument for the action it ($\langle it, din, d, + \rangle$) that is acceptable in 1's dialogue iVAF (Fig. 4), and so going to the Italian restaurant is agreeable to 1. Hence, 1 must make an assert move proposing an argument for the action it, and there are three such arguments that the Pick function can select from ($\langle it, din, d, + \rangle$, $\langle it, din, c, + \rangle$, $\langle it, din, e1, + \rangle$). Let us assume that $\langle it, din, c, + \rangle$ is selected.

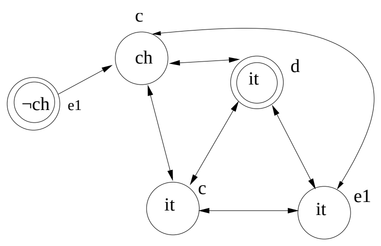


Fig. 4. Agent 1's dialogue iVAF at t = 2, *dVAF*$(1, D^2)$.

$$m_3 = \langle 1, assert, \langle it, din, c, + \rangle \rangle$$

This new argument is added to 2's dialogue iVAF, to give *dVAF*$(2, D^3)$ (Fig. 5).

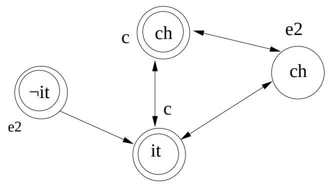


Fig. 5. Agent 2's dialogue iVAF at t = 3, *dVAF*$(2, D^3)$.

Going to the Italian restaurant is now agreeable to agent 2 since the new argument introduced promotes the value ranked most highly for agent 2, i.e. cost, and so this argument is acceptable. So, 2 agrees to this action.

$$m_4 = \langle 2, agree, it \rangle$$

Going to the Italian restaurant is also agreeable to agent 1 (as the argument $\langle it, din, d, + \rangle$ is acceptable in its dialogue iVAF, which is still the same as that shown in Fig. 4 as 2 has not asserted any new arguments), hence 1 also agrees to this action.

$$m_5 = \langle 1, agree, it \rangle$$

Note that the dialogue has terminated successfully and the agents are each happy to agree to go to the Italian restaurant; however, this action is agreeable to each agent for a different reason. Agent 1 is happy to go to the Italian restaurant as it promotes the value of distance to travel (the Italian restaurant is close by), whereas agent 2 is happy to go to the Italian restaurant as it will promote the value of cost (as it is a cheap restaurant). The agents need not be aware of one another's audience in order to reach an agreement.

It is worth mentioning that, as we have left the Pick function unspecified, our strategy could have generated a longer dialogue if, for example, agent 1 had instead chosen to assert the argument $\langle it, din, d, + \rangle$ at the move m_3. This illustrates how an agent's perception of the other participant may be useful: in the previous example agent 1 may make the assumption that, as agent 2 has previously asserted an argument that promotes cost, cost is something that agent 2 values; or an agent may use its perception of another agent's personality to guide argument selection [11].

Another point to note concerns the arguments generated by CQ10. Such arguments do not dispute that the action should be performed, but do dispute the reasons as to why, and so they are modelled as attacks despite being for the same action. Pinpointing this distinction here is important for two main reasons. Firstly, an advantage of the argumentation approach is that agents make explicit the reasons as to why they agree and disagree about the acceptability of arguments, and the acceptability may well turn on such reasons. Where there are two arguments proposed for the same action but each is based upon different values, an agent may only accept the argument based on one of the values. Hence such arguments are seen to be in conflict. Secondly, by participating in dialogues agents reveal what their value orderings are, as pointed out in [12]. If an agent will accept an argument for action based upon one particular value but not another, then this is potentially useful information for future dialogue interactions; if agreement is not reached about a particular action proposal, then dialogue participants will know the values an opposing agent cares about and this can guide the selection of further actions to propose, as we discuss later on in section 5.

A final related issue to note is that of accrual of arguments. If there are multiple arguments for an action and the values promoted are acceptable to the agents then some form of accrual might seem desirable. However, the complex issue of how best to accrue such arguments has not been fully resolved and this is not the focus here.

4 Properties

Certainly (assuming the cooperative agents do not abandon the dialogue for some reason), all dialogues generated by our system terminate. This is clear as we assume that the sets of actions and values available to an agent are finite, hence the set of arguments that an agent can construct is also finite. As the protocol does not allow the agents to keep asserting the same argument, or to keep agreeing to the same action unless a new argument has been asserted, either the dialogue will terminate successfully else the agents will run out of legal assert and agree moves and so each will make a close move.

Proposition 1: *If D^t is a well-formed deliberation dialogue, then $\exists t'$ ($t \leq t'$) s.t. $D^{t'}$ is a well-formed deliberation dialogue that terminates at t' and $D^{t'}$ extends D^t.*

It is also the clear from the definition of the strategy (which only allows an action to be agreed to if that action is agreeable to the agent) that if the dialogue terminates with a successful outcome of action a, then a is agreeable to both agents.

Proposition 2: *If D^t is a well-formed deliberation dialogue that terminates successfully at t with outcome a, then $a \in AgActs(x, D^t)$ and $a \in AgActs(\overline{x}, D^t)$.*

Similarly, we can show that if there is an action that is agreeable to both agents when the dialogue terminates, then the dialogue will terminate successfully. In order to show this, however, we need a subsidiary lemma that states: if an agent makes a close move, then any arguments that it can construct that are for actions that it finds agreeable must have been asserted by one of the agents during the dialogue. This follows from the definition of the strategy, which only allows agents to make a close move once they have exhausted all possible assert moves.

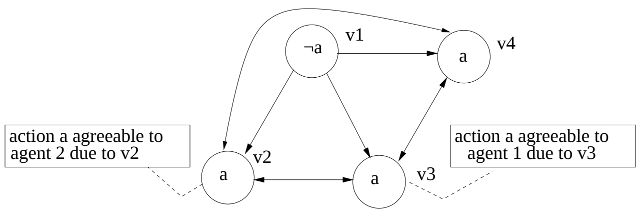

¬a
v1
a
v4
action a agreeable to agent 2 due to v2
a
v2
a
v3
action a agreeable to agent 1 due to v3

Fig. 6. The joint iVAF

Lemma 1: *Let D^t be a well-formed deliberation dialogue with* $\mathsf{Topic}(D^t) = \gamma$*, s.t.* $m_t = \langle x, close, \gamma\rangle$ *and* $\text{dVAF}(x, D^t) = \langle \mathcal{X}, \mathcal{A}\rangle$*. If* $A = \langle a, \gamma, v, +\rangle \in \mathcal{X}$ *and* $a \in AgActs(x, D^t)$*, then* $\exists m_{t'} = \langle x', assert, A, \rangle$ *(*$1 < t' \leq t$*,* $x' \in \{x, \overline{x}\}$*).*

Now we show that if there is an action that is agreeable to both agents when the dialogue terminates, then the dialogue will have a successful outcome.

Proposition 3: *Let D^t be a well-formed deliberation dialogue that terminates at t. If* $a \in AgActs(x, D^t)$ *and* $a \in AgActs(\overline{x}, D^t)$*, then D^t terminates successfully.*

Proof: *Assume that D^t terminates unsuccessfully at t and that* $\mathsf{Sender}(m_t) = \overline{x}$*. From Lemma 1, there is at least one argument A for a that has been asserted by one of the agents. There are two cases. Case 1: x asserted A. Case 2: $\overline{x}$ asserted A.*

Case 1: x asserted A. Hence (from the protocol) it would have been legal for $\overline{x}$ to make the move $m_t = \langle \overline{x}, agree, a\rangle$ *(in which case x would have had to replied with an agree, giving successful termination), unless $\overline{x}$ had previously made a move* $m_{t'} = \langle \overline{x}, agree, a\rangle$ *but had not made a move* $m_{t''} = \langle \overline{x}, assert, A\rangle$ *with* $t' < t'' < t$*. However, if this were the case, then we would have* $AgActs(x, D^{t'}) = AgActs(x, D^t)$ *(because no new arguments have been put forward by $\overline{x}$ to change x's dialogue iVAF), hence x would have had to respond to the move $m_{t'}$ with an agree, terminating the dialogue successfully. Hence contradiction.*

Case 2: works equivalently to case 1. Hence, D^t terminates successfully. □

We have shown then: all dialogues terminate; if a dialogue terminates successfully, then the outcome will be agreeable to both participants; if a dialogue terminates and there is some action that is agreeable to both agents, then the dialogue will have a successful outcome.

It would be desirable to show that if there is some action that is agreeable in the **joint iVAF**, which is the iVAF that can be constructed from the union of the agents' arguments (i.e. the iVAF $\langle \mathcal{X}, \mathcal{A}\rangle$, where $\mathcal{X} = Args^x_\gamma \cup Args^{\overline{x}}_\gamma$ and γ is the topic of the dialogue), then the dialogue will terminate successfully. However, there are some cases where there is an action that is agreeable in the joint iVAF to each of the participants and yet still they may not reach an agreement. Consider the following example in which there is an action a that is agreeable to both the agents given the joint iVAF (see Fig.6) and yet the dialogue generated here terminates unsuccessfully.

The participants ($\{1, 2\}$) have the following audiences.

$$v3 \succ_1 v1 \succ_1 v4 \succ_1 v2$$
$$v2 \succ_2 v1 \succ_2 v4 \succ_2 v3$$

Agent 1 starts the dialogue.

$$m_1 = \langle 1, open, p\rangle$$

The agents' dialogue iVAFs at this stage in the dialogue can be seen in Figs. 7 and 8.

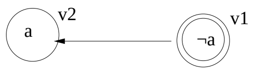


Fig. 7. Agent 1's dialogue iVAF at t = 1, *dVAF*$(1, D^1)$.

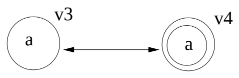


Fig. 8. Agent 2's dialogue iVAF at t = 1, *dVAF*$(2, D^1)$.

At this point in the dialogue, there is one action that is agreeable to agent 2 (a, as there is an argument *for* a that is acceptable in Fig. 8); hence (following the basic dialogue strategy), agent 2 must assert one of the arguments that it can construct for a (either $\langle a, p, v3, +\rangle$ or $\langle a, p, v4, +\rangle$). Recall, we have not specified the Pick function that has to choose between these two possible proposing assert moves. Let us assume that the Pick function makes an arbitrary choice to assert $\langle a, p, v4, +\rangle$.

$$m_2 = \langle 2, assert, \langle a, p, v4, +\rangle\rangle$$

This new argument is added to agent 1's dialogue iVAF, to give *dVAF*$(1, D^2)$ (Fig. 9).

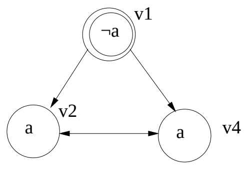


Fig. 9. Agent 1's dialogue iVAF at t = 2, *dVAF*$(1, D^2)$.

From Fig. 9, we see that the only argument that is now acceptable to agent 1 is the argument *against* a ($\langle a, p, v1, -\rangle$), hence there are no actions that are agreeable to agent 1. Thus agent 1 must make an attacking assert move.

$$m_3 = \langle 1, assert, \langle a, p, v1, -\rangle\rangle$$

This new argument is added to agent 2's dialogue iVAF, to give *dVAF*$(2, D^3)$ (Fig. 10).

We see from Fig. 10 that the only argument that is now acceptable to agent 2 is the argument *against* a that 1 has just asserted ($\langle a, p, v1, -\rangle$); hence, a is now no longer an agreeable action for agent 2. As there are now no actions that are agreeable to agent

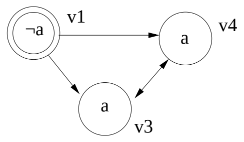


Fig. 10. Agent 2's dialogue iVAF at t = 3, *dVAF*$(2, D^3)$.

2, it cannot make any proposing assert moves. It also cannot make any attacking assert moves, as the only argument that it can construct against an action has already been asserted by agent 1. Hence, agent 2 makes a close move.

$$m_4 = \langle 2, close, p \rangle$$

Thus, the dialogue iVAF for 1 is still the same as that which appears in Fig. 9. As there are no actions that are agreeable to agent 1, it cannot make any proposing assert moves. It cannot make any attacking assert moves, as the only argument that it can construct against an action has already been asserted. Hence, agent 1 also makes a close move.

$$m_5 = \langle 1, close, p \rangle$$

The dialogue has thus terminated unsuccessfully and the agents have not managed to reach an agreement as to how to achieve the goal p. However, we can see that if the Pick function instead selected the argument $\langle a, p, v3, + \rangle$ for agent 2 to assert for the move m_2, then the resulting dialogue would have led to a successful outcome.

This example then illustrates a particular problem: the arguments exist that will enable the agents to reach an agreement (we can see this in the joint iVAF, Fig. 6, in which each agent finds a agreeable) and yet the particular arguments selected by the Pick function may not allow agreement to be reached. The choice of moves made in a deliberation dialogue affects the dialogue outcome; hence, strategic manoeuvring within the dialogue is possible in order to try to influence the dialogue outcome.

This evaluation helps us to understand the complex issues and difficulties involved in allowing agents with different preferences to agree how to act. We discuss possible responses to some of these difficulties in the next section.

5 Proposed extensions

One way in which we could aim to avoid the problem illustrated in the previous example is by allowing agents to develop a model of which values they believe are important to the other participant. This model can then be used by the Pick function in order to select arguments that are more likely to lead to agreement (i.e. those that the agent believes promote or demote values that are highly preferred by the other participant). Consider the above example, if agent 2 believed that value $v3$ was more preferred to agent 1 than value $v4$, then 2 would have instead asserted $\langle a, p, v3, + \rangle$ for the move m_2, which would have led to a successful outcome.

Therefore, the first extension that we plan to investigate is to design a particular Pick function that takes into account what values the agent believes are important to the other participant. We also plan to develop a mechanism which allows the agent to build up its model of the other participant, based on the other participant's dialogue behaviour; for example, if an agent x asserts an argument for an action a because it promotes a particular value v, and the other participant $\overline{x}$ does not then agree to a, agent x may have reason to believe that $\overline{x}$ does not highly rank the value v.

Another problem that may be faced with our dialogue system is when it is not possible for the agents to come to an agreement no matter which arguments they choose to assert. The simplest example of this is when each agent can only construct one argument to achieve the topic p: agent 1 can construct $\langle a1, p, v1, + \rangle$; agent 2 can construct $\langle a2, p, v2, + \rangle$. Now if agent 1's audience is such that it prefers $v1$ to $v2$ and agent 2's audience is such that it prefers $v2$ to $v1$, then the agents will not be able to reach an agreement with the dialogue system that we have proposed here; this is despite the fact that both agents do share the goal of coming to some agreement on how to act to achieve p. The agents in this case have reached an impasse, where there is no way of finding an action that is agreeable to both agents given their individual preferences over the values.

The second extension that we propose to investigate aims to overcome such an impasse when agreement is nevertheless necessary. We plan to define a new type of dialogue (which could be embedded within the deliberation dialogue we have defined here) that allows the agents to discuss their preferences over the values and to suggest and agree to compromises that allow them to arrive at an agreement in the deliberation dialogue. For example, if agent 1's audience is $v1 \succ_1 v2 \succ_1 v3$ and agent 2's audience is $v3 \succ_2 v2 \succ_2 v1$, then they may both be willing to switch their first and second most preferred values if this were to lead to an agreement (i.e. giving $v2 \succ_1 v1 \succ_1 v3$ and $v2 \succ_2 v3 \succ_2 v1$).

We would also like to extend our system to deal with the situation in which the other stages of practical reasoning (problem formulation and epistemic reasoning) may be flawed. In [5], an approach to dealing with epistemic reasoning was presented, that allowed an embedded inquiry subdialogue with which agents could jointly reason epistemically about the state of the world. Thus, the third extension that we propose is to develop a new type of dialogue that will allow agents to jointly reason about the elements of a VATS in order to consider possible flaws in the problem formulation stage.

6 Related Work

There is existing work in the literature on argumentation that bears some relation to what we have presented here, though the aims and contributions of these approaches are markedly different.

Our proposal follows the approach in [5, 13] but the types of moves are different, and the protocol and strategy functions are substantially altered from those presented in either [5] or [13]. This alteration is necessary as neither of [5, 13] allow agents to participate in deliberation dialogues. In [13], a dialogue system is presented for epistemic inquiry dialogues; it allows agents to jointly construct argument graphs (where the arguments refer only to beliefs) and to use a shared defeat relation to determine the acceptability of particular arguments.

The proposal of [5] is closer to that presented here, as both are concerned with how to act. However, the dialogue system in [5] does not allow deliberation dialogues as the outcome of any dialogue that it generates is predetermined by the union of the participating agents' knowledge. Rather, the dialogues of [5] are better categorised as a joint inference; they ensure that the agents assert all arguments that may be relevant to the question of how to act, after which a universal value ordering is applied to determine the outcome. As a shared universal value ordering is used in [5], there is an objective view of the "best" outcome (being that which you would get if you pooled the agents' knowledge and applied the shared ordering); this is in contrast to the dialogue system we present here, where the "best" outcome is subjective and depends on the point of view of a particular agent. As the agents presented here each have their own distinct audience, they must come to an explicit agreement about how to act (hence the introduction of an agree move) despite the fact that their internal views of argument acceptability may conflict. Also, here we define the attack relation (in the iVAF), which takes account of the relevant CQs, whilst in [5] the attack relation is only informally discussed.

Deliberation dialogues have only been considered in detail by the authors of [2, 3]. Unlike in our work, in [2] the evaluation of arguments is not done in terms of argumentation frameworks, and strategies for reaching agreement are not considered; and in [3] the focus is on goal selection and planning.

In [12] issues concerning audiences in argumentation frameworks are addressed where the concern is to find particular audiences (if they exist) for which some arguments are acceptable and others are not. Also considered is how preferences over values emerge through a dialogue; this is demonstrated by considering how two agents can make moves within a dialogue where both are dealing with the same joint graph. However, the graph can be seen as a static structure within which agents are playing moves, i.e. putting forward acceptable arguments, rather than constructing a graph that is not complete at the outset, as in the approach we have presented.

There is also some work that considers how Dungian argumentation frameworks associated with individual agents can be merged together [14]. The merging is done not through taking the union of the individual frameworks, but through the application of criteria that determine when arguments and attacks between them can be merged into a larger graph. The main goal of the work is to characterise the sets of arguments acceptable by the whole group of agents using notions of joint acceptability, which include voting methods. In our work we are not interested in merging individual agent's graphs *per se*; rather, an agent develops its own individual graph and uses this to determine if it finds an action agreeable. In [14] no dialogical interactions are considered, and it is also explicitly noted that consideration has not been given to how the merging approach can be applied to value-based argument systems.

Prakken [15] considers how agents can come to a public agreement despite their internal views of argument acceptability conflicting, allowing them to make explicit attack and surrender moves. However, Prakken does not explicitly consider value-based arguments, nor does he discuss particular strategies.

Strategic argumentation has been considered in other work. For example, in [16] a dialogue game for persuasion is presented that is based upon one originally proposed in [1] but makes use of Dungian argumentation frameworks. Scope is provided for three strategic considerations which concern: revealing inconsistencies between an opponent's commitments and his beliefs; exploiting the opponent's reasoning so as to create such inconsistencies; and revealing blunders to be avoided in expanding the opponent's knowledge base. These strategies all concern reasoning about an opponent's beliefs, as opposed to reasoning about action proposals with subjective preferences, as done in our work, and the game in [16] is of an adversarial nature, whereas our setting is more co-operative.

One account that does consider strategies when reasoning with value-based arguments is given in [7], where the objective is to create obligations on the opponent to accept some argument based on his previously expressed preferences. The starting point for such an interaction is a fixed joint VAF, shared by the dialogue participants. In our approach the information is not centralised in this manner, the argument graphs are built up as the dialogue proceeds, we do not assume perfect knowledge of the other agent's graph and preferences, and our dialogues have a more co-operative nature.

A related new area that is starting to receive attention is the application of game theory to argumentation (e.g. [17]). This work has investigated situations under which rational agents will not have any incentive to lie about or hide arguments; although this is concerned mainly with protocol design, it appears likely that such work will have implications for strategy design.

A few works do explicitly consider the selection of dialogue targets, that is the selection of a particular previous move to respond to. In [15] a move is defined as relevant if its target would (if attacked) cause the status of the original move to change; properties of dialogues are considered where agents are restricted to making relevant moves. In [18] this is built on to consider other classes of move relevance and the space that agents then have for strategic manoeuvring. However, these works only investigate properties of the dialogue protocols; they do not consider particular strategies for such dialogues as we do here.

7 Concluding Remarks

We have presented a dialogue system for joint deliberation where the agents involved in the decision making may each have different preferences yet all want an agreement to be reached. We defined how arguments and critiques are generated and evaluated, and how this is done within the context of a dialogue. A key aspect concerns how agents' individual reasoning fits within a more global context, without the requirement to completely merge all knowledge. We presented some properties of our system that show when agreement can be guaranteed, and have explored why an agreement may not be reached. Identifying such situations is crucial for conflict resolution and we have discussed how particular steps can be taken to try to reach agreement when this occurs. In future work we intend to give a fuller account of such resolution steps whereby reasoning about other agents' preferences is central.

Ours is the first work to provide a dialogue strategy that allows agents with different preferences to come to an agreement as to how to act. The system allows strategical manoeuvring in order to influence the dialogue outcome, thus laying the important foundations needed to understand how strategy design affects dialogue outcome when the preferences involved are subjective.

—

References

1. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press, Albany, NY, USA (1995)
2. McBurney, P., Hitchcock, D., Parsons, S.: The eightfold way of deliberation dialogue. International Journal of Intelligent Systems **22**(1) (2007) 95–132
3. Tang, Y., Parsons, S.: Argumentation-based dialogues for deliberation. In: 4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems. (2005) 552–559
4. Dignum, F., Vreeswijk, G.: Towards a testbed for multi-party dialogues. In: AAMAS Int. Workshop on Agent Communication Languages and Conversation Policies. (2003) 63–71
5. Black, E., Atkinson, K.: Dialogues that account for different perspectives in collaborative argumentation. In: 8th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems. (2009) 867–874
6. Walton, D.N.: Argumentation Schemes for Presumptive Reasoning. Lawrence Erlbaum Associates, Mahwah, NJ, USA (1996)
7. Bench-Capon, T.J.M.: Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. Informal Logic **22**(3) (2002) 231–245
8. Atkinson, K., Bench-Capon, T.J.M.: Practical reasoning as presumptive argumentation using action based alternating transition systems. Artificial Intelligence **171**(10–15) (2007) 855–874
9. Wooldridge, M., van der Hoek, W.: On obligations and normative ability: Towards a logical analysis of the social contract. J. of Applied Logic **3** (2005) 396–420
10. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77** (1995) 321–357
11. van der Weide, T., Dignum, F., J.-J, Meyer, Prakken, H., Vreeswijk, G.: Personality-based practical reasoning. In: 5th Int. Workshop on Argumentation in Multi-Agent Systems. (2008) 3–18
12. Bench-Capon, T.J.M., Doutre, S., Dunne, P.E.: Audiences in argumentation frameworks. Artificial Intelligence **171**(1) (2007) 42–71
13. Black, E., Hunter, A.: An inquiry dialogue system. Autonomous Agents and Multi-Agent Systems **19**(2) (2009) 173–209
14. Coste-Marquis, S., Devred, C., Konieczny, S., Lagasquie-Schiex, M.C., Marquis, P.: On the merging of Dung's argumentation systems. Artificial Intelligence **171**(10–15) (2007) 730–753
15. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. J. of Logic and Computation **15** (2005) 1009–1040
16. Devereux, J., Reed, C.: Strategic argumentation in rigorous persuasion dialogue. In: 6th Int. Workshop on Argumentation in Multi-Agent Systems. (2009) 37–54
17. Rahwan, I., Larson, K.: Mechanism design for abstract argumentation. In: 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems. (2008) 1031–1038
18. Parsons, S., McBurney, P., Sklar, E., Wooldridge, M.: On the relevance of utterances in formal inter-agent dialogues. In: 6th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems. (2007) 1002–1009

An Argument-Based Multi-Agent System for Information Integration

Marcela Capobianco and Guillermo R. Simari

Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
Universidad Nacional del Sur – Av. Alem 1253, (8000) Bahía Blanca ARGENTINA
EMAIL: {*mc,grs*}*@cs.uns.edu.ar*

Abstract. In this paper we address the problem of obtaining a consolidated view of the knowledge that a community of information agents possesses in the form of private, possibly large, databases. Each agent in the community has independent sources of information and each database could contain information that is potentially inconsistent and incomplete, both by itself and/or in conjunction with some of the others. These characteristics make the consolidation difficult by traditional means. The idea of obtaining a single view is to provide a way of querying the resulting knowledge in a skeptical manner, *i.e.*, receiving one answer that reflects the perception of the information community.
Agents using the proposed system will be able to access multiple sources of knowledge represented in the form of deductive databases as if they were accessing a single one. One application of this schema is a novel architecture for decision-support systems (DSS) that will combine database technologies, specifically federated databases, which we will cast as information agents, with an argumentation-based framework.

Categories and Subjects Descriptors: I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—Representation languages, Representations (procedural and rule-based); H.1.0 [Models and Principles]: Systems and Information Theory—General systems theory.
General Terms: Algorithms, Design, Performance.
Keywords: Argumentation, Knowledge representation, Design languages for agent systems.

1 Introduction

Information systems, and the capability to obtain answers from them, play a key role in our society. In particular, data intensive applications are in constant demand and there is need for computing environments with much more intelligent capabilities than those present in today's Data-base Management Systems (DBMS). The expected requirements for these systems change every day: they constantly become more complex and more advanced features are demanded from them.

In this paper we address the problem of obtaining a consolidated view of the knowledge that a community of information agents possess in the form of private, possibly large, databases. Each agent in the community has independent sources of information and each database could contain information that is potentially inconsistent and incomplete, both by itself and/or in conjunction with some of the others. These characteristics make the consolidation difficult by traditional means. The idea of obtaining a single view is to provide a way of querying the resulting knowledge in a skeptical manner, *i.e.*, receiving one answer that reflects the perception of the information community.

Agents using the proposed system will be able to access multiple sources of knowledge represented in the form of deductive databases as if they were accessing a single one. One application of this schema is a novel architecture for decision-support systems (DSS) that will combine database technologies, specifically federated databases, which we will cast as information agents, with an argumentation-based framework.

Recently, there has been much progress in developing efficient techniques to store and retrieve data, and many satisfactory solutions have been found for the associated problems. However it remains an open problem how to understand and interpret large amounts of information. To do this we need specific formalisms that can perform complicated inferences, obtain the appropriate conclusions, and justify their results. We claim that these formalisms should also be able to access seamlessly databases distributed over a network.

In the field of deductive databases there has been a continued effort to produce an answer to this problem. Deductive databases store explicit and implicit information; explicit information is stored in the manner of a relational database and implicit information is recorded in the form of rules that enable inferences based on the stored data. These systems combine techniques and tools from relational databases with rule based formalisms. Hence, they are capable of handling large amounts of information and perform some sort of reasoning based on it. Nevertheless, these systems have certain limitations and shortcomings for knowledge representation and modeling commonsense reasoning, especially for managing incomplete and potentially contradictory information, as argued by several authors [18, 24, 17].

Argumentation frameworks [10, 20, 15] are an excellent starting point for building intelligent systems with interesting reasoning abilities. Research in argumentation has provided important results while striving to obtain tools for commonsense reasoning, and this has sprung a new set of argument-based applications in diverse areas where knowledge representation issues play a major role [11, 5, 8].

We believe that deductive databases can be combined with argumentation formalisms to obtain interactive systems able to reason with large databases, even in the presence of incomplete and potentially contradictory information. This can be a significant advantage with respect to systems based on logic programming, such as datalog, that cannot deal with contradictory information.[1] In particular, this could be useful in contexts where information is obtained from multiple databases, and these databases may be contradictory among themselves.

The multi-agent system introduced here virtually integrates different databases into a common view; in that manner users of this system can query multiple databases as if they were a single one. This schema can be applied to obtain a novel system architecture for decision-support systems (DSS) that combines database technologies, specifically federated databases [19], with an argumentation based framework.

In our proposal we consider that information is obtained from a number of different heterogeneous database systems, each represented by a particular agent. The reasoning mechanisms, based on an argumentative engine, use this information to construct a consolidated global view of the database. This task is performed by the reasoning agent, that is based on a set of rules expressed in a particular argumentation framework. This agent can deal with incomplete and contradictory information and can also be personalized for any particular DSS in a relatively simple way.

We have also considered that one of the design objectives of interactive systems is that they can respond in a timely manner to users' queries. So far the main objection to the use of argumentation in interactive systems is their computational complexity. In previous work [6] we have addressed the issue of optimizing argumentation systems, where the optimization technique consisted in using a precompiled knowledge component as a tool to allow significant speed-ups in the inference process. We also apply this technique in our reasoning agent.

To understand the advantages of the proposed reasoning mechanism used in our multiagent system, consider a set of databases used by the employers responsible of drug administration, sales, and delivery in a given hospital. These databases contains information regarding drugs, patients, known allergies, and addictions. Suppose a deductive database system in the style of datalog is used to query this information to derive certain relations. In this setting, there is a rule establishing that a drug should be given to a patient if the patient has a prescription for this drug signed by a physician. There could also be a rule saying that the drug should not be sold if the prescription is signed by the patient. In this case, if Dr. Gregory House enters the clinic with a prescription signed by himself to get Vicodin, the employers could query the system to see if the drug should or should not be sold. If a traditional deductive database is used, in the style of datalog or another logic programming based system, this would give raise to a contradiction and the system would not be able to recommend a course of action. Using our system, an argument can be built backing that the medication should be sold, given that there is a prescription signed by a doctor that warrants it. However, an argument for not selling the drug could also be built considering that the doctor and the patient are the same person. Our

[1] Some extensions of datalog handle negation using CWA (see [9]), but these approaches do not allow negation in the head of the rules in the style of extended logic programming.

argument-based framework can then compare both arguments, decide that the second argument is preferred, and answer the query saying that the drug should no be sold. In addition, it can also explain the reasons that back its answer. Note that this kind of behavior cannot be obtained in Datalog-like systems.

The rest of this article is organized as follows. Section 2 sums up related work, Section 3 contains our proposal for the system architecture, and section 4 formally describes the reasoning module, a key component of this architecture. Section 5 presents a key optimization for the argumentation-based reasoning process, and Section 6 shows a realistic example of the system's mechanics. Finally, Section 7 states the conclusions.

2 Integrating DBMS and Reasoning Systems

Previous work on the integration of databases and reasoning systems has almost been restricted to coupling Prolog interpreters and relational databases. These approaches were motivated in the declarative nature of logic programming languages and the data-handling capabilities of database systems. Several researchers have built intelligent database systems coupling Prolog and a relational DBMS or extending Prolog with database facilities [9]. These works were motivated by the fact that Prolog attracted attention in the 80's for its ability to support rapid development of complex applications. Besides, Prolog is based on Horn Clauses that are close relatives of database query languages and its language is more powerful than SQL [25].

Brodie and Jarke [4] envisioned several years ago that large scale data processing would require more efficient and more intelligent access to databases. He proposed the integration of logic programming and databases to meet future requirements. First, he identified two different approaches for coupling a Prolog interpreter and a Relational DBMS, which are usually called "tight coupling' and "loose coupling". In the tight coupling approach the Prolog interpreter and the Relational DBMS are strongly integrated. For example, the Prolog interpreter can directly use low level functionalities of the DBMS, like relation management in secondary memory, and relation access via indexes [13]. In contrast, in the loose coupling approach, the Relational DBMS is called by the Prolog interpreter at the top level, that acts like a standard user. It sends Relational queries to the DBMS, and the corresponding answers are treated as ground clauses by the interpreter.

Brodie and Jarke also identified four basic architectural frameworks for combining Prolog and a database system:

- Loose coupling of an existing Prolog implementation to an existing relational database system;
- Extending Prolog to include some facilities of the relational database system;
- Extending an existing relational database to include some features of Prolog;
- Tightly integrating logic programming techniques with those of relational database systems.

They recommend the fourth alternative (tight integration), based on the belief that a special purposed language for large scale knowledge base systems would best address issues regarding performance, knowledge representation and software engineering. They also put forward a number of issues concerning the best division of tasks between logic programming and a DBMS.

Zaniolo [26] proposed an approach to intelligent databases based on deductive databases. He advocates for elevating database programming to the more abstract level of rules and knowledge base programming to create an environment more supportive of the new wave of database applications. To achieve these goals the *LDL*/*LDL*+ project was developed. During the project a new logic-based language was designed along with the definition of its formal semantics, new implementation techniques were developed for the efficient support of rule-based logic languages, and it was successfully used in a wide range of application domains. The system supported an open architecture and SQL schema from external databases could be incorporated into the *LDL* program seamlessly.

In the following section we present the system architecture for our proposal. We believe that argumentation can offer a new perspective into the problem of reasoning with large databases, giving more expressive power to the reasoning component, making it able to decide even in the presence of uncertain and/or contradictory information. This addresses a limitation that was present in each of the deductive database systems considered in this section.

3 System Architecture

In this section we present an architectural pattern for our multiagent system that can be applied to design information-based applications where a certain level of intelligence is required. Such applications will be engineered for contexts where: (1) information is uncertain and heterogeneous, (2) handling of great volume of data flows is needed, and (3) data may be incomplete, vague, or contradictory. These applications are also expected to integrate multiple information systems such as databases, knowledge bases, source systems, etc.

Our system architecture is presented in Figure 1. The architecture is modular and is independent of any particular domain or application. We have used a layered architectural style, where every layer provides a series of services to the one above. The first of our layers concerns data and knowledge acquisition. This layer will receive heterogeneous sources of data and will extract and transform this data into the formats required of the particular application. It can work with diverse sources, such as laboratory data, different types of sensors, knowledge bases, etc.

The received data will be formatted to comply with the relational models provided by a group of federated databases that share a common export schema. In our system, each one of the databases is represented by an agent. The common export schema will be the result of a negotiation process among these agents. The union of the views of these databases will generate a key element of our framework, the extensional database that contains the information needed for the

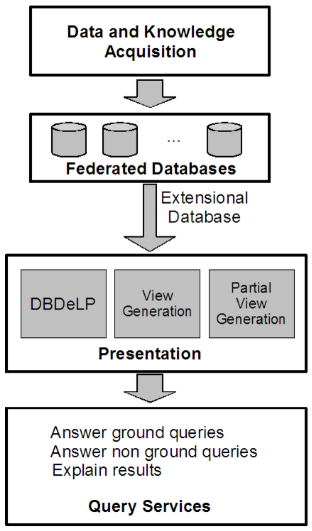


Fig. 1. Proposed architectural pattern

reasoning module. The extensional database also will provide the data elements with a certainty degree that depends of the credibility of the data source from where it was obtained.

We have chosen to use a multi-source perspective for the characterization of data quality [3]. In this case, the quality of data can be evaluated by comparison with the quality of other homologous data (i.e., data from different information sources which represent the same reality but may have contradictory values). The approaches usually adopted to reconcile heterogeneity between values of data are: (1) to prefer the values of the most reliable sources, (2) to mention the source ID for each value, or (3) to store quality meta-data with the data.

For our proposed architecture, we have used the second approach. In multi-source databases, each attribute of a multiple source element has multiple values with the ID of their source and their associated *Quality of Expertise*, which is represented as meta-data associated with each value, such as a given certainty degree. This degree may be obtained weighting the plausibility of the data value, its accuracy, the credibility of its source, and the freshness of the data.

The federated database layer provides the extensional database to the presentation layer. The extensional database can be computed on demand and is not necessarily stored in a physical component. The presentation layer contains the services related with the reasoning process and its optimization. This is the core of our proposal and will be described later on in Sections 4 and 5. The reasoning agent that generated the consolidated view is part of this layer. It contains the set of rules that encode the specific knowledge of the application. These rules will be used by the argumentation-based inference engine. The presentation layer also commands the generation of the extensional database, and the selection if it is going to be done on demand (following a lazy approach) or if it has to be computed completely. It can also generate a partial view of the system according to these rules, resulting in an optimization mechanism. This partial view depends only on the set of rules and must be changed accordingly if changes on the rules are produced. Finally, the query services layer is composed by an interactive agent that receives user queries, provides answers, and can also explain the reasons backing these answers.

4 The DB_DeLP Argumentation Framework

In this section we formally define the argumentation system that is used by the reasoning agent in the Query Services Layer of the proposed system architecture. Here we detail the semantics and proof theory of the framework and we also show some practical examples. A simplified view of our system would describe it as a deductive database whose inference engine is based on a specialization of the DeLP language [16]. This particular framework will be known as *Database Defeasible Logic Programming* (DB_DeLP). Formally, DB_DeLP is a language for knowledge representation and reasoning that uses *defeasible argumentation* to decide between contradictory conclusions through a *dialectical analysis*. DB_DeLP also incorporates uncertainty management, taking elements from Possibilistic Defeasible Logic Programming (P_DeLP) [2, 1], an extension of DeLP in which the elements of the language have the form (ϕ, α), where ϕ is a DeLP clause or fact and α expresses a lower bond for the certainty of ϕ in terms of a necessity measure. Conceptually, our deductive database consists of an extensional database **EDB**, an intensional database **IDB**, and a set of integrity constrains **IC**. In what follows, we formally define these elements.

The language of DB_DeLP follows a logic programming style. Standard logic programming concepts (such as signature, variables, functions, *etc.*) are defined in the usual way. Literals are atoms that may be preceded by the symbol "$\sim$" denoting *strict* negation, as in extended logic programming.

Definition 1. [Literal–Weighted Literal] *Let Σ be a signature, then every atom A of Σ is a positive literal, while every negated atom $\sim A$ is a negative literal. A* literal *of Σ is a positive literal or a negative literal. A* certainty weighted literal*, or simply a weighted literal, is a pair (L, α) where L is a literal and $\alpha \in [0, 1]$ expresses a lower bound for the certainty of L in terms of a necessity measure.*

The extensional database **EDB** is composed by a set of certainty weighted literals, according to the export schema of the federated database that is part of our architecture. Conceptually, it accounts for the union of the views of every particular database that belongs to the federation [19]. When implementing the system, this set of ground literals may not be physically stored in any place, and may simply be obtained on demand when information about a particular literal is needed.

The certainty degree associated with every literal is assigned by the federated database layer that assigns a particular degree to every data source according to its plausibility. The resulting extensional database is not necessarily consistent, in the sense that a literal and its complement w.r.t. strong negation may both be present, with different or the same certainty degrees. In this case, the system decides according to a given criterion which fact will prevail and which one will be removed from the view.

`species(X,Y)`	`age(X,Y)`
(`species(simba,lion)`, 0.6)	(`age(simba,young)`, 0.65)
(`species(mufasa,lion)`, 0.7)	(`age(mufasa,old)`, 0.7)
(`species(grace,lion)`, 0.6)	(`age(grace,adult)`, 0.8)
(`species(grace,leopard)`, 0.4)	(`age(dumbo,baby)`, 0.8)
...	...

Fig. 2. An Extensional Database in DB_DeLP

The intensional part of a DB_DeLP database is formed by a set of *defeasible rules* and *integrity constraints* Defeasible rules provide a way of performing tentative reasoning as in other argumentation formalisms [10, 20, 15].

Definition 2. [Defeasible Rule] *A* defeasible rule *expresses a tentative, weighted, relation between a literal* L_0 *and a finite set of literals* $\{L_1, L_2, \ldots, L_k\}$ *and has the form* $(L_0 \prec L_1, L_2, \ldots, L_k, \alpha)$ *where* $\alpha \in [0, 1]$ *expresses a lower bound for the certainty of the rule in terms of a necessity measure.*

In previously defined argumentation systems, the meaning of defeasible rules $L_0 \prec L_1, L_2, \ldots, L_k$ was understood as *"$L_1, L_2, \ldots, L_k$ provide tentative reasons to believe in L_0"* [23], but these rules did not have an associated certainty degree. In contrast, DB_DeLP adds the certainty degree, that expresses how strong is the connection between the premises and the conclusion. A defeasible rule with a certainty degree 1 will model a strong rule. Figures 2 and 3 show an extensional and an intensional database in our formalism.

```
(feline(X) ⤙ species(X,lion),1)
(climbs_tree(X) ⤙ feline(X),0.65)
(~climbs_tree(X) ⤙ species(X,lion),0.70)
(climbs_tree(X) ⤙ species(X,lion), age(X,young).,0.75)
(~climbs_tree(X) ⤙ sick(X),0.45)
```

Fig. 3. An Intensional Database in DB_DeLP

Note that DB_DeLP programs are *range-restricted*, a common condition for deductive databases: a program is said to be range-restricted if and only if every variable that appears in the head of the clause also appears in its body. This implies that all the facts in the program must be ground (cannot contain variables). These programs can be interpreted more efficiently since full unification is not needed, only matching that is significantly more efficient. Nevertheless, the reason for this decision comes from a semantic standpoint, given that a defeasible reason in which there is no connection between the head and the body has no clear meaning; the range restriction ensures this connection.

Integrity constraints are rules of the form $L \leftarrow L_0, L_1, \ldots, L_n$ where L is a literal, and $L_0, L_1, \ldots, L_k$ is a non-empty finite set of literals, These rules are used to compute the *derived negations* as follows. For the extensional and intensional databases regarding felines, consider that the set of integrity constraints is composed by {`~leopard(X)` ← `lion(X)`, `~lion(X)` ← `leopard(X)`} and the negations { (`~species(grace,lion)`, 0.4), (`~species(grace,leopard)`, 0.6)} are then added to the extensional database. The certainty degree of the added rule is calculated as the minimum of the certainty degree of the literals that are present in the body of the integrity constraint rule used to obtain it. Note that a conflict may arise with information received from other knowledge bases, since we may want to add a literal and its complement may be already present in the extensional database. Then the system will decide according to a given criterion which fact will prevail and which one will be removed from the view. A standard criterion in this case would be using the plausibility of the source, the certainty degree of the literals, or a combination of both. Databases in DB_DeLP, for short called *defeasible databases*, can also include built-in predicates as needed along with their corresponding axioms.

The P_DeLP language [12], which presented the novel idea of mixing argumentation and possibilistic logic, is based on Possibilistic Gödel Logic or PGL [2, 1], which is able to model both uncertainty and fuzziness and allows for a partial matching mechanism between fuzzy propositional variables. In DB_DeLP, for simplicity reasons, we will avoid fuzzy propositions, and hence it will be based on the necessity-valued classical Possibilistic logic [14]. As a consequence, possibilistic models are defined by possibility distributions on the set of classical interpretations, and the proof theory for our formulas, written $\mid\!\sim$, is defined by derivation based on the following instance of the Generalized Modus Ponens rule (GMP): $(L_0 \prec L_1 \wedge \cdots \wedge L_k, \gamma), (L_1, \beta_1), \ldots, (L_k, \beta_k) \vdash (L_0, \min(\gamma, \beta_1, \ldots, \beta_k))$, which is a particular instance of the well-known possibilistic resolution rule, and which provides the *non-fuzzy* fragment of DB_DeLP with a complete calculus for determining the maximum degree of possibilistic entailment for weighted literals. Literals in the extensional database are the base case of the derivation sequence; for every literal Q in **EDB** with a certainty degree α it holds that (Q, α) can be derived from the corresponding program.

A query presented to a DB_DeLP database is a a ground literal Q which must be supported by an *argument*. Deduction in DB_DeLP is argumentation-based, thus a derivation is not enough to endorse a particular fact, and queries must be supported by arguments. In the following definition *instances(**IDB**)* accounts for any set of ground instances of the rules in **IDB**, replacing free variables for ground literals in the usual way.

Definition 3. [Argument]–[Subargument] *Let $\mathcal{DB} = (\boldsymbol{EDB}, \boldsymbol{IDB}, \boldsymbol{IC})$ be a defeasible database, $\mathcal{A} \subseteq$ instances(**IDB**) is an* argument *for a goal Q with necessity degree $\alpha > 0$, denoted as $\langle \mathcal{A}, Q, \alpha \rangle$, iff:* (1) *$\Psi \cup \mathcal{A} \mid\!\sim (Q, \alpha)$,* (2) *$\Psi \cup \mathcal{A}$ is non contradictory, and* (3) *there is no $\mathcal{A}_1 \subset \mathcal{A}$ such that $\Psi \cup \mathcal{A}_1 \mid\!\sim (Q, \beta)$, $\beta > 0$. An argument $\langle \mathcal{A}, Q, \alpha \rangle$ is a* subargument *of $\langle \mathcal{B}, R, \beta \rangle$ iff $\mathcal{A} \subseteq \mathcal{B}$.*

Arguments in DB_DeLP can attack each other; this situation is captured by the notion of *counterargument.* An argument $\langle \mathcal{A}_1, Q_1, \alpha \rangle$ *counter-argues* an argument $\langle \mathcal{A}_2, Q_2, \beta \rangle$ at a literal Q if and only if there is a sub-argument $\langle \mathcal{A}, Q, \gamma \rangle$

of $\langle \mathcal{A}_2, Q_2, \beta \rangle$, (called *disagreement subargument*), such that Q_1 and Q are complementary literals. Defeat among arguments is defined combining the counter-argument relation and a preference criterion "$\preceq$". This criterion is defined on the basis of the necessity measures associated with arguments.

Definition 4. [Preference criterion $\succeq$][12] *Let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ be a counterargument for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is* preferred *over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (denoted $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succeq \langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff $\alpha_1 \geq \alpha_2$. If it is the case that $\alpha_1 > \alpha_2$, then we will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is* strictly preferred *over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \succ \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$. Otherwise, if $\alpha_1 = \alpha_2$ we will say that both arguments are* equi-preferred, *denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \approx \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$.*

Definition 5. [Defeat]][12] *Let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments built from a program $\mathcal{P}$. Then $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$* defeats *$\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (or equivalently $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a* defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff (1) Argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counter-argues argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \alpha \rangle$; and (2) Either it is true that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succ \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a* proper defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, or $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \approx \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$.*

As in most argumentation systems [10, 21], DB_DeLP relies on an exhaustive dialectical analysis which allows to determine if a given argument is *ultimately* undefeated (or *warranted*) w.r.t. a program $\mathcal{P}$. An *argumentation line* starting with an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is a sequence $[\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \ldots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (even-numbered arguments) and an *opponent* (odd-numbered arguments).

Given a program $\mathcal{P}$ and an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, the set of all acceptable argumentation lines starting with $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ (i.e. all possible dialogs rooted in$\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$), formalized as a *dialectical tree* and denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$. Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ can be marked as *undefeated* or *defeated* nodes (U-nodes and D-nodes, resp.). A dialectical tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ will be marked as U-nodes (as they have no defeaters), and every inner node is to be marked as a *D-node* iff it has at least one U-node as a child, and as a *U-node* otherwise. An argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is ultimately accepted as valid (or *warranted*) iff the root of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ is labeled as a *U-node.*

Definition 6. [Warrant][12] *Given a database $\mathcal{DB}$, and a literal Q, Q is warranted w.r.t. $\mathcal{DB}$ iff there exists a warranted argument $\langle \mathcal{A}, Q, \alpha \rangle$ that can be built from $\mathcal{P}$.*

Example 1. Suppose the system has to solve the query *climbs(simba)*. Then argument

$$\mathcal{A}_2 = \{(\textit{climbs(simba)} \prec \textit{feline(simba)}, 0.65), (\textit{feline(simba)} \prec \textit{species(simba,lion)}, 1)\}$$

must be built. This argument has a certainty degree of 0.6, taking into account the certainty degree of the literals on which the deduction is founded.

Next, the system looks for the defeaters. The only defeater is:

$$\langle \mathcal{A}_4, {\sim}\textit{climbs(simba)}, 0.6\rangle, \mathcal{A}_4 = \{({\sim}\textit{climbs(simba)} \prec \textit{species(simba,lion)}, 0.75)\}$$

But this argument is in turn defeated by $\langle \mathcal{A}_3, \textit{climbs(simba)}, 0.6\rangle$,

$$\mathcal{A}_3 = \{(\textit{climbs(simba)} \prec \textit{species(simba,lion),age(simba,young)}, 0.75)\}$$

Thus, *climbs(simba)* is warranted.

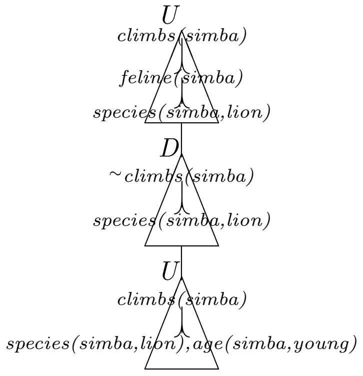


Fig. 4. Dialectical tree from Example 1

5 Optimization of DB_DeLP's Dialectical Process

To obtain faster query processing in the DB_DeLP system we integrate pre-compiled knowledge to avoid the construction of arguments which were already computed. The approach follows the proposal presented in [7] where the pre-compiled knowledge component is required to: (1) minimize the number of stored arguments in the pre-compiled base of arguments (for instance, using one structure to represent the set of arguments that use the same defeasible rules); and (2) maintain independence from the observations that may change with new perception in order to avoid modifying also the pre-compiled knowledge when new observations are incorporated.

Considering these requirements, we define a database structure called *dialectical graph*, which will keep a record of all possible *arguments* in an DB_DeLP database $\mathcal{DB}$ (by means of a special structure named potential argument) as well as the counterargument relation among them. Potential arguments, originally defined in [7], contain non-grounded defeasible rules, thus depending only on the set of rules in the **IDB**, *i.e.*, they are independent from the extensional database.

Potential arguments have been can be thought as schemata that sum up arguments that are obtained using *different* instances of the *same* defeasible rules. Recording every generated argument could result in storing many arguments which are structurally identical, only differing on the constants being used to build the corresponding derivations. Thus, a potential argument stands for several arguments which use the same defeasible rules. Attack relations among potential arguments can be also captured, and in some cases even defeat can be pre-compiled. In what follows we introduce the formal definitions, adapted from [7] to fit the DB_DeLP system.

Definition 7. [Weighted Potential Argument] *Let **IDB** be an intensional database. A subset* A *of **IDB** is a* potential argument *for a literal* Q *with an upper bound* γ *for its certainty degree, noted as* $\langle\!\langle \mathsf{A}, Q, \gamma \rangle\!\rangle$ *if there exists a non-contradictory set of weighted literals* Φ *and an instance* $\mathcal{A}$ *that is obtained by finding an instance for every rule in* A*, such that* $\langle \mathcal{A}, Q, \alpha \rangle$ *is an argument w.r.t. the database with* Φ *as its extensional database and **IDB** as its intensional database* ($\alpha \leq \gamma$) *and there is no instance* $\langle \mathcal{B}, Q, \beta \rangle$ *of* A *such that* $\beta > \gamma$*.*

Definition 7 does not specify how to obtain the set of potential arguments from a given database. The interested reader may consult [7] for a constructive definition and its associated algorithm. The calculation of the upper bound γ deserves special mention, since the algorithm in [7] was devised for a different system, without uncertainty management. This element will be used later on to speedup the extraction of the dialectical tree from the dialectical graph for a given query. To calculate γ for a potential argument A we simply choose the lower certainty degree of the defeasible rules present in A.

The nodes of the dialectical graph are the potential arguments. The arcs of our graph are obtained by calculating the counterargument relation among the nodes previously obtained. To do this, we extend the concept of counter-argument for potential arguments. A potential argument $\langle\!\langle \mathsf{A}_1, Q_1, \alpha \rangle\!\rangle$ *counter-argues* $\langle\!\langle \mathsf{A}_2, Q_2, \beta \rangle\!\rangle$ at a literal Q if and only if there is a non-empty potential sub-argument $\langle\!\langle \mathsf{A}, Q, \gamma \rangle\!\rangle$ of $\langle\!\langle \mathsf{A}_2, Q_2, \beta \rangle\!\rangle$ such that Q_1 and Q are contradictory literals.[2] Note that potential counter-arguments may or may not result in a real conflict between the instances (arguments) associated with the corresponding potential arguments. In some cases instances of these arguments cannot co-exist in any scenario (*e.g.*, consider two potential arguments based on contradictory observations). Now we can finally define the concept of dialectical graph:

Definition 8. [Dialectical Graph] *Let* $\mathcal{DB} = (\boldsymbol{EDB}, \boldsymbol{IDB}, \boldsymbol{IC})$ *be a defeasible database. The* dialectical graph *of **IDB**, denoted as* $\mathcal{G}_{\boldsymbol{IDB}}$*, is a pair* $(\mathrm{PotArgs}(\boldsymbol{IDB}), C)$ *such that: (1)* $\mathrm{PotArgs}(\boldsymbol{IDB})$ *is the set* $\{\langle\!\langle A_1, Q_1, \alpha_1 \rangle\!\rangle, \ldots, \langle\!\langle A_k, Q_k, \alpha_k \rangle\!\rangle\}$ *of all the potential arguments that can be built from **IDB**; (2)* $\mathcal{C}$ *is the counterargument relation over the elements of* $\mathrm{PotArgs}(\boldsymbol{IDB})$*.*

Example 2. Consider the feline database previously presented; its dialectical graph is composed by:
(`feline(X)` $\prec$ `species(X,lion)`,1)
(`climbs_tree(X)` $\prec$ `feline(X)`,0.65)
(~`climbs_tree(X)` $\prec$ `species(X,lion)`,0.70)

[2] Note that $P(X)$ and ~$P(X)$ are contradictory literals even though they are non-grounded. The same idea is applied to identify contradiction in potential arguments.

(`climbs_tree(X)` $\prec$ `species(X,lion), age(X,young)`,0.75)
(~`climbs_tree(X)` $\prec$ `sick(X)`,0.45)

- $\langle\!\langle \mathsf{A}_1, \mathsf{climbs(X)}, 0.65\rangle\!\rangle$, $\mathsf{A}_1 = \{(\texttt{climbs(X)} \prec \texttt{feline(X)}, 0.65)\}$.
- $\langle\!\langle \mathsf{A}_2, \mathsf{climbs(X)}, 0.65\rangle\!\rangle$, $\mathsf{A}_2 = \{(\texttt{climbs(X)} \prec \texttt{feline(X)}, 0.65),$
  $(\texttt{feline(X)} \prec \texttt{species(X,lion)}, 1)\}$.
- $\langle\!\langle \mathsf{A}_3, \mathsf{climbs(X)}, 0.75\rangle\!\rangle$,
  $\mathsf{A}_3 = \{(\texttt{climbs(X)} \prec \texttt{species(X,lion), age(X,young)}, 0.75)\}$.
- $\langle\!\langle \mathsf{A}_4, {\sim}\mathsf{climbs(X)}, 0.75\rangle\!\rangle$, $\mathsf{A}_4 = \{({\sim}\texttt{climbs(X)} \prec \texttt{species(X,lion)}, 0.75)\}$.
- $\langle\!\langle \mathsf{A}_5, {\sim}\mathsf{climbs(X)}, 0.45\rangle\!\rangle$, $\mathsf{A}_5 = \{({\sim}\texttt{climbs(X)} \prec \texttt{sick(X)}, 0.45)\}$.
- $\langle\!\langle \mathsf{A}_6, \mathsf{feline(X)}, 1\rangle\!\rangle$, $\mathsf{A}_6 = \{(\texttt{feline(X)} \prec \texttt{species(X,lion)}, 1)\}$.
- $D_p = \{(\mathsf{A}_2, \mathsf{A}_4), (\mathsf{A}_4, \mathsf{A}_3)\}$
- $D_b = \{(\mathsf{A}_1, \mathsf{A}_4), (\mathsf{A}_4, \mathsf{A}_1), (\mathsf{A}_1, \mathsf{A}_5), (\mathsf{A}_5, \mathsf{A}_1), (\mathsf{A}_2, \mathsf{A}_5), (\mathsf{A}_5, \mathsf{A}_2), (\mathsf{A}_3, \mathsf{A}_5), (\mathsf{A}_5, \mathsf{A}_3)\}$.

The relations D_b and D_p can be depicted as shown in Figure 2, where blocking defeat is indicated with a double headed arrow.

$$\begin{tikzcd}
\mathsf{A}_1 \arrow[d, leftrightarrow] \arrow[dr, leftrightarrow] & \mathsf{A}_2 \arrow[d, leftrightarrow] & \mathsf{A}_3 \arrow[dll] \arrow[dl, leftrightarrow] \\
\mathsf{A}_4 \arrow[ur] & \mathsf{A}_5 & \mathsf{A}_6
\end{tikzcd}$$

Fig. 5. Dialectical graph corresponding to Example 2.

Having defined the dialectical graph we can now use a specific graph traversing algorithm to extract a particular dialectical tree rooted in a given potential argument. The facts present in the **EDB** will be used as evidence to instantiate the potential arguments in the dialectical graph that depend on the intensional database **IDB**. This gives rise to the inference process of the system. This process starts when a new query is formulated. Consider the dialectical graph in Example 2 and suppose the set of facts in Figure 2 is present in the extensional database. Lets see how the system works when faced with the query *climbs(simba)*.

First, the set of potential arguments in the dialectical graph is searched to see if there exists an element whose conclusion can be instantiated to match the query. It finds $\langle\!\langle \mathsf{A}_2, \mathsf{climbs(X)}, 0.65\rangle\!\rangle$,

$$\mathsf{A}_2 = \{(\texttt{climbs(X)} \prec \texttt{feline(X)}, 0.65), (\texttt{feline(X)} \prec \texttt{species(X,lion)}, 1)\}$$

A_2 can be instantiated to

$$\mathcal{A}_2 = \{(\textit{climbs(simba)} \prec \textit{feline(simba)}, 0.65), (\textit{feline(simba)} \prec \textit{species(simba,lion)}, 1)\}$$

that has a certainty degree of 0.6 taking into account the certainty degree of the literals on which the deduction is founded.

Now, to see if *climbs(simba)* is inferred by the system from the intensional and the extensional database, we must check whether $\mathcal{A}_2$ can sustain its conclusion when confronted with its counterarguments. Using the links in the dialectical graph we find one defeater for $\mathcal{A}_2$, instantiating potential argument

$$\mathsf{A}_4 = \{(\sim\texttt{climbs(X)} \prec \texttt{species(X,lion)}, 0.75)\}$$

to

$$\mathcal{A}_4 = \{(\sim\textit{climbs(simba)} \prec \textit{species(simba,lion)}, 0.75)\}$$

The argument $\langle \mathcal{A}_4, \sim\textit{climbs(simba)}, 0.6\rangle$ is defeated by $\langle \mathcal{A}_3, \textit{climbs(simba)}, 0.6\rangle$ (an instantiation of $\langle\!\langle \mathsf{A}_3, \mathsf{climbs(X)}, 0.75\rangle\!\rangle$). Thus, *climbs(simba)* is warranted and we found the same dialectical tree that was found in example 1 with an optimized inference mechanism. Note that the links for the defeaters present in the dialectical graph are used to find the conflicts. This makes it easier to recover the tree from the dialectical graph of the framework.

The deductive database can be subject to constant changes as is the case with every real world database. The only restriction is that it must not be changed while a query is being solved. The dialectical graph is not affected by changes in the extensional database.

We present now a classic example in traditional deductive database systems based on logic programming, that usually causes problems with the semantics. In our case the system follows a cautious semantics, not deriving either $p(a)$ or $q(a)$.

Example 3. Consider a deductive database composed by:

- **EDB** $= \{(r, 0.6), (s, 0.6)\}$,
- **IDB** $= \{(\texttt{p(X)} \prec \sim\texttt{q(X)}, 0.8), (\texttt{q(X)} \prec \sim\texttt{p(X)}, 0.8)\}$

The dialectical graph $\mathbf{G}_{IDB}$ is composed by the two potential arguments:

- $\langle\!\langle \mathsf{A}_1, \mathsf{p(X)}, , \rangle\!\rangle$ $\mathsf{A}_1 = \{(\texttt{p(X)} \prec \sim\texttt{q(X)}, 0.8)\}$.
- $\langle\!\langle \mathsf{A}_2, \mathsf{q(X)}, , \rangle\!\rangle$ $\mathsf{A}_1 = \{(\texttt{q(X)} \prec \sim\texttt{p(X)}, 0.8)\}$.

and the defeat relation $D_b = \{(A_1, A_2), (A_2, A_1)\}$.

Suppose the system is faced with the query $p(a)$. The dialectical tree for this query is formed by argument $\langle \mathcal{A}_1, \sim\textit{q(a)}, 0.6\rangle$, $\mathcal{A}_1 = \{(\texttt{p(a)} \prec \sim\texttt{q(a)}, 0.6)\}$ that is in turn defeated by $\langle \mathcal{A}_2, \textit{q(a)}, 0.6\rangle$, $\mathcal{A}_1 = \{(\texttt{q(a)} \prec \sim\texttt{p(a)}, 0.6)\}$.

The situation with query q(a) is analogous and therefore the system cannot derive $p(a)$ nor $q(a)$.

Note that the DB_DeLP system can seamlessly treat this example without semantic or operational problems. Furthermore, there is no need for imposing additional restrictions, such as requiring predicate stratification. Traditional systems would enter a loop jumping from one rule to the other. This is prevented in DB_DeLP by the circularity condition imposed on argumentation lines of dialectical trees.[3] This condition does not allow the re-introduction of $\mathcal{A}_1$ as a defeater of $\mathcal{A}_2$ in the dialectical tree of the previous example.

[3] This condition was inherited from the original DeLP system, the interested reader may consult [16].

6 A Worked Example

In this section we present an example to illustrate the practical uses of defeasible databases. The example is based on a classical benchmark in deductive databases concerning data on prescriptions, physicians and patients [22]. The system is a DSS to help employees decide whether a given medication should be sold to a patient. The relation *prescription* (pres) means that there is a prescription for a given drug to be administered to a given patient. *Allergic* shows known allergic reactions in patients, *physician* lists where physicians work, *patient* lists insurance company and clinics to which a patient usually goes, and *psychiatrist* (psy) establishes that a physician is also a psychiatrist (see Figure 6).

`patients(patient_id,clinic,insurance)`
(`patients(456,new_line,hope)`, 0.6)
(`patients(587,delta,hope)`, 0.6)
(`patients(234,new_line,trust)`, 0.6)
(`patients(1211,delta,trust)`, 0.6)
(`patients(254,star,trust)`, 0.6)
...

`physicians(phy_id,clinic)`
(`physicians(432,star)`, 0.7)
(`physicians(54,delta)`, 0.7)
(`physicians(672,new_line)`, 0.7)
(`physicians(432,delta)`, 0.7)
...

`pres(note_id,patient_id,phy_id,drug,text)`
(`pres(23445,587,432,pen,text1)`, 0.6)
(`pres(23446,587,54,amoxicillin,text2)`, 0.6)
(`pres(23447,587,54,vicodin,text3)`, 0.6)
(`pres(23448,1211,54,morphine,text4)`, 0.6)
(`pres(23449,234,672,diazepam,text5)`, 0.6)
...

`allergic(patient_id,drug)`
(`allergic(587,pen)`, 0.7)
(`allergic(1211,pen)`, 0.6)
(`allergic(1211,morphine)`, 0.6)
...

`psy(phy_id)`
(`psy(672)`, 0.8)
(`psy(54)`, 0.8)
...

The intensional database is formed by the rules in Figure 6. The first rule says that a medication should be sold if there is prescription for it. The second rule says that it should not be sold if the physician is suspended and the third says that it should not be sold if the patient is allergic. The fourth rule concerns special drugs that have to be authorized before being sold and for that should have been prescribed by a psychiatrist. The fifth rule establishes the drug is not authorized when it is prescribed by a psychiatrist that has been suspended.

The dialectical graph contains arguments A, B, C, D, and E:

– $\langle\langle \mathsf{A}, \mathsf{sell(Patient,Drug)}, 0.65\rangle\rangle$,
  $\mathsf{A} = \{(\texttt{sell(Patient,Drug)} \mathrel{-\!\!\prec} \texttt{pres(X,Patient,Y,Drug)}, 0.65)\}$.
– $\langle\langle \mathsf{B}, \sim\mathsf{sell(Patient,Drug)}, 0.75\rangle\rangle$,
  $\mathsf{B} = \{(\sim\texttt{sell(Patient,Drug)} \mathrel{-\!\!\prec} \texttt{pres(X,Patient,Y,Drug,Text),susp(Y)}, 0.75)\}$.
– $\langle\langle \mathsf{C}, \sim\mathsf{sell(Patient,Drug)}, 0.95\rangle\rangle$,
  $\mathsf{C} = \{(\sim\texttt{sell(Patient,Drug)} \mathrel{-\!\!\prec} \texttt{allergic(Patient,Drug)}, 0.95)\}$.

(`sell(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug)`,0.65)
(~`sell(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug),susp(Y)`,0.75)
(~`sell(Patient,Drug)` $\prec$ `allergic(Patient,Drug)`,0.95)
(`auth_pres(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug),psychiatrist(Y)`,0.6)
(~`auth_pres(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug),psy(Y),susp(Y)`,0.7)

- $\langle\langle$D, authorize_pres(Patient,Drug), $0.6\rangle\rangle$,
  D $= \{($`auth_pres(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug,Text),psy(Y)`$, 0.6)\}$.
- $\langle\langle$E, ~authorize_pres(Patient,Drug), $0.7\rangle\rangle$,
  E $= \{($~`auth_pres(Patient,Drug)` $\prec$ `pres(X,Patient,Y,Drug,Text),psy(Y),susp(Y)`$, 0.7)\}$.

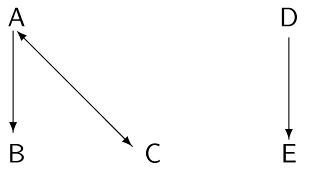


Fig. 6. Dialectical graph for clinical database.

Suppose the system is faced with a query regarding the fact sell(587,vicodin). It first finds a potential argument that can be instantiated to support this fact, so it selects A and instantiates it to:
$\mathcal{A} = \{($`sell(787,vicodin)` $\prec$ `pres(23447,587,54,vicodin,text3)`$, 0.6)\}$. Using the dialectical graph we can see that there are two links that connect A with its defeaters, so we can explore to see if an instance of B or C can be built to attack argument $\mathcal{A}$. Since this is not the case argument $\mathcal{A}$ is the only argument in the dialectical tree and the answer is yes.

Next, the system is faced with query sell(587,pen). The structure is similar to the previous case, but in this situation potential argument A is instantiated to

$$\mathcal{A} = \{(\texttt{sell(587,pen)} \prec \texttt{pres(23445,587,432,pen,text1)}, 0.6)\}$$

and following the links in the dialectical graph we find defeater B that can be instantiated to:
$\mathcal{B} = \{($~`sell(587,pen)` $\prec$ `allergic(587,pen)`$, 0.7)\}$. No more defeaters can be added to this dialectical tree so the answer to sell(587,pen) is no.

Now the query auth_pres(234,diazepam) is performed. In this case potential argument D is instantiated to:
$\mathcal{D} = \{$`auth_pres(234,diazepam)` $\prec$ `pres(23449,234,672,diazepem,text5),`
`psy(672)`$, 0.6)\}$ and no defeater can be found for $\mathcal{D}$ thus the answer is yes.

Facts can be added to the database and also new tables can be created. Suppose we add a new table that contains a list of doctors that have been suspended due to legal issues. This table contains the fact (`suspended(672)`$, 0.8)$. If query

authorize_pres(234,diazepam) is re-processed by the system the answer would now be no, given that a new argument:
$\mathcal{E} = \{(\sim$`auth_pres(234,diazepam)` $\prec$ `pres(23449,234,672,diazepam,text5), psychiatrist(672),suspended(672)`$, 0.7)\}$. can be built by instantiating E, resulting in a defeater for $\mathcal{D}$. Thus, $\mathcal{D}$ is no longer warranted. Note how new tables and new facts can be added to the system without the need for rebuilding the dialectical graph.

7 Conclusions

In this work, we have defined a multi-agent system which virtually integrates different databases into a common view. We have also presented a layered architectural model that we have designed to develop practical applications for reasoning with data from multiple sources. This model provides a novel system architecture for decision-support systems (DSS) that combines database technologies with an argumentation based framework.

We have also defined an argumentation-based formalism that integrates uncertainty management and is specifically tailored for database integration. This formalism was also provided with an optimization mechanism based on precompiled knowledge. Using this mechanism, the argumentation system can comply with real time requirements needed to manage data and model reasoning over this data in dynamic environments.

Future work may be done in different directions. In the first place we plan to integrate DB_DeLP with a massive data component to obtain experimental results regarding the system's complexity. We also plan to extend the language of DB_DeLP to use it in practical systems, particularly to implement argumentation-based active databases.

References

1. T. Alsinet, C. I. Chesñevar, Lluis Godo, and G. R. Simari. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*, 159(10):1208–1228, 2008.
2. T. Alsinet and L. Godo. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 1–10. ACM Press, 2000.
3. L. Berti. Quality and recommendation of multi-source data for assisting technological intelligence applications. In *Proc. of 10th International Conference on Database and Expert Systems Applications*, pages 282–291, Italy, 1999. AAAI.
4. M. L. Brodie and M. Jarke. On integrating logic programming and databases. In *Expert Database Workshop 1984*, pages 191–207, 1984.
5. D. Bryant and P. Krause. An implementation of a lightweight argumentation engine for agent applications. *Logics in Artificial Intelligence, Lecture Notes in Computer Science*, 4160(1):469–472, 2006.
6. M. Capobianco, C. I. Chesñevar, and G. R. Simari. Argumentation and the dynamics of warranted beliefs in changing environments. *Journal of Autonomous Agents and Multiagent Systems*, 11:127–151, 2005.

7. M. Capobianco, C. I. Chesñevar, and G. R. Simari. Argumentation and the dynamics of warranted beliefs in changing environments. *Journal of Autonomous Agents and Multiagent Systems*, 11:127–151, 2005.
8. D. Carbogim, D. Robertson, and J. Lee. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review*, 15(2):119–149, 2000.
9. S. Ceri, G. Gottlob, and L. Tanca. What you always wanted to know about datalog (and never dared to ask). *IEEE Trans. on Knowledge and Data Eng.*, 1(1), 1989.
10. C. I. Chesñevar, A. G. Maguitman, and R. P. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
11. C. I. Chesñevar, A. G. Maguitman, and G. R. Simari. Argument-based critics and recommenders: A qualitative perspective on user support systems. *Data & Knowledge Engineering*, 59(2):293–319, 2006.
12. C. I. Chesñevar, G. R. Simari, T. Alsinet, and L. Godo. A logic programming framework for possibilistic argumentation with vague knowledge. In *Proc. of Uncertainty in Artificial Intelligence Conference (UAI 2004), Banff, Canada*, 2004.
13. F. Cuppens and R. Demolombe. Cooperative answering: a method to provide intelligent access to databases. In *Proc. 2nd Conf. on Expert Database Systems*, pages 621–643, 1988.
14. D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.Gabbay, C.Hogger, and J.Robinson, editors, *Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning)*, pages 439–513. Oxford Univ. Press, 1994.
15. P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–357, 1995.
16. A. J. García and G. R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
17. L. V. S. Lakshmanan and F. Sadri. Probabilistic deductive databases. In *Proc. of the Int. Logic Programming Symposium*, pages 254–268, 1994.
18. L. V.S. Lakshmanan and N. Shiri. A parametric approach to deductive databases with uncertainty. *Journal of Intelligent Information Systems*, 13(4):554–570, 2001.
19. D. McLeod and D. Heimbigner. A federated architecture for information management. *ACM Transactions on Information Systems*, 3(3):253–278, 1985.
20. H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In *Handbook of Philosophical Logic*, volume 4, pages 219–318. 2002.
21. H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In *Handbook of Philosophical Logic*, volume 4, pages 219–318. 2002.
22. X. Quian. Query folding. In *Proc. 12th Intl. Conf on Data Engineering*, pages 48–55, 1996.
23. G. R. Simari and R. P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53(1–2):125–157, 1992.
24. V. S. Subrahmanian. Paraconsistent disjunctive deductive databases. *Theorethical Computer Science*, 93(1):115–141, 1992.
25. C. Zaniolo. Prolog: A database query language for all seasons. In *Expert Database Workshop 1984*, pages 219–232, 1984.
26. C. Zaniolo. Intelligent databases: Old challenges and new opportunities. 3/4(1):271–292, 1992.

On the Benefits of Argumentation-derived Evidence in Learning Policies

Chukwuemeka David Emele[1], Timothy J. Norman[1], Frank Guerin[1], and Simon Parsons[2]

[1] University of Aberdeen, Aberdeen, AB24 3UE, UK
{c.emele, t.j.norman, f.guerin}@abdn.ac.uk
[2] Brooklyn College, City University of New York, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu

Abstract. An important and non-trivial factor for effectively developing and resourcing plans in a collaborative context is an understanding of the policy and resource availability constraints under which others operate. We present an efficient approach for identifying, learning and modeling the policies of others during collaborative problem solving activities. The mechanisms presented in this paper will enable agents to build more effective argumentation strategies by keeping track of who might have, and be willing to provide the resources required for the enactment of a plan. We argue that agents can improve their argumentation strategies by building accurate models of others' policies regarding resource use, information provision, etc. In a set of experiments, we demonstrate the utility of this novel combination of techniques through empirical evaluation, in which we demonstrate that more accurate models of others' policies (or norms) can be developed more rapidly using various forms of evidence from argumentation-based dialogue.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multi-agent Systems

General Terms

Algorithms, Experimentation

Keywords

Argumentation, Machine learning, Policies, Norms

1 Introduction

Distributed problem solving activities often require the formation of a team of collaborating agents. In such scenarios agents often operate under constraints placed on them by the organisations or interests that they represent. When these constraints are part of the standard operating procedures of the agents or the organisations in question, we refer to them as *policies* (also known as norms). Members of the team agree to collaborate and perform joint activities in a mutually acceptable fashion. Often, agents in the team represent different organisations, and so there are different organisational constraints imposed on them. Even within a single organisation, team members often represent sub-organisations with different procedures and constraints. Examples of such constraints are those due to policies that guide the behaviour of representatives of organisations. Furthermore, team members may possess individual interests and goals that they seek to satisfy, which are not necessarily shared with other members of the team. These individual motivations largely determine the way in which members carry-out tasks assigned to them during joint activities.

In this paper, we focus on policy and resource availability constraints, and define policies as explicit prohibitions that members of the team are required to adhere to. Policy constraints may be team-wide or individual. We focus on individual policies. These policies are often private to that individual member or subset of the team, and are not necessarily shared with other members of the team. In order to develop effective plans, an understanding of the policy and resource availability constraints of other members in the team is beneficial. However, tracking and reasoning about such information is non-trivial.

Our conjecture is that machine learning techniques may be employed to aid decision making in this regard. Although this is not a new claim [7], it is novel to combine it with evidence derived from argumentation-based dialogue, which we call argumentation-derived evidence (ADE). We present a system where agents learn from dialogue by automatically extracting useful information (evidence) from the dialogue and using these to model the policies of others in order to adapt their behaviour in the future. We describe an experimental framework and present results of our evaluation in a resource provisioning scenario [5], which show empirically (1) that evidence derived from argumentation-based dialogue can indeed be effectively exploited to learn better (more complete and correct) models of the policy constraints that other agents operate within; and (2) that through the use of appropriate machine learning techniques more accurate and stable models of others' policies can be derived more rapidly than with simple memorisation of past experiences.

For example, consider the following snippet of dialogue that may occur between two agents i and j collaborating to hang a picture [11].

Example 1:	**Example 2:**
i: Can I have a *screw-driver*?	i: Can I have a *screw-driver*?
j: What do you want to use it for?	j: What do you want to use it for?
i: To hang a picture.	i: To hang a picture.
j: No.	j: I can provide you with a *hammer* instead.
	i: I accept a *hammer*.

Following from the interaction in example 1, there is very little that we can learn from the encounter. It is unclear why agent j said no to agent i's request. It could be that there exists some policy X that forbids agent

j from providing the *screw-driver* to agent i or it could be that the *screw-driver* is not available at the moment. On the other hand, suppose we have an argumentation framework that allows agents to suggest alternatives as in example 2 or ask for and receive explanations as in examples 3 and 4, then agent i can, potentially, gather more evidence regarding the provision of the resources involved.

Example 3:	**Example 4:**
i: Can I have a *screw-driver*?	i: Can I have a *screw-driver*?
j: What do you want to use it for?	j: What do you want to use it for?
i: To hang a picture.	i: To hang a picture.
j: No.	j: No.
i: Why?	i: Why?
j: I'm not permitted to release the *screw-driver*.	j: *Screw-driver* is not available.

Considering examples 3 and 4, it is worth noting that without the additional evidence, obtained by the information-seeking dialogue, the two cases are indistinguishable. This means that the agent will effectively be guessing which class these cases fall into. The additional evidence allows the agent to learn the right classification for each of the cases. It should be noted here that although in example 3, we now have a statement that the resource is not to be provided for policy reasons, the question remains: what are the important characteristics of the prevailing circumstances that characterise this policy?

In a domain where there are underlying constraints that could yield similar results, standard machine learning techniques will have limited efficacy. Using argumentation to gather additional evidence could improve the accuracy of the information learned about the policies of others. We claim that significant improvements can be achieved because argumentation can help clarify reasons behind decisions made by the *provider*.

In the research presented in this paper, we intend to validate the following hypotheses: (1) Allowing agents to exchange arguments during practical dialogue (like negotiation) will mean that the proportion of correct policies learned during interaction will increase faster than when there is no exchange of arguments. (2) Through the use of appropriate machine learning techniques more accurate and stable models of others' policies can be derived more rapidly than with simple memorisation of past experiences.

The remainder of this paper is organised as follows: In section 2 we briefly describe argumentation-based dialogue and introduce the protocol employed. Learning policies is discussed in section 3 and section 4 describes our simulation environment. Experimental results are reported in section 5. Section 6 discusses related work and future direction, and the paper concludes in section 7.

2 Argumentation-based Dialogue

In this section we present the argumentation-based negotiation protocol which will be used in guiding the negotiation process, and for obtaining additional evidence from the interaction. This protocol uses information-seeking dialogue [17] to probe for additional evidence.

2.1 The Negotiation Protocol

The negotiation for resources takes place in a turn-taking fashion, where the *seeker* agent sends a request for resource to a *provider* agent. Figure 1 captures the negotiation protocol in a AUML-like interaction diagram (*www.fipa.org*). If the *provider* agent has the requested resource in its resource pool and it is in a usable state then it checks whether there is any policy constraint that forbids it from providing the resource to the *seeker* or not. If the *provider* agent needs more information from the *seeker* in order to make a decision, the *provider* agent would ask for more information to be provided. This is the information gathering stage. The information gathering cycle will continue until the *provider* has acquired enough information (necessary to make the decision), or the *seeker* refuses to provide more information and the negotiation ends.

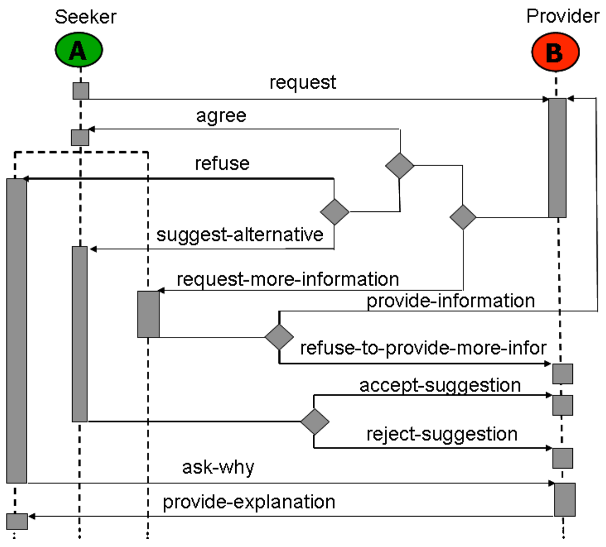


Fig. 1. The negotiation protocol.

The *provider* agent releases the resource to the *seeker* agent if there is no policy that prohibits the *provider* agent from doing so. Otherwise, the *provider* agent offers an alternative resource (if there are no policies that forbid that line of action and the alternative resource is available). When an alternative resource is suggested by the *provider* agent, the *seeker* agent evaluates it. If it is acceptable, the *seeker* agent accepts it and the negotiation ends. Otherwise, the *seeker* agent refuses the alternative (in principle, this cycle may be repeated until an alternative is accepted or the negotiation ends). However, for simplicity and brevity, only one suggest-refuse cycle is permitted per request.

From a learning point of view, the suggestion of alternative resources is a positive evidence that the *provider* agent does not have any policy that forbids the provision of the alternative resource to the *seeker*. In addition, it provides an evidence that the alternative resource is also available. This extra evidence, we anticipate, may help to improve the performance of the learner in predicting the policy constraints of the *provider* agents in future encounters.

If there is a policy constraint that forbids the provision of the resource, or the resource is not available then the *provider* agent will refuse to provide the resource to the *seeker* agent. From the *seeker*'s perspective, the refusal could be as a result of policy constraint or because the resource is not available. In order to disambiguate which of these constraints are responsible for the refusal, the *seeker* agent switches to argumentation based dialogue. The *seeker* agent asks for explanations for the refusal so as to gather further evidence and thereby identify the underlying constraints. The *provider* agent, therefore, responds with some explanations and the negotiation ends. Three categories of explanations are possible in this framework: (1) Policy constraints (2) Resource not available (3) Won't tell you. These pieces of evidence will be explored in the following section.

2.2 Argumentation-derived Evidence

Following the argumentation-based negotiation protocol described earlier, the agents could ask for more information (with respect to a request or the response to a request), which indicates what constraints others may be operating within. For instance, let us assume that a *provider* agent has a policy that forbids it from providing a *screw-driver* to any *seeker* agent that intends to use it for *hanging a picture*. Then, whenever a *screw-driver* is requested the *provider* agent will probe for more information to ascertain that the purpose the *seeker* intends to use the *screw-driver* for is not *hanging a picture*. This extra evidence could be useful. Similarly, whenever a *seeker* agent's request is refused then the *seeker* agent will ask for explanations/justifications for the refusal. These additional evidence are beneficial, and we expect them to improve the quality of the models of other agents that can be inferred in future encounters.

Figure 2 shows two simple examples of the kind of dialogue that may occur between two agents, i and j. For the purpose of the example, we use **need(**R**, P, L, D)** to denote that the *seeker* agent intends to use the resource R for purpose P at location L on day D. Note that although this is presented as a dialogue between two agents, in reality the initiator (agent i, the agent that wishes to resource its plan) may engage in multiple instances of this dialogue with other agents.

3 Learning Policies

In this section we discuss the machine learning techniques that we have explored for learning policies through argumentation-derived evidence. These techniques include decision tree learning (C4.5), instance-based

	Example A
i:	**request**(i, j, *screw-driver*)
j:	**ask-infor**(j, i, need(*screw-driver*, P, L, D))
i:	**provide-infor**(i, j, need(*screw-driver*, P=x, L=y, D=z))
j:	**refuse**(j, i, *screw-driver*)
i:	**why**(i, j, refuse(*screw-driver*))
j:	**inform**(j, i, *screw-driver*, reason(policy-constraints))

	Example B
i:	**request**(i, j, *nail*)
j:	**refuse**(j, i, *nail*)
i:	**why**(i, j, refuse(*nail*))
j:	**inform**(j, i, *nail*, reason(wont-tell-you))
i:	**request**(i, j, *table*)
j:	**agree**(j, i, *table*)

Fig. 2. Dialogue snippets between agents i and j

learning (k-Nearest Neighbours, abbreviated as k-NN) and rule-based learning (Sequential Covering, abbreviated as SC).

Our technique does not attempt to replace machine learning nor compete with existing techniques. Rather, we seek ways to combine argumentation analysis with already existing machine learning techniques with a view to improving the performance of agents at predicting the policy constraints of others. We anticipate that this could enable them to build more effective argumentation strategies. In other words, we argue that evidence derived from argumentation-based dialogue can indeed be effectively exploited to learn better (more complete and correct) models of the policy constraints that other agents operate within. Also, we claim that through the use of appropriate machine learning techniques more accurate and stable models of others' policies can be derived more rapidly than with simple memorisation of past experiences. In future encounters, the *seeker* agent attempts to predict the policies of the *provider* agent based on the model it has built.

3.1 Decision Tree Learning (C4.5)

C4.5 [13] builds decision trees from a set of training data, using the concept of information entropy [8] (beyond the scope of this paper). Generally, the training data is a set $S = s_1, s_2, ..., s_n$ of already classified samples. Each sample $s_i = x_1, x_2, ..., x_m$ is a vector where $x_1, x_2, ..., x_m$ represent attributes of the sample. The training data is augmented with a vector $C = c_1, c_2, ..., c_n$ where $c_1, c_2, ..., c_n$ represent the class to which each sample belongs.

Integrating this algorithm into our system with the intention of learning policies is appropriate since the algorithm supports concept learning and policies can be conceived as concepts/features of an agent. Agent policies are represented as a vector of attributes (e.g. resource, purpose, location, etc.) and these attributes are communicated back and forth during negotiation. The C4.5 algorithm is then used to classify each set of attributes (policy instance) into a class. There are two classes: grant and deny. Grant means that the *provider* agent will possibly provide the resource that is requested while deny implies that the *provider* agent will potentially refuse. The leaf nodes of a decision tree hold the class labels of the instances while the non-leaf nodes hold the test attributes. In order to classify a test instance, the C4.5 algorithm searches from the root node by examining the value of test attributes until a leaf node is reached and the label of that node becomes the class of the test instance.

The problem with this algorithm is that it is not incremental, which means all the training examples should exist before learning. To overcome this problem, the system keeps track of the *provider* agent's responses. After a number of interactions, the decision tree is rebuilt. Without doubt, there is a computational drawback involved in periodically reconstructing the decision tree. However, in practice, we have evaluated C4.5 to be fast and the reconstruction cost to be small. Our approach is similar to the incremental induction of decision trees proposed in [16].

The C4.5 algorithm has three base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

```
Algorithm 1. The C4.5 algorithm
1: Check for base cases
2: For each attribute $D$,
     Find the normalised information gain from
     splitting on $D$
3: Let $D\_best$ be the attribute with the highest
   normalised information gain
4: Create a decision node that splits on $D\_best$
5: Recurse on the sublists obtained by splitting on
   $D\_best$, and add those nodes as children of the node
```

Fig. 3. The C4.5 algorithm.

3.2 Instance-based Learning (*k*-NN)

The k-nearest neighbours algorithm (k-NN) [3] is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The universal set of all the policies an agent may be operating within could be conceived as a feature space (or a grid) and the various policy instances represent points on the grid. Using k-NN, a policy instance is classified by a majority vote of its neighbours, with the policy instance being assigned to the class most common amongst its k nearest neighbours, where k is a positive integer, typically small. The k-NN algorithm is incremental, which means all the training examples need not exist at the beginning of the learning process. This is a good feature because the policy model could be updated as new knowledge is learned.

The k-nearest neighbour algorithm is sensitive to the local structure of the data and this, interestingly, makes k-NN a good candidate for learning policies because slight changes in the variables/attributes of a policy could trigger different action. For example:

Policy1: *You are **permitted** to release resource R to team member X if his affiliation is O and R is to be deployed at location L for purpose P on day* 1.

Policy2: *You are **prohibited** from releasing resource R to team member X if his affiliation is O and R is to be deployed at location L for purpose P on day* 2.

In order to identify neighbours, the policy instances are represented by position vectors in a multidimensional feature space. In this approach, new policy instances are classified based on the closest training examples in the feature space. A policy instance is assigned to the class c if it is the most frequent class label among the k nearest training samples. It is usual to use the Euclidean distance, though other distance measures, such as the Manhattan distance, Hamming distance could in principle be used instead. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the actual classification phase, the test sample is represented as a vector in the feature space. Distances from the new vector to all stored vectors are computed and k closest samples are selected.

A major drawback to using this technique to classify a new vector to a class is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the k nearest neighbours when the neighbours are computed due to their large number. The distance-weighted k-NN algorithm, which weights the contribution of each of the k neighbours according to their distance to the new vector, uses distance weights to minimise the bias caused by the imbalance in the training examples by giving greater weight to closer neighbours. In our work, the weight of a neighbour is computed as the inverse of its distance from the new vector.

3.3 Rule-based Learning (Sequential Covering)

Since policies guide the way entities within a community (or domain) act by providing rules for their behaviour it makes sense to learn policies as rules. Sequential covering algorithm [8, 2] is a rule-based learning technique, which constructs rules by sequentially covering the examples. The sequential covering algorithm, SC for short, is a method that induces one rule at a time (by selecting attribute-value pairs that satisfy the rule), removes the data covered by the rule and then iterates the process. SC generates rules for each class by looking at the training data and adding rules that completely describe all tuples in that class. For each class value, rule antecedents are initially empty sets, augmented gradually for covering as many examples as possible. Figure 4 outlines the sequential covering algorithm in pseudo-code.

```
    Algorithm 2. The Sequential Covering Algorithm
 1: Input the training data ($D$) and the classes ($C$)
 2: For each class $c \in C$
 3: Initialise $E$ to the instance set
 4: Repeat
 5:    Create a rule $R$ with an empty left-hand
          side (LHS) that predicts class $c$:
 6:    Repeat
 7:      For each ($Attribute, Value$) pair found in $E$
 8:         Consider adding the condition
                $Attribute = Value$ to the LHS of $R$
 9:      Find $Attribute = Value$ that maximises $p/t$
10:         (break ties by choosing the condition with
             the largest $p$)
11:      Add $Attribute = Value$ to $R$
12:    Until $R$ is perfect (or no more attributes to use)
13:    Remove the instances covered by $R$ from $E$
14: Until $E$ contains no more instances that belong to $c$
```

Fig. 4. The Sequential Covering Algorithm.

In this study we used three different machine learning mechanisms: *Decision tree learning*, *Instance-based learning* and *Rule-based learning*. These three mechanisms represent very different classes of machine learning algorithms. The rationale for exploring a range of learning techniques is to demonstrate the utility of argumentation-derived evidence regardless of the machine learning technique employed. Thus, we hypothesize that the use of evidence acquired through argumentation significantly improves the performance of machine learning in the development and refinement of models of other agents. Also, we claim that through the use of appropriate machine learning techniques more accurate and stable models of others' policies can be derived more rapidly than with simple memorisation of past experiences.

4 Simulation Environment

To test our hypotheses, we developed a simulation environment that combines mechanisms for agents to engage in argumentative dialogue and to learn from dialogical encounters with other agents. For the purpose of resourcing plans, agents may act as resource seekers, which collaborate and communicate with potential providers to perform joint actions. The enactment of both *seeker* and *provider* roles are governed by individual policies that regulate their actions. A *seeker* agent requires resources in order to carry out some assigned tasks. The *seeker* agent generates requests in accordance with its policies and negotiates with the *provider* agents based on these constraints. On the other hand, *provider* agents have access to certain resources and may have policies that govern the provision of such resources to other members of the team.

Although agents may have prior assumptions about the policies that constrain the activities of others, these models are often incomplete and may be inaccurate. *Provider* agents do not have an unlimited pool of resources and so some resources may be temporarily unavailable. By a resource being available we mean that it is not committed to another task (or agent) at the time requested and the resource is in a usable state. Both *seeker* and *provider* agents have access to the team-wide policies but not the individual policies of others. Agents in this domain play the role of a *seeker* or a *provider* in different interactions.

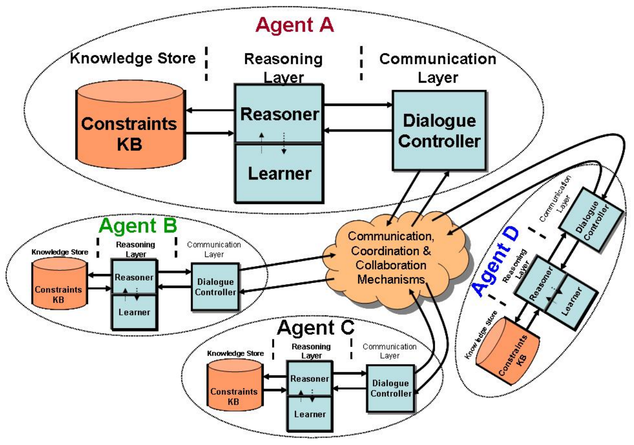


Fig. 5. Architecture of the framework for learning policies in team-based activities using dialogue.

4.1 Architecture

Figure 5 depicts our architecture. Each agent has two main layers, the communication layer and the reasoning layer. The communication layer embodies the dialogue controller, which handles all communication with other agents in the domain. The dialogue controller sends/receives messages to/from other agents, and the reasoning layer reasons over the dialogue. If an agent is playing the role of a *seeker* agent then the dialogue controller sends out the request for resources. On the other hand, if the agent is a *provider* agent then the dialogue controller receives a request and passes it on to the reasoning layer.

The reasoning layer consists of two modules: the reasoner and the learner. Upon receiving a message (e.g. a request), the reasoner evaluates the message and determines the response of the agent. In most cases, the reasoner looks up policy constraints from the knowledge-base and generates the appropriate response for the agent. Policy and non-policy constraints are stored in the constraints knowledge-base. Whenever the agent observes a new pattern of behaviour the agent uses this experience as evidence for learning, and updates the model of the other agent accordingly. The learner uses standard machine learning techniques to learn policies based on the perceived actions of other agents. The learning techniques are discussed in Section 3.

The knowledge store in Figure 5 acts as a repository where an agent stores the constraints it has learned by interacting with other agents in the domain. The information includes the features that an agent requires in order to make a decision about providing a resource or not. For example, following from [11], a *provider* agent B may need to know what the purpose for requesting a *screw-driver* is before deciding whether to release the *screw-driver* or not. The *seeker* agent stores such information about agent B in the knowledge store. Also, the decision of B after the purpose has been revealed will also be learned for future interactions.

To achieve this, we have developed a simple dialogue game[3] involving *seeker* agents and *provider* agents operating under different constraints. The players take turns and the game starts with an agent, i, sending a request to another agent, j, for the use of some resources needed to fulfill a plan. The other agent (j) responds with an agree or refuse based on the prevailing context, e.g. policy constraints. The requesting agent could ask for explanations and reasons for an action, and so on until the game ends.

4.2 Implementation

We implemented a simulation environment for agent support in team-based problem solving and integrated our learning and argumentation mechanisms into the framework. The policies are encoded as rules in a rule engine [6]. The application programming interface in Weka [18] was used to integrate standard machine learning algorithms into the framework. We note that, although these three learning algorithms were used, the framework is configured such that other machine learning algorithms can be plugged in. As discussed in the previous section, we evaluated the performance of a decision tree learner (C4.5), an Instance based learner (k-Nearest Neighbour algorithm) and a rule based learner (Sequential Covering) in learning policies through argumentation-derived evidence.

The simulation environment allows us to generate multiple *providers* with randomised policies, *seeker* agents with randomised initial models of the policies of *providers* in the simulation and randomised problems for the *seeker* to solve (that is, random resource requirements). The *seeker* predicts (based on the model of the *provider*) whether the *provider* has a policy that forbids/permits the provision of such resource in that context. The *seeker* requests the required resource from the *provider* agent and the *provider* uses a

[3] Dialogue games have proven extremely useful for modeling various forms of reasoning in many domains [1].

simple decision function (See Figure 6) to decide whether to grant or deny the request.

If the decision of the *provider* agent deviates from the predictions of the *seeker* agent then the *seeker* agent seeks additional evidence (through dialogue) to disambiguate whether the deviation was as a result of policy or resource availability constraints. The dialogue follows the protocol specified in Figure 1, and at the end of the interaction the outcome is learned by the *seeker* and the model of the *provider* is updated accordingly. This adaptive learning process serves to improve the quality of the models of the other agents that can be inferred from their observable actions in future interactions.

```
Assume seeker A requests resource R from provider P
IF      ( is_available($R$) $\wedge$ NOT ($forbid$(release($R$, A)) )
THEN    agree( release($R$, A))
ELSE    refuse( release($R$, A))
```

Fig. 6. *Provider* agents' pseudo decision function

5 Experiments and Results

In a series of experiments, we show how learning techniques and argumentation can support agents engaging in collaborative activities, increase their predictive accuracy, avoid unnecessary policy conflicts, hence improve their performance. The experiments show that agents can effectively and rapidly increase their predictive accuracy of the learned model through the use of dialogue.

The scenario adopted in this research involves a team of five software agents (one *seeker* and four *provider* agents) collaborating to complete a joint activity in a region over a period of three days. The region is divided into five locations. There are five resource types, and five purposes that a resource could be used to fulfill. A task involves the *seeker* agent identifying resource needs for a plan and collaborating with the *provider* agents to see how that plan can be resourced.

Argumentation-derived evidence (ADE) was incorporated into the learning process of the three machine learning techniques (C4.5, *k*-NN, and SC) described earlier, and their performances in learning the policy constraints of others were evaluated. A simple lookup table (hereafter called, LT) was used as a control condition and it serves as a structure for simple memorisation of outcomes from past encounters.

5.1 Results

This section presents the results of the experiments carried out to validate this work. Experiments were conducted with *seeker* agents ini-

Table 1. Average percentage of policies classified correctly and standard deviation

Approach \ Tasks	1000	2000	3000	4000	5000	6000
LT-ADE	65.1±6.5	70.3±10.3	75.6±6.7	78.1±10.2	79.3±8.3	81.3±10.1
LT+ADE	66.3±6.0	79.3±9.3	83.6±8.2	81.7±11.2	81.4±7.8	84.7±9.1
C4.5-ADE	58.3±15.1	69.2±16.6	75.1±12.0	82.1±12.3	85.3±8.9	88.2±8.2
C4.5+ADE	60.3±14.4	75.0±12.6	83.6±6.5	89.9±5.2	93.0±3.4	95.6±5.1
k-NN-ADE	65.2±9.8	71.0±7.8	75.3±5.3	80.7±3.8	81.0±4.1	82.0±3.8
k-NN+ADE	71.1±9.0	85.9±7.3	92.0±4.6	96.8±3.1	97.3±3.6	98.4±1.7
SC-ADE	66.7±8.2	71.7±6.0	78.7±8.4	84.3±6.5	87.4±6.0	90.6±5.3
SC+ADE	67.7±7.7	87.1±6.4	94.1±4.2	96.6±4.1	97.5±2.6	99.2±1.0

tialised with random models of the policies of *provider* agents. 100 runs were conducted for each case, and tasks were randomly created during each run from 375 possible configurations.

Table 1 illustrates the effectiveness of identifying and learning policies through argumentation-derived evidence using the three machine learning techniques described earlier, and the control condition (lookup table). It shows the average percentage of policies classified correctly and the standard deviations for each of the approaches, namely: Lookup Table without the aid of argumentation-derived evidence (LT-ADE), Lookup Table enhanced with argumentation-derived evidence (LT+ADE), C4.5-ADE, C4.5+ADE, *k*-NN-ADE, *k*-NN+ADE, SC-ADE, and SC+ADE. In each case, the model of others' policies is recomputed after each set of 1000 tasks. For all three machine learning techniques considered, the percentage of policies predicted correctly as a result of exploiting evidence derived from argumentation was consistently and significantly higher than those predicted without such evidence. Figure 7 gives a graphical illustration of the effectiveness of learning policies with the aid of argumentation-derived evidence using rule-based learning technique, for instance. After 3000 tasks, the accuracy of the approach with additional evidence had risen above 94% while the configuration without additional evidence was approaching 79%. It is easy to see that the experiments where additional evidence was combined with machine learning significantly and consistently outperformed those without additional evidence. These results show that the exchange of arguments during practical dialogue enabled agents to learn and build more accurate models of others' policies much faster than scenarios where there was no exchange of arguments.

Figure 8 captures the effectiveness of the three machine learning techniques described earlier, and a simple memorisation technique (a lookup table) in learning policies. The result shows that both instance-based learning (*k*-NN+ADE) and rule-based learning (SC+ADE) constantly and consistently outperform the control condition (LT+ADE) through-

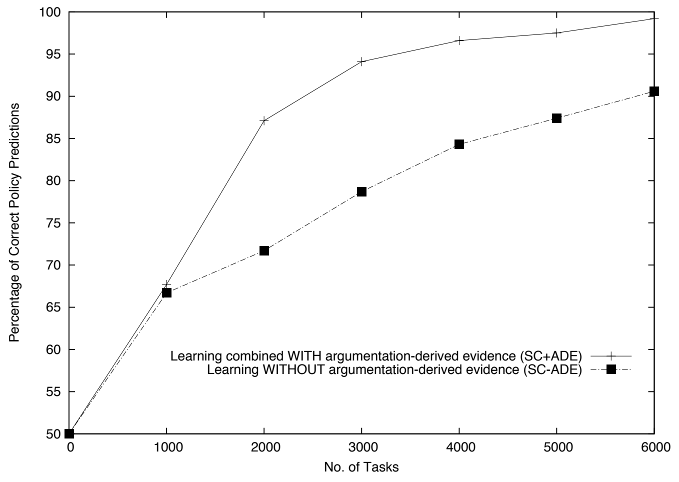


Fig. 7. Graph showing the effectiveness of allowing the exchange of arguments in learning policies.

out the experiment. It is interesting to see that, with relatively small training set, the control condition performed better than the decision tree learner (C4.5+ADE). This is, we believe, because the model built by the decision tree learner overfit the data. The tree was pruned after each set of 1000 tasks and after 3000 tasks the accuracy of the C4.5+ADE model rose to about 83% to tie with the control condition and from then the decision tree learner performed better than the control condition. The performance of the control condition dropped to about 81% after 4000 tasks. After 6000 tasks the accuracy of the decision tree learner had risen above 95% while that of the control condition was just over 84%.

Tests of statistical significance were applied to the results. The standard deviations of the results were analysed and the trend line plotted. (See Figure 9). Using linear regression, the analysis of variance (ANOVA) shows that as the number of tasks increases, each of the three machine learning techniques (with or without argumentation-derived evidence) consistently converges with a 95% confidence interval. Furthermore, for all the pairwise comparisons, the scenarios where argumentation-derived evidence was combined with machine learning techniques consistently yielded higher rates of convergence ($p < 0.02$) than those without additional evidence. Specifically, the decision tree learner enhanced with argumentation-derived evidence (C4.5+ADE) converges ($y = 15.3944 - 0.0022x$) with a F value of 15.66 and significance $p = 0.0167$. The k-NN+ADE converges ($y = 9.7983 - 0.0014x$) with a F value of 38.58 and significance $p = 0.0034$, and the SC+ADE ($y = 8.819 - 0.0013x$)

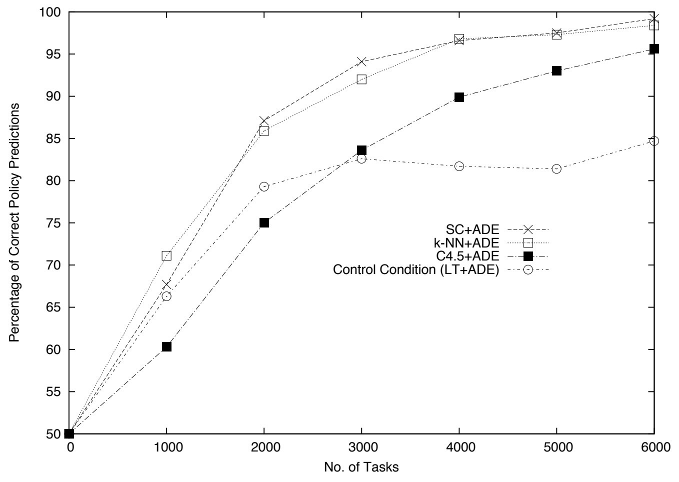


Fig. 8. Graph showing the effectiveness of learning policies with the aid of argumentation-derived evidence using various techniques (LT+ADE, C4.5+ADE, *k*-NN+ADE & SC+ADE).

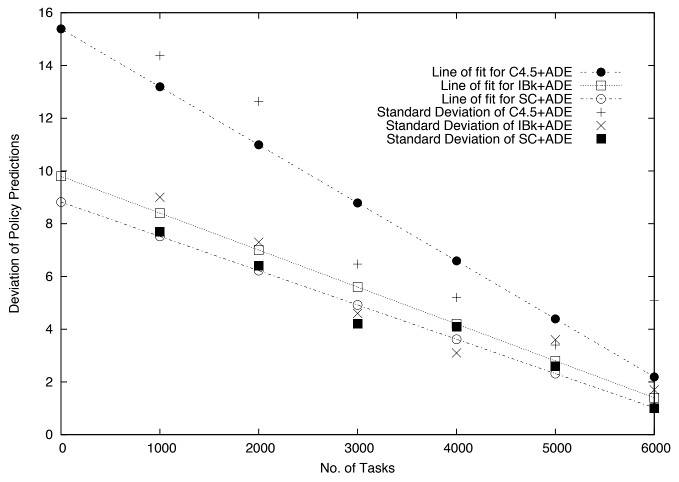


Fig. 9. Graph showing the rate of convergence of the three techniques enhanced with ADE in learning policies (C4.5+ADE, *k*-NN+ADE, & SC+ADE).

converges with a F value of 136.45 and significance $p = 0.0003$. On the other hand, with a significance $p = 0.3957$, there is no statistical significance as to whether LT+ADE converges or not. These results confirm our hypotheses.

6 Discussion and Related Work

The research presented in this paper represents the first model for using evidence derived from argumentation to learn underlying social characteristics (e.g. policies/norms) of others. There is, however, some prior research in combining machine learning and argumentation, and in using argument structures for machine learning. In that research, Možina et al. [9] propose a novel induction-based machine learning mechanism using argumentation. The work implemented an argument-based extension of CN2 rule learning (ABCN2) and showed that ABCN2 out-performed CN2 in most tasks. However, the framework developed in that research will struggle to disambiguate between constraints that may produce similar outcome/effect, which is the main issue we are addressing in our work. Also, the authors assume that the agent knows and has access to the arguments required to improve the prediction accuracy, but we argue that it is not always the case. As a result, we employ information-seeking dialogue to tease out evidence that could be used to improve performance.

In related research, Rovatsos et al. [14] use hierarchical reinforcement learning in modifying symbolic constructs (*interaction frames*) that regulate agent *conversation patterns*, and argue that their approach could improve an agent's conversation strategy. In our work, we used information-seeking dialogue to obtain evidence from the interaction and learned the entire sequence as against a segment (frame) of the interaction [14]. We have demonstrated the effectiveness of using argumentation-derived evidence to learn underlying social characteristics (e.g. policies) without assuming that those underlying features are public knowledge.

In recent research, Sycara et al. [15] investigate agent support for human teams in which software agents aid the decision making of team members during collaborative planning. One area of support that was identified as important in this context is guidance in making policy-compliant decisions. This prior research focuses on giving guidance to humans regarding their own policies. An important and open question, however, is how can agents support human decision makers in developing models of others' policies and using these in guiding the decision maker? Our work is aimed at bridging this gap (a preliminary version was presented in [4]). We employ a novel combination of techniques in identifying, learning and building accurate models of others' policies, with a view to exploiting these in supporting human decision making.

In our future work, we plan to develop strategies for advising human decision makers on how a plan may be resourced and who to talk to on the basis of policy and resource availability constraints learned [10].

Parsons et al. [12] investigated the properties of argumentation-based dialogues and examined how different classes of protocols can have different outcomes. Furthermore, we plan to explore ideas from this work to see which class of protocol will yield the "best" result in this kind of task. We are hoping that some of these ideas will drive the work on developing strategies for choosing who to talk to.

7 Conclusions

In this paper, we have presented a technique that combines machine learning and argumentation for learning policies in a team of collaborating agents engaging in joint activities. We believe, to the best of our knowledge, that this is the first study into learning models of other agents using argumentation-derived evidence. The results of our empirical investigations show that evidence derived from argumentation can have a statistically significant positive impact on identifying, learning and modeling others' policies during collaborative activities. The results also demonstrate that through the use of appropriate machine learning techniques more accurate and stable models of others' policies can be derived more rapidly than with simple memorisation of past experiences. Accurate policy models can inform strategies for advising human decision makers on how a plan may be resourced and who to talk to [15], and may aid in the development of more effective strategies for agents [10]. Our results demonstrate that significant improvements can be achieved by combining machine learning techniques with argumentation-derived evidence. Having shown that accurate models of others' policies could be learned through argumentation-derived evidence, we conjecture that one could, in principle, learn accurate models of other agents' properties (e.g. priorities, preferences, and so on).

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. T. J. M. Bench-Capon, J. B. Freeman, H. Hohmann, and H. Prakken. Computational models, argumentation theories and legal practice. In

C. Reed and T. J. Norman, editors, *Argumentation Machines. New Frontiers in Argument and Computation*, pages 85–120, Dordrecht, The Netherlands, 2003. Kluwer Academic Publishers.

2. J. Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370, 1987.
3. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, 13(1):21–27, 1967.
4. C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons. Learning policies through argumentation-derived evidence (extended abstract). In van der Hoek, Lesprance, Kaminka, Luck, and Sen, editors, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada, 2009. To appear.
5. C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons. Learning policy constraints through dialogue. In *Proc. of the AAAI Fall Symposium on The Uses of Computational Argumentation*, pages 20–26, Virginia, USA, 2009.
6. E. Friedman-Hill. *Jess in Action.* Manning, 2003.
7. A. Kelemen, Y. Liang, and S. Franklin. A comparative study of different machine learning approaches for decision making. In N. E. Mastorakis, editor, *Recent Advances in Simulation, Computational Methods and Soft Computing*, pages 181–186, Piraeus, Greece, 2002. WSEAS Press.
8. T. M. Mitchell. *Machine Learning.* McGraw Hill, 1997.
9. M. Možina, J. Žabkar, and I. Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.
10. N. Oren, T. J. Norman, and A. Preece. Loose lips sink ships: A heuristic for argumentation. In *Proc. of the 3rd Int'l Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, pages 121–134, 2006.
11. S. Parsons and N. R. Jennings. Negotiation through argumentation-A preliminary report. In *Proc. of the 2nd Int'l Conference Multi-Agent Systems (ICMAS'96)*, pages 267–274, Kyoto, Japan, 1996.
12. S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexities of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.
13. J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
14. M. Rovatsos, I. Rahwan, F. Fischer, and G. Weiss. Practical strategic reasoning and adaptation in rational argument-based negotiation. In *Argumentation in Multi-Agent Systems*, volume 4049 of *LNCS*, pages 122–137. Springer-Berlin, 2005.
15. K. Sycara, T. J. Norman, J. A. Giampapa, M. J. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. H. Strub. Agent support for policy-driven collaborative mission planning. *The Computer Journal*, page bxp061, 2009.
16. P. E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4(2):161–186, 1989.
17. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* SUNY Press, Albany, NY, USA, 1995.
18. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.

On a Computational Argumentation Framework for Agent Societies

Stella Heras, Vicente Botti, and Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022, Valencia, Spain
email: sheras@dsic.upv.es

Abstract. In this paper, we analyse the requirements that argumentation frameworks should take into account to be applied in agent societies. Then, we propose a generic framework for the computational representation of argument information. It is able to represent different types of complex arguments in open multi-agent societies, where agents have social relations between them. In addition, we have formalised our framework by defining an argumentation framework based on it.
ACM Categories: Coherence and Coordination, Multi-Agent Systems.
Keywords: Agreement Technologies, Argumentation.

1 Introduction

A recent trend in Multi-Agent Systems (MAS) research is to broaden the applications of the paradigm to open MAS [20], where heterogeneous agents could enter into (or leave) the system, interact, form societies and adapt to changes in the environment. This and other paradigms for computing, such as grid computing or peer-to-peer technologies, have given rise to a new approach of computing as interaction [17]. This notion is used to define large complex systems in terms of the services that their entities or agents can offer and consume and consequently, in terms of the interactions between them. The high dynamism of these systems requires them to have a way of harmonising knowledge inconsistencies and reaching common agreements, for instance, when agents in an open MAS are faced with the goal of collaborating and solving a problem together.

Argumentation is probably the most natural way of harmonising conflicts. It provides a fruitful means of dealing with non-monotonic and defeasible reasoning. During the last decade, this important property has made many Artificial Intelligence (AI) researchers to pay attention on argumentation theory. In addition, research on argumentation is also at its peak in the Multi-Agent Systems (MAS) community, since it has been very successful to implement agents' internal and practical reasoning and to manage multi-agent dialogues [23]. Nowadays, argumentation is an active research area in AI and MAS [4].

However, most argumentation systems consider abstract notions of argument that are not intended for performing automated reasoning over them (automatic argument generation, selection and evaluation). In fact, the proposed computational argumentation frameworks take a narrow view of the argument structure [26]. On the other hand, most MAS whose agents have argumentation capabilities use ad-hoc and domain-dependent representations for arguments [30][31]. Moreover, little work, if any, has been done to study the effect of the social relations between agents in the way that they argue and manage arguments. Commonly, the term *agent society* is used in the argumentation and AI literature as a synonym for an *agent organisation* [12] or a *group of agents* that play specific roles, follow some established interaction patterns and collaborate to reach global objectives [14][19]. Nevertheless, the social context of agent societies (the social dependencies between agents and the effects of their membership to a group in the way that they can argue with other agents), is not analysed.

To our knowledge, no research is done to adapt argumentation frameworks to represent and manage arguments in agent societies taking into account their social context both in the representation of arguments and in the argument management process. Nevertheless, this social information plays an important role in the way agents can argue and learn from argumentation experiences. Depending on their social relations with other agents, an agent can accept arguments from a member of its society that it would never accept before acquiring social dependencies with this member. For instance, in a company a subordinate must sometimes accept arguments from his superior that go against his own ideas and that he would never accept without this power relation. Also, despite having no knowledge about an opponent agent, an agent could try to infer the potential willingness of the opponent to accept an argument by taking into account its social relation with the opponent, or even its social relation with similar agents in the past. These are major considerations that should be studied to apply argumentation techniques in real domains modelled by means of open MAS.

The purpose of this paper is twofold. On one hand, Section 2 analyses the requirements for an argumentation framework for agent societies and proposes a generic computational representation of arguments. This framework stresses the importance of the social dependencies between agents and the effects of their membership to a group in the way that they argue. On the other hand, in Section 3 we formalise this proposal by defining a computational argumentation framework (AF) for the design and implementation of argumentation dialogues in MAS. Our notion of argument relies on technological standards for argument and data interchange on the web. Hence, our argumentation framework can be adapted to work in multiple domains and distributed environments.

2 A Computational Model for Argument Representation in Agent Societies

In this section, we introduce the formal definition of the concepts that define our approach for agent societies. Then, we analyse the issues that have been considered to choose a suitable argumentation framework for agent societies. Taking them into account, we propose a computational representation of arguments. Finally, an example is provided.

2.1 Society Model

In this work, we follow the approach of [10] and [2], who define an *agent society* in terms of a set of *agents* that play a set of *roles*, observe a set of *norms* and a set of *dependency relations* between roles and use a *communication language* to collaborate and reach the global objectives of the *group*. This definition is generic enough to fit most types of agent societies, such as social networks of agents or open agent organisations. Broadly speaking, it can be adapted to any open MAS where there are norms that regulate the behaviour of agents, roles that agents play, a common language that allow agents to interact defining a set of permitted locutions and a formal semantics for each of these elements. Moreover, the set of norms in open MAS define a *normative context* (covering both the set of norms defined by the system itself as well as the norms derived from agents' interactions)[8].

However, we consider that the values that individual agents or groups want to promote or demote and preference orders over them have also a crucial importance in the definition of an argumentation framework for agent societies. These values could explain the reasons that an agent has to give preference to certain beliefs, objectives, actions, etc. Also, dependency relations between roles could imply that an agent must change or violate its value preference order. For instance, an agent of higher hierarchy could impose their values to a subordinate or an agent could have to adopt a certain preference order over values to be accepted in a group. Therefore, we endorse the view of [21], [28] and [3], who stress the importance of the audience in determining whether an argument (e.g. for accepting or rejecting someone else's beliefs, objectives or action proposals) is persuasive or not. Thus, we have included in the above definition of agent society the notion of values and preference orders among them. Next, we provide a formal definition for the model of society that we have adopted:

Definition 1 (Agent Society). *An Agent society S in a MAS is a tuple $S = < A, R, D, G, N, V, Roles, Dependency, Group, val, ValPref_Q >$ where:*

- $A = \{a_1, ..., a_I\}$ *is the set of I agents members of S in a certain time.*
- $R = \{r_1, ..., r_J\}$ *is the set of J roles that have been defined in S.*
- $D = \{d_1, ..., d_K\}$ *is the set of K possible dependency relations in S.*
- $G = \{g_1, ..., g_L\}$ *is the set of groups that the agents of S form, where each $g_l = \{a_1, ..., a_M\}, M \leq I$ consist of a set of agents $a_i \in A$ of S.*
- N *is the normative context of S. That is, the defined set of norms that affect the roles that the agents play in S*[1].
- $V = \{v_1, ..., v_P\}$ *is the set of P values predefined in the S.*
- *$Roles : A \rightarrow 2^R$ is a function that assigns an agent its roles in S.*

[1] Thus, we assume normative MAS, such as the one proposed in [8] for instance.

– *Dependency* $:<_D^S \subseteq RxR$ *defines a reflexive and transitive pre-order relation over roles*[2]*. That is,* $\forall r_1, r_2, r_3 \in R, r_1 <_d^S r_2 <_d^S r_3$ *implies that* r_3 *has the highest rank with respect to the dependency relation* d *in* S*. Also,* $r_1 <_d^S r_2$ *and* $r_2 <_d^S r_1$ *implies that* r_1 *and* r_2 *have the same rank with respect to* d*.*
– $Group : A \rightarrow 2^G$ *is a function that assigns an agent its groups in* S*.*
– $val : A \rightarrow V$ *is a function that assigns an agent the set of values that it has.*
– $ValPref_Q \subseteq VxV$*, where* $Q = A \vee Q = G$*, defines a reflexive and transitive pre-order relation* $<_Q^S$ *over the values. For instance,* $\forall v_1, v_2, v_3 \in V, Valpref_a = v_1 <_a^S v_2 <_a^S v_3^S$ *implies that agent* a *prefers value* v_3 *to* v_2 *and value* v_2 *to value* v_1 *in* S*. Similarly,* $Valpref_g = v_1 <_g^S v_2 <_g^S v_3^S$ *implies that group* g *prefers value* v_3 *to* v_2 *and value* v_2 *to value* v_1 *in* S*.*

Once the concepts that we use to define agent societies are specified, the next section analyses the computational requirements for argument representation in these societies. Then, our approach for agent societies and the analysed requirements are used to propose a new computational representation for arguments.

2.2 Computational Requirements for Arguments in Agent Societies

An argumentation process is conceived as a reasoning model with several steps:

1. Building arguments (supporting or attacking conclusions) from knowledge bases.
2. Defining the strengths of those arguments by comparing them in conflict situations.
3. Evaluating the acceptability of arguments in view of the other arguments that are posed in the dialogue.
4. Defining the justified conclusions of the argumentation process.

The first step to design MAS whose agents are able to perform argumentation processes is to decide how agents represent arguments. According to the *interaction problem* defined in [7], *"...representing knowledge for the purpose of solving some problem is strongly affected by the nature of the problem and the inference strategy to be applied to the problem..."*. Therefore the way in which agents computationally represent arguments should ease the automatic performance of argumentation processes.

Most research effort on the computational representation of arguments is performed in the area of developing models for argument authoring and diagramming [25][27](OVA[3]). However, these systems assume human users interacting with the software tool and are not conceived for performing agents' automatic reasoning processes. Other research works where the computational modelling

[2] For instance, in the society S of the example of Section 2.4, a basin administrator has a power dependency relation over any farmer of its river basin, which has to accept arguments from this administrator that it would never accept without this power relation ($Farmer <_{Power}^S Basin\ Administrator$).
[3] `OVA at ARG:dundee: www.arg.dundee.ac.uk`

of arguments has been studied are those on case-based argumentation. From the first uses of argumentation in AI, arguments and cases are intertwined [29]. Case-based argumentation particularly reported successful applications in American common law [4], whose judicial standard orders that similar cases must be resolved with similar verdicts. In [5] a model of legal reasoning with cases is proposed. But, again, this model assumed human-computer interaction and cases were not thought to be only acceded by software agents. Case-Based Reasoning (CBR) systems [1] allow agents to learn from their experiences. In MAS, the research in case-based argumentation is quite recent with just a few proposals to date. These proposals are highly domain-specific or centralise the argumentation functionality in a *mediator* agent that manages the dialogue between the agents of the system [15].

As pointed out before, we focus on argumentation processes performed among a set of agents that belong to an agent society and must reach an agreement to solve a problem taking into account their social dependencies. Each agent builds its individual position in view of the problem (a solution for it). At this level of abstraction, we assume that this could be a generic problem of any type (e.g. a resource allocation problem, an agreed classification, a joint prediction, etc.) that could be characterised with a set of features. Thus, we assume that each agent has its individual knowledge resources to generate a potential solution. Also, agents have their own argumentation system to create arguments to support their positions and defeat the ones of other agents.

Taking into account the above issues, there are a set of requirements that a suitable framework to represent arguments in agent societies should met:

– be computationally tractable and designed to ease the performance of automatic reasoning processes over it.
– be rich enough to represent general and context dependent knowledge about the domain and social information about the agents' dependency relations or the agents' group.
– be generic enough to represent different types of arguments.
– comply with the technological standards of data and argument interchange on the web.

These requirements suggest that an argumentation framework for agent societies should be easily interpreted by machines and have highly expressive formal semantics to define complex concepts and relations over them. Thus, we propose a Knowledge-Intensive (KI) case-based argumentation framework [9], which allows automatic reasoning with semantic knowledge in addition to the syntactic properties of cases. Reasoning with cases is specially suitable when there is a weak (or even unknown) domain theory, but acquiring examples encountered in practice is easy. Most argumentation systems produce arguments by applying a set of inference rules. In open MAS the domain is highly dynamic and the set of rules that model it is difficult to specify in advance. However, tracking the arguments that agents put forward in argumentation processes could be relatively simple. Other important problem with rule-based systems arises when the knowledge-base of rules must be updated (e.g. adding a new rule). Updates imply to check the knowledge-base for conflicting or redundant rules. Case-based systems are in most cases easier to maintain than rule-based systems and hence, more suitable for being applied in dynamic domains.

In the following section, we present the framework proposed accordingly to the above requirements. This framework is also conceived to allow agents to improve their argumentation skills and be able to evaluate the persuasive power of arguments for specific audiences in view of their previous argumentation experiences.

2.3 Case-Based Model for Argument Representation

In open multi-agent argumentation systems the arguments that an agent generates to support its position can conflict with arguments of other agents and these conflicts are solved by means of argumentation dialogues between them. To allow agents to take the maximum profit from previous argumentation experiences, the structure that agents use to store information related to their argumentation experiences must be able to represent knowledge about individual arguments and also about the argumentation dialogues where arguments were posed. Therefore, agents that implement our argumentation framework have an individual case-based argumentation system with the following knowledge resources:

- Domain-cases: a set of cases that store information about problems that were solved in the past. The structure and concrete feature set of these cases depends on the specific application domain, but at least, they have a minimum set of features that represent the problem and the solution applied to it.
- Argument-cases: a set of cases that store information about arguments that the agent posed in the past and the results that were obtained by putting forward them in a previous argumentation dialogue[4].
- Dialogue graphs: a set of directed graphs that link argument-cases and represent previous argumentation dialogues. Nodes represent arguments and arrows between nodes represent attack relations.
- Ontology of Argumentation Schemes: an ontology that encodes the set of argumentation schemes that agents can use to produce arguments. These schemes are stereotyped patterns of reasoning [32] that can be used to create presumptive arguments from a set of premises that characterise the problem to solve. In addition, argumentation schemes have a set of critical questions, which represent attacks to the conclusion drawn from the scheme.

The argument-cases are the main structure that we use to implement our framework and computationally represent arguments in agent societies. Argument-cases have two main objectives: 1) they can be used by agents as knowledge resource to generate new arguments in view of past argumentation experiences

[4] Note that argument-cases and arguments are not the same, but the former are knowledge structures that store information about previous arguments (and maybe represent a generalisation of several arguments)

PROBLEM	Domain Context	Premises = {Volume, Price, etc.}	
	Social Context	Proponent	ID = F2
			Role = Farmer
			Norms = N_{F2}
			$ValPref_{F2}$ = [EC<SO]
		Opponent	ID = BA
			Role = Basin Administrator
			Norms = N_{BA}
			$ValPref_{BA} = \emptyset$
		Group	ID = RB
			Norms = N_{RB}
			$ValPref_{RB}$ = [SO<EC]
		Dependency Relation = Power	
SOLUTION	Argument Type = Inductive		
	Conclusion = $F2tr$ (F2 wins the water-right transfer)		
	Acceptability State = Acceptable		
	Received Attacks	Critical Questions = $\emptyset$	
		Distinguish Case = $\emptyset$	
		Counter Examples = {C1}	
JUSTIFICATION	Cases = {C2}		
	Schemes = $\emptyset$		
	Associated Dialogue Graph		

Table 1. Structure of an Argument Case

and 2) they can be used to store new argumentation knowledge that agents gain in each dialogue, improving the agents' argumentation skills. Due to space restrictions, we focus here on explaining this knowledge resource. Table 1 shows an example of the structure of a specific argument-case (explained in the example of Section 2.4). As it is usual in CBR systems, the argument-cases have three main parts: the description of the *problem* that the case represents, the *solution* applied to this problem and the *justification* why this particular solution was applied. An argument-case stores the information about a previous argument that an agent posed in certain step of a dialogue with other agents.

Problem: The problem description stores the *premises* of the argument, which represent the context of the domain where the argument was put forward. In addition, if we want to store an argument and use it to generate a persuasive argument in the future, the features that characterise the audience of the previous argument (the social context) must also be kept.

For the definition of the social context of arguments, we follow our model of society presented in Section 2.1. Therefore, we store in the argument-case the social information about the *proponent* of the argument, the *opponent* to which the argument is addressed, the *group* to which both agents belong and the dependency relation established between the roles that these agents play. For the sake of simplicity, in what follows we assume that in each step of the dialogue, one proponent agent generates an argument and sends it to one opponent agent that belongs to its same group. However, either the proponent or the opponent's features could represent information about agents that act as representatives of a group and any agent can belong to different groups at the same time.

Thus, the proponent and opponent's features represent information about the agent that generated the argument and the agent that received it respectively. Concretely, for each agent the argument-case stores a unique *ID* that identifies it in the system and the *role* that the agent was playing when it sent or received the argument (e.g. farmer and basin administrator, do not confuse with the role of proponent and opponent from the argumentation perspective). In addition, a reference to the set of norms that governed the behaviour of the agents at this step of the dialogue is also stored, since the normative context of agents could force or forbid them to accept certain facts and the arguments that support them (e.g. a norm could invalidate a dependency relation or a value preference order). Moreover, if known, we also store the preferences of each agent over the pre-defined set of general values in the system (e.g. security, solidarity, economy, etc.). As pointed out before, these preferences ($ValPref_{F2}$ and $ValPref_{BA}$) affect the persuasive power of the proponent's argument over the opponent's behaviour.

Regarding the group features, the argument-case stores the unique identifier *ID* of the agents' group, the set of *norms* that regulates the behaviour of the group members at this moment, since changes can occur due to norm emergence, and the preference order ($ValPref_{RB}$) about the *social values*[5] of the group. Finally, the dependency relation between the proponent's and the opponent's roles is also stored. To date, we define the possible dependency relations between roles as in [10]:

- *Power*: when an agent has to accept a request from other agent because of some pre-defined domination relationship between them (e.g. in a society S that manages the water of a river basin, $Farmer <^{S}_{Power} BasinAdministrator$, since farmers must comply with the laws announced by the basin administrator).
- *Authorisation*: when an agent has committed itself to other agent for a certain service and a request from the latter leads to an obligation when the conditions are met (e.g. in the society S, $Farmer_i <^{S}_{Authorisation} Farmer_j$, if $Farmer_j$ has contracted a service that offers $Farmer_i$).
- *Charity*: when an agent is willing to answer a request from other agent without being obliged to do so (e.g. in the society S, by default $Farmer_i <^{S}_{Charity} Farmer_j$ and $Farmer_j <^{S}_{Charity} Farmer_i$).

Solution: In the solution part, the *argument type* that defines the method by which the conclusion of the argument was drawn and this *conclusion* itself are stored. By default, we do not assume that agents have a pre-defined set of rules to infer deductive arguments from premises, which is difficult to maintain in

[5] We use the term social values to refer to those values that are agreed by (or commanded to) the members of a society as the common values that this society should promote (e.g. justice and solidarity in an ideal society) or demote.

open MAS. In our framework, agents have the following ways of generating new arguments:

- *Presumptive arguments*: by using the premises that describe the problem to solve and an argumentation scheme whose premises match them.
- *Inductive arguments*: by using similar argument-cases and/or domain-cases stored in the case-bases of the system.
- *Mixed arguments*: by using premises, cases and argumentation schemes.

Moreover, the argument-case stores the information about the *acceptability state* of the argument at the end of the dialogue. This feature shows if the argument was deemed *acceptable*, *unacceptable* or *undecided* in view of the other arguments that were put forward during the dialogue (see Section 3 for details). Regardless of the final acceptability state of the argument, the argument-case also stores the information about the possible *attacks* that the argument received. These attacks could represent the justification for an argument to be deemed unacceptable or else reinforce the persuasive power of an argument that, despite being attacked, was finally accepted. Argument-cases can store different types of attacks, depending on the type of argument that they represent:

- For presumptive arguments: *critical questions* associated with the scheme.
- For inductive arguments [5]: either
  - Premises which value in the context where the argument was posed was different (or non-existent) than the value that it took in the cases used to generate the argument (*distinguish the case*) or
  - Cases which premises also match the premises of the context where the argument was posed, but which conclusion is different than the conclusion of the case(s) used to generate the argument (*counter-examples*).
- For mixed arguments: any of the above attacks.

Justification: The justification part of the argument-case stores the information about the knowledge resources that were used to generate the argument represented by the argument-case (e.g. the set argumentation schemes in presumptive arguments, the set of cases in inductive arguments and both in mixed arguments). In addition, each argument-case has associated a dialogue-graph that represents the dialogue where the argument was posed. This graph can be used later to develop dialogue strategies. The same dialogue graph can be associated with several argument-cases.

Following a CBR methodology, the knowledge resources of the agents' case-based argumentation system allow them to automatically generate, select and evaluate arguments. However, the complete argument management process (how agents generate, select and evaluate arguments by using the knowledge resources of their argumentation systems) is out of the scope of this paper. Also, the framework presented is flexible enough to represent different types of arguments and their associated information, but the value of some features on argument-cases and domain-cases could remain unspecified in specific domains. For instance, in some open MAS, the preferences over values of other agents could not be previously known. However, agents could try to infer the unknown features by using CBR adaptation techniques [16].

2.4 Example

To exemplify our framework, let us propose a simple scenario of an open MAS that represents a water market [6], where agents are users of a river basin that can buy or sell their water-rights to other agents. A water-right is a contract with the basin administration organism that specifies the rights that a user has over the water of the basin (e.g. the maximum volume that he can spend, the price that he must pay for the water or the district where it is settled[6]). In this setting, suppose that two agents that play the role of farmers, F1 and F2, are arguing with a basin administrator, BA, to decide over a water-right transfer agreement that will grant an offered water-right to a farmer. Then, the premises of the domain context would store data about the water-right transfer offer and other domain-dependent data about the current problem. All agents belong to the same group (the river basin RB) whose behaviour is controlled by certain set of norms N_{RB}, its value preference order promotes economy over solidarity (SO<EC) and commands a dependency relation of charity (C) between two farmers and power relation (P) between a basin administrator and a farmer. Also, F1 prefers economy over solidarity (SO<EC) and has a normative context N_{F1}, F2 prefers solidarity over economy (EC<SO) and has a normative context N_{F2} and by default, BA has the value preference order of the basin (SO<EC) and a normative context N_{BA}.

Suppose that F1 has a domain-case C1 that represents a previous water-right transfer agreement that granted a similar water-right to a farmer whose land was adjacent to the district associated with the current water-right offer. Thus, F1 would put forward an argument to BA, generated by using C1.

> A1: I should be the beneficiary of the transfer because my land is adjacent to the owner's land.

Here, we suppose that the closer the lands the cheaper the transfers between them and then, this argument would promote economy. However, F2 has a domain-case C2 that represents a previous water-right transfer agreement that granted a similar water-right to a farmer whose land needed an urgent irrigation to save the crop due to a drought. Thus, F2 would put forward the following argument to BA, generated by using C2.

> A2: I should be the beneficiary of the transfer because there is a drought and my land is almost dry.

In this argument, we assume that crops are lost in dry lands and helping people to avoid losing crops promotes solidarity. In addition, suppose that as basin administrator, BA knows that there is a drought in the basin, which is a new

[6] Following the Spanish Water Law, a water-right is always associated to a district.

premise that should be considered. Also, its ontology of argumentation schemes includes an *Argument for An Exceptional Case* scheme [32] S1 stating that the value preference order of the basin can be waived in case of drought and changed for EC<SO. Therefore, BA could generate an argument by using S1 and certain domain-case C3 that granted a similar water-right transfer to a farmer whose land was dry in a drought to promote solidarity.

A3: There is a drought in the basin and dry lands must be irrigated first.

Table 1 shows the argument-case that F2 could store for A2 at the end of the dialogue, including the attacks received and the knowledge resources that support the argument. The dialogue graph of this argument-case would point to the node that represents it in the whole dialogue (represented with several argument-cases interlinked). Assuming that in our open MAS all agents can receive the arguments posed by the agents of their group, A1 and A2 will attack each other. In addition, A3 will attack A1, which do not takes into account the exceptional case of drought in the basin. Also, assuming that in this society S the administrator BA has a power dependency relation over any farmer ($Farmer <^{S}_{Power}$ *Basin Administrator*), F1 would have to accept the attack that defeats its argument A1 and withdraw it. If the dialogue ends here, the water-right transfer would be granted to F2.

Recall that argument-cases store the social information about roles, values, norms, etc. Therefore, agents can use this information when they are faced with the task of selecting a case from a set of possible cases to support their arguments. For instance, suppose that BA has also found a domain-case C4 that turned down a similar water-right transfer to a farmer whose land was dry in a drought. To decide which C3 or C4 is most suitable to draw a conclusion for the current problem, BA can check its arguments case-base. Then, suppose that BA finds the argument-case that represents the argument that ended the past dialogue that motivated the creation of the domain-case C4 by turning down the transfer. However, the social information about the group does not match with the current one. Thus, BA could infer that in those situation, the farmer was member of a different group where the irrigation of dry lands does not take priority in the case of drought and hence, C4 could not be cited in the current situation.

Up to this point, we have specified our approach for agent societies, analysed the requirements that a suitable argumentation framework for these type of societies should met and proposed our framework. In next section, we formalise this framework.

3 Case-Based Argumentation Framework for Agent Societies

Following our case-based computational representation of arguments, we have designed a formal AF as an instantiation of Dung's AF [11]. The main advantages that this framework contributes over other existent AFs are: 1) the ability to represent social information in arguments; 2) the possibility of automatically managing arguments in agent societies; 3) the improvement of the agents' argumentation skills; and 4) the easy interoperability with other frameworks that follow the argument and data interchange web standards. Next, the elements of the AF (according to Prakken's AF elements [22]) are specified.

3.1 The Notion of Argument: Case-Based Arguments

We have adopted the Argument Interchange Format (AIF) [33] view of arguments as a set of interlinked premiss-illative-conclusion sequences. The notion of argument is determined by our KI case-based framework to represent arguments. In our framework agents can generate arguments from previous cases (domain-cases and argument-cases), from argumentation schemes or from both. However, note that the fact that a proponent agent use one or several knowledge resources to generate an argument does not imply that it has to show all this information to its opponent. The argument-cases of the agents' argumentation systems and the structure of the actual arguments that are interchanged between agents is not the same. Thus, arguments that agents interchange are tuples of the form:

Definition 2 (Argument). $Arg = \{\phi, < S >\}$*, where* ϕ *is the conclusion of the argument and* $< S >$ *is a set of elements that support it.*

This support set can consist of different elements, depending on the argument purpose. On one hand, if the argument provides a potential solution for a problem (e.g. who should be the beneficiary of the transfer), the support set is the set of features (premises) that describe the problem to solve and optionally, any knowledge resource used by the proponent to generate the argument (domain-cases, argument-cases, argumentation schemes or elements of them). On the other hand, if the argument attacks the argument of an opponent, the support set can also include any of the allowed attacks in our framework (critical questions, distinguishing premises or counter-examples). Then, the support set consists of the following tuple of sets of support elements[7]:

Definition 3 (Support Set). $S =< \{Premises\}, \{DomainCases\}, \{ArgumentCases\}, \{ArgumentationSchemes\}, \{CriticalQuestions\}, \{DistinguishingPremises\}, \{CounterExamples\} >$

For instance, assuming that $\sim$ stands for the logical negation and the set of n premises is defined as $Pre = \{pre_1, ..., pre_n\}$, in our example we have that:

$A1 = \{F1tr, < Pre, \{C1\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >\}$
$A2 = \{F2tr, < Pre, \{C2\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset >\}$
$A3 = \{\sim C1, < Pre \cup \{Drought\}, \emptyset, \emptyset, \{S1\}, \emptyset, \{Drought\}, \{C3\} >\}$

where $F1tr$ and $F2tr$ mean that the transfer should be granted to the farmers F1 or F2 respectively and $\sim C1$ means that this case cannot be applied in this context, due to the new distinguishing premise $\{Drought\}$ and the counter-example $C3$.

[7] This representation is only used for illustrative purposes and efficiency considerations about the implementation are obviated.

3.2 The Logical Language

The logical language represents argumentation concepts and possible relations among them. In our framework, these concepts are represented in the form of KI cases and argumentation schemes. Therefore, the logical language of the AF is defined in terms of the vocabulary to represent these resources. In this section, we focus on the definition of the logical language to represent cases. To represent schemes, we use the AIF ontology proposed in [24].

The vocabulary of cases is defined by using an ontology inspired by the approach proposed in [9] and the AIF ontology. We have selected the Ontology Web Language OWL-DL [8] as the formal logics to represent the vocabulary of cases. This variant is based on Description Logics (DL) and guarantees computational completeness and decidability. Thus, it allows for automatic description logic reasoning over argument-cases and domain-cases. In addition, it facilitates the interoperability with other systems. Next, we provide a partial view of the top levels of the ontology[9] for the AF proposed.

In the top level of abstraction, the terminological part of the ontology distinguishes between three disjoint concepts: *Case*, which is the basic structure to store the argumentation knowledge of agents; *CaseComp*, which represent the usual parts that cases have in CBR systems; and *CaseAtt*, which are the specific attributes that make up each component:

$Case \sqsubseteq Thing \quad Case \sqsubseteq \neg CaseComp$
$CaseComp \sqsubseteq Thing \quad CaseComp \sqsubseteq \neg CaseAtt$
$CaseAtt \sqsubseteq Thing \quad CaseAtt \sqsubseteq \neg Case$

As pointed out before, there are two disjoint types of cases:

$ArgumentCase \sqsubseteq Case \quad DomainCase \sqsubseteq Case$
$ArgumentCase \sqsubseteq \neg DomainCase$

Both argument-cases and domain-cases have the three possible types of components that usual cases of CBR systems have: the description of the state of the world when the case was stored (*Problem*); the solution of the case (*Conclusion*); and the explanation of the process that gave rise to this conclusion (*Justification*):

$Problem \sqsubseteq CaseComp \quad Conclusion \sqsubseteq CaseComp$
$Justification \sqsubseteq CaseComp$
$Case \sqsubseteq \forall hasProblem.Problem$
$Case \sqsubseteq \forall hasConclusion.Conclusion$
$Case \sqsubseteq \forall hasJustification.Justification$

Case components are composed of one or more attributes:

$CaseComp \sqsubseteq \geq 1 hasAttribute.CaseAtt$

For instance, the attributes of the solution description of an argument-case are presented below. The cardinality of the possible attacks that an argument-case can receive is not specified, since the case could not have been attacked.

$ArgumentType \sqsubseteq CaseAtt$
$Solution \sqsubseteq = 1 hasArgumentType.ArgumentType$
$Conclusion \sqsubseteq CaseAtt$

[8] `http://www.w3.org/TR/owl-guide/`
[9] The complete specification of the ontology is out of the scope of this paper.

$Solution \sqsubseteq = 1 hasConclusion.Conclusion$
$AcceptabilityState \sqsubseteq CaseAtt$
$Solution \sqsubseteq = 1 hasAcceptabilityState.AcceptabilityState$
$ReceivedAttacks \sqsubseteq CaseAtt$
$CriticalQuestions \sqsubseteq ReceivedAttacks$
$DistinguishingPremises \sqsubseteq ReceivedAttacks$
$CounterExamples \sqsubseteq ReceivedAttacks$

In addition, some additional properties about the concepts of the ontology can also be defined. For instance, instances of argument-cases can have a unique identifier, a creation date or a date for the last time that the case was used, which could be used to determine if a case is outdated and should be removed from the case-base[10]. For simplicity, these elements are not shown in Table 1.

$Case \sqsubseteq = 1 identifier$
$T \sqsubseteq \forall identifier.ID \quad T \sqsubseteq \forall identifier^{-}.Case$
$Case \sqsubseteq = 1 creationDate$
$T \sqsubseteq \forall creationDate.Date \quad T \sqsubseteq \forall creationDate^{-}.Case$
$Case \sqsubseteq = 1 lastUsed$
$T \sqsubseteq \forall lastUsed.Date \quad T \sqsubseteq \forall lastUsed^{-}.Case$

3.3 The Concept of Conflict between arguments

The concept of conflict between arguments defines in which way arguments can attack each other. There are two typical attacks studied in argumentation: *rebut* and *undercut*. In an abstract definition, rebuttals occur when two arguments have contradictory conclusions. Similarly, an argument undercuts other argument if its conclusion is inconsistent with one of the elements of the support set of the latter argument or its associated conclusion. This section shows how our AF instantiates these two attacks. Taking into account the possible elements of the support set, rebut and undercut attacks can be formally defined as follows. Let $Arg_1 = \{\phi_1, < S_1 >\}$ and $Arg_2 = \{\phi_2, < S_2 >\}$ be two different arguments, where $S_1 =< \{Premises\}_1, ..., \{CounterExamples\}_1 >$, $S_2 =< \{Premises\}_2, ..., \{CounterExamples\}_2 >$, $\sim$ stands for the logical negation, $\Rightarrow$ stands for the logical implication and $conc(x)$ is a function that returns the conclusion of the formula x. Then:

Definition 4 (Rebut). *Arg_1 rebuts Arg_2 iff $\phi_1 = \sim\phi_2$ and $\{Premises\}_1 \supseteq \{Premises\}_2$*

That is, if Arg_1 supports a different conclusions for a problem description that includes the problem description of Arg_2. Assuming $F1tr = \sim F2tr$ and vice-versa, in our example, $A1$ and $A2$ rebut each other.

Definition 5 (Undercut). *Arg_1 undercuts Arg_2 if*
$1)\phi_1 = \sim conc(as_k) /$
$\exists cq \in \{CriticalQuestions\}_1 \wedge \exists as_k \in \{ArgumentationSchemes\}_2 \wedge$

[10] In DL, the range of a property C is specified as $T \sqsubseteq \forall R.C$ and its domain as $T \sqsubseteq \forall R^{-}.C$.

$cq \Rightarrow \sim conc(as_k)$, *or*
$2)\phi_1 = dp/$
$(\exists dp \in \{DistinguishingPremises\}_1 \wedge \exists pre_k \in \{Premises\}_2 \wedge dp = \sim pre_k) \vee$
$(dp \notin \{Premises\}_2)$, *or*
$3)\phi_1 = ce/$
$(\exists ce \in \{CounterExamples\}_1 \wedge \exists dc_k \in \{DomainCases\}_2$
$\wedge\, conc(ce) = \sim conc(dc_k)) \vee$
$(\exists ce \in \{CounterExamples\}_1 \wedge$
$\exists ac_k \in \{ArgumentCases\}_2 \wedge conc(ce) = \sim conc(ac_k))$

That is, if the conclusion drawn from Arg_1 makes one of the elements of the support set of Arg_2 or its conclusion non-applicable in the current context of the argumentation dialogue. In our example, $A3$ undercuts $A1$, since its conclusion makes $C1$ non-applicable due to the counter-example $C3$ and the distinguishing premise $\{Drought\}$, which is not considered in the premises that describe the previous problem that is represented by $C1$ and made $F1$ to infer $A1$ from it.

3.4 The Notion of Defeat between arguments

Once possible conflicts between argument have been defined, the next step in the formal specification of an AF is to define the defeat relation between a pair of arguments. This comparison must not be misunderstood as a strategical function to determine with which argument an argumentation dialogue can be won [22]. A function like this must also consider other factors, such as other arguments put forward in the dialogue or agents' profiles. Therefore, it only tells us something about the relation between two arguments. Hence, the relation of defeat between two arguments is defined in our AF as follows. Let $Arg_1 = \{\phi_1, < S_1 >\}$ and $Arg_2 = \{\phi_2, < S_2 >\}$ be two conflicting arguments. Then:

Definition 6 (Defeat). *Arg_1 defeats Arg_2 if Arg_1 rebuts Arg_2 and Arg_2 does not undercut Arg_1, or else Arg_1 undercuts Arg_2*

The fist type of defeat poses a stronger attack on an argument, directly attacking its conclusion. In addition, an argument can strictly defeat other argument.

Definition 7 (Strict Defeat). *Arg_1 strictly defeats Arg_2 if Arg_1 defeats Arg_2 and Arg_2 does not defeat Arg_1*

In our example, $A1$ and $A2$ defeat each other and $A3$ strictly defeats $A1$.

3.5 The Acceptability State of arguments

The acceptability state of arguments determines their status on the basis of their interaction. Only comparing pairs of arguments is not enough to decide if their conclusions are acceptable, since defeating arguments can also be defeated by other arguments. Taking into account the underlying domain theory of a dialectical system, arguments can be considered *acceptable*, *unacceptable* and *undecided* [11]. However, the acquisition of new information in further steps of the dialogue could change the acceptability state of arguments.

Therefore, to decide the acceptability state of arguments a proof theory that takes into account the dialogical nature of the argumentation process is necessary. To evaluate the acceptability of arguments by using a dialogue game is a common approach. Dialogue games are interactions between two or more players, where each one moves by posing statements in accordance with a set or predefined rules [18]. In our AF, the acceptability state of arguments could be decided by using a dialogue game and storing in the argument-case associated to each argument its acceptability state when the dialogue ends. However, the definition of this game is out of the scope of this paper.

4 Discussion

In this paper, we have presented a computational framework to represent arguments in agent societies. This framework takes into account the social dependencies between agents and the effects of their membership to a group in the way that they can argue. However, although the framework is flexible enough to store complex knowledge about arguments and dialogues, the value of some case features could not be specified or known in some domains. For instance, the proponent of an argument obviously knows its own preferences over its set of values, probably knows the preferences of its group but, in a real open MAS, we cannot assume that it also knows the value preferences of its opponent. However, the proponent can know the value preferences of the opponent's group (if both belong to the same) or have some previous knowledge about the value preferences of similar agents playing the same role that the opponent is playing now. The same could happen when agents belong to different groups. Thus, the group features could be unknown, but the proponent could try to use its experience with other agents of the opponent's group and infer these features.

In addition, the argumentation framework was inspired by the standard for argument interchange on the web and hence, an argumentation system based on it can interact with other systems that comply with the standard. Elements of cases are specified by using an ontologic case representation language. This means that agents that implement our case-based framework for argument representation and management could argue with agents with other models of reasoning. Each element of the knowledge structures of the argumentation framework proposed can be translated to a concept of the AIF ontology [25] or an ontology for CBR systems based on [9]. For instance, domain premises can be translated into AIF *Premise Descriptions Forms* and premise values into *Premise I-Nodes*, value preferences can instantiate *Preference-Application-Nodes S-Nodes* and argument types *Presumptive Rule-of-Inference Schemes*. Even temporal propositions, agents, roles and norms can be described with OWL ontologies, as proposed in [13]. Although agents in open MAS are heterogeneous, by sharing these ontologies they can *understand* the arguments interchanged in the system.

Moreover, a formal argumentation framework has been presented. This framework is aimed at providing agents with the ability of having argumentation dialogues with other agents in agent societies, with a weak or unknown domain theory. Moreover, the KI case-based approach used for representing argumentation related information allows agents to apply CBR techniques to learn from the experience and improve their argumentation skills. Current work is focused on the development of the necessary CBR algorithms to generate, select and evaluate arguments from domain-cases, argument-cases and argumentation schemes.

Acknowledgment

This work is supported by the Spanish government grants CONSOLIDER INGENIO 2010 CSD2007-00022, TIN2008-04446 and TIN2009-13839-C03-01 and by the GVA project PROMETEO 2008/051.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, 7, no. 1:39–59, 1994.
2. A. Artikis, M. Sergot, and J. Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic*, 10(1), 2009.
3. T. Bench-Capon and K. Atkinson. *Argumentation in Artificial Intelligence*, chapter Abstract argumentation and values, pages 45–64. 2009.
4. T. Bench-Capon and P. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–938, 2007.
5. T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.
6. V. Botti, A. Garrido, A. Giret, and P. Noriega. Managing water demand as a regulated open mas. In *W. on Coordination, Organization, Institutions and Norms in agent systems in on-line communities, COIN-09*, volume 494, pages 1–10, 2009.
7. T. C. Bylander and B. Chandrasekaran. Generic tasks in knowledge-based reasoning: The right level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies*, 26(2):231–243, 1987.
8. N. Criado, E. Argente, and V. Botti. A Normative Model For Open Agent Organizations. In *International Conference on Artificial Intelligence, ICAI-09*, 2009.
9. B. Diaz-Agudo and P. A. Gonzalez-Calero. *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, chapter An Ontological Approach to Develop Knowledge Intensive CBR Systems, pages 173–214. 2007.
10. V. Dignum. *PhD Dissertation: A model for organizational interaction: based on agents, founded in logic*. PhD thesis, 2003.
11. P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming, and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
12. J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: an organizational view of multi-agent systems. volume 2935, pages 214–230, 2003.
13. N. Fornara and M. Colombetti. Ontology and Time Evolution of Obligations and Prohibitions using Semantic Web Technology. In *Workshop on Declarative Agent Languages and Technologies, DALT-09*, 2009.

14. D. Gaertner, J. A. Rodriguez, and F. Toni. Agreeing on institutional goals for multi-agent societies. In *5th International Workshop on Coordination, Organizations, Institutions, and Norms in agent systems, COIN-08*, 2008.
15. S. Heras, V. Botti, and V. Julian. Challenges for a CBR framework for argumentation in open mas. *Knowledge Engineering Review*, 24(4):327–352, 2009.
16. R. López de Mántaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M.-L. Maher, M. Cox, K. Forbus, M. Keane, and I. Watson. Retrieval, Reuse, Revision, and Retention in CBR. *The Knowledge Engineering Review*, 20(3):215–240, 2006.
17. M. Luck and P. McBurney. Computing as interaction: agent and agreement technologies. In *IEEE Int. Conference on Distributed Human-Machine Systems*, 2008.
18. P. McBurney and S. Parsons. Dialogue games in multi-agent systems. *Informal Logic. Special Issue on Applications of Argumentation in Computer Science*, 22(3):257–274, 2002.
19. E. Oliva, P. McBurney, and A. Omicini. Co-argumentation artifact for agent societies. In *5th International Workshop on Argumentation in Multi-Agent Systems, ArgMAS-08*, 2008.
20. S. Ossowski, V. Julian, J. Bajo, H. Billhardt, V. Botti, and J. M. Corchado. Open issues in open mas: An abstract architecture proposal. In *Conf. de la Asociacion Española de Inteligencia Artificial, CAEPIA-07*, volume 2, pages 151–160, 2007.
21. C. Perelman and L. Olbrechts-Tyteca. *The New Rhetoric: A Treatise on Argumentation*. 1969.
22. H. Prakken and G. Sartor. A dialectical model of assesing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.
23. I. Rahwan. Argumentation in multi-agent systems. *Autonomous Agents and Multiagent Systems, Guest Editorial*, 11(2):115–125, 2006.
24. I. Rahwan and B. Banihashemi. Arguments in OWL: a progress report. In *2nd Int. Conf. on Computational Models of Argument, COMMA-08*, pages 297–310, 2008.
25. I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171(10-15):897–921, 2007.
26. C. Reed and F. Grasso. Recent advances in computational models of natural argument. *International Journal of Intelligent Systems*, 22:1–15, 2007.
27. G. Rowe and C. Reed. Diagramming the argument interchange format. In *Conference on Computational Models of Argument, COMMA-08*, pages 348–359, 2008.
28. J. Searle. *Rationality in Action*. 2001.
29. D. Skalak and E. Rissland. Arguments and cases: An inevitable intertwining. *Artificial Intelligence and Law*, 1(1):3–44, 1992.
30. L.-K. Soh and C. Tsatsoulis. A real-time negotiation model and a multi-agent sensor network implementation. *Autonomous Agents and Multi-Agent Systems*, 11(3):215–271, 2005.
31. P. Tolchinsky, U. Cortés, S. Modgil, F. Caballero, and A. López-Navidad. Increasing human-organ transplant availability: Argumentation-based agent deliberation. *IEEE Intelligent Systems*, 21(6):30–37, 2006.
32. D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. 2008.
33. S. Willmott, G. Vreeswijk, C. Chesñevar, M. South, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, and G. Simari. Towards an argument interchange format for Multi-Agent Systems. In *3rd International Workshop on Argumentation in Multi-Agent Systems, ArgMAS-06*, pages 17–34, 2006.

A Formal Argumentation Framework for Deliberation Dialogues

Eric M. Kok, John-Jules Ch. Meyer, Henry Prakken, and Gerard A. W. Vreeswijk

Department of Information and Computing Sciences,
Utrecht University,
The Netherlands

Abstract. Agents engage in deliberation dialogues to collectively decide on a course of action. To solve conflicts of opinion that arise, they can question claims and supply arguments. Existing models fail to capture the interplay between the provided arguments as well as successively selecting a winner from the proposals. This paper introduces a general framework for agent deliberation dialogues that uses an explicit reply structure to produce coherent dialogues, guides in outcome selection and provide pointers for agent strategies.

Category I.2.11 [**Artificial Intelligence**] Distributed Artificial Intelligence—Languages and structures, multi-agent systems
General Terms Design, Languages
Keywords Argumentation, multi-agent communication, deliberation dialogues

1 Introduction

In multi-agent systems the agents need to work together in order to achieve their personal and mutual goals. Working together means communication and often these dialogues will be on finding consensus over some belief, action or goal. In the last decade frameworks and protocols for such dialogues have been designed using argumentation theory. Walton and Krabbe [14] give a classification of dialogues types based on their initial situation, main goals and participant aims. In a *persuasion dialogue* agents need to find resolution for some conflicting point of view. They will try to persuade the others by forwarding arguments. In *negotiation*, there is not a conflict on some claim, but rather a potential conflict on the division of resources. A deal needs to be made in which each agent tries to get their most preferred resource allocation. *Deliberation* dialogues in contrast, have a significant cooperative aspect. There is a need for action and the agents need to mutually reach a decision. Although agreement is pursued, individual interests also play part.

The literature on argumentation in multi-agent systems has mainly focused on persuasion and negotiation type dialogues. Few systems for deliberation have so far been proposed. The most sophisticated work is that of McBurney et al.

[5] To accommodate deliberating agents, a language and protocol are given that allow for forwarding and discussing of proposals for action. The protocol that they use is liberal in the sense that very few restrictions are imposed on the agents. The modelling of conflicts on beliefs and interests of the agents is limited to the assessment of commitments. It is stated that a voting phase can be used to derive a winner.

It seems that the inquisitive nature of the deliberation process has been well captured in the existing literature. However, the conflicts that arise are left more or less indeterminate. In persuasion, on the other hand, dealing with conflicts is explicitly modelled. Frameworks for these dialogues allow to determine whether given arguments are justified and consequently point out a winner. Such conflicts can be modelled this way for deliberation as well. This can be used to control the deliberation process by maintaining focus on the topic and support the selection of a winning proposal.

For persuasion dialogues, Prakken [10] has proposed a framework that uses an explicit reply structure to capture the relation between arguments. This in turn is used to ensure coherent dialogues as well as to determine the dialogical status of the initial claim. Our framework will be based on this work, adjusting it for use with deliberation dialogues. This will give several advantages. First, proposals can be assigned a status, which can be used to ensure coherent dialogues. Second, the proposed actions can be classified to guide in the selection of a winner. Moreover, the framework will be general to allow for domain specific instantiations and to capture existing protocols in it.

2 The Deliberation Dialogue

A deliberation dialogue commences when the need for action arises. In other words, it needs to be decided upon what action should be taken. A group of people may need to decide where to go for dinner or some automotive company needs to plan what type of car to develop. Agents will need to conceive novel proposals for action and move them in the dialogue. These proposed actions can then be reasoned upon by the agents. If a proposal is unfavourable to the agent it can question it, while it can support the proposal if it seems advantageous. Agents can even express preferences on the proposals. All this is done to influence the dialogue outcome.

In a multi-agent system, deliberation dialogues are only a part of the full communication system. Other types of dialogue, such as argument-based mutual planning [12] or persuasion, can also be part of the system. Deliberation dialogues are thus part of a context. In particular, it commences when in the context the agents belief they mutually need to decide on some action to realize a common goal. Both the goal and need for action can originate from various sources in the context, such as an authority or an earlier dialogue. When the deliberation dialogue starts, agents have, at least in our framework, already agreed on them and can start generating and evaluating proposals.

Agents will have different personal interests and beliefs, because of which conflicts of opinion will come to light during the dialogue. These conflicts can be solved by embedding persuasion-style dialogues. Agents move arguments and question claims to convince other agents. A decision on the winning proposal may be reached through agreement, a voting system or through some authority. Depending on the domain however, both the supplied arguments and the expressed preferences can still be used.

While persuasion is always competitive, deliberation is partially a cooperative process as well. This is expressed in a mutual goal that every agent needs to respect once they accept to engage in deliberation. Support for their proposals needs to show how the action will achieve this common goal. Agents thus need to mediate between their personal opinions and the mutual objective.

As an example, consider a dialogue between three agents that need to find a place for dinner where they will all enjoy the food. They all have an incentive to work towards an agreement on the restaurant, but as the dialogue progresses, differences on beliefs will also need to be resolved.

- a_1: We should go to the local pizzeria.
- a_2: Why should we go there? I propose we go to the nearby bistro.
- a_1: Well, the pizzeria serves tasty pizza's. Why should we go to the bistro?
- a_2: The toppings at the pizzeria are very dull, while the bistro has the best steaks in town.
- a_3: I agree on going to the bistro, because the seafood there is great.
- a_1: The bistro doesn't even server steaks any more.
- a_3: What makes you think the pizza toppings are so dull?
- a_2: Because the menu hasn't been changed for a very long time. We could also just go to pub.
- a_1: No, I don't want to go there.

3 A Formal Deliberation Framework

As explained, our framework will build on the argumentation framework for persuasion dialogues of Prakken, altering and extending it for use with deliberation dialogues. It models persuasion as a dialogue game in which agents make utterances in a communication language while being restricted by a protocol. The utterances, or moves, are targeted at earlier moves. Every reply is either an attacker of surrender, forming an explicit dialogue reply structure. The moves contain claims and arguments in the topic language with an argumentation logic. Since it is a framework it allows for various instantiations of the languages and protocol. In the most basic form the protocol is very liberal, only disallowing agents to speak at the same time and requiring that moves are replies to earlier moves. The dialogue terminates when one of the agents cannot make a legal move. The protocol is defined such that there are no legal moves when there is agreement on the original claim.

The explicit reply structure is utilized in two ways. First, moves have a dialectic status. The idea is that a dialogue move is *in* if it is surrendered or else all its attackers are *out*, and that it is *out* if it has an attacker that is *in*. Now the outcome of the persuasion dialogue can be determined based on the dialogical status of the original claim, viz. if at termination this claim is *in* the proponent is the winner. Second, the protocol may be extended with a relevance rule. This compels the agents to stay focussed on the dialogue topic, giving rise to more coherent dialogues.

To make the framework suitable for deliberation dialogues, several modifications are needed. First, multiple agents need to be supported, while the persuasion framework only covers one proponent and one opponent. Several notions, such as relevance, and protocol rules, such as for termination, need to be revised accordingly. Second, there are multiple proposals instead of a single claim to discuss. The communication language needs support for forwarding, rejecting and questioning them. Multiple proposals also means there are multiple dialogical trees to which the agents may contribute. Third, the dialogue outcome is no longer a direct result of the moves. A winning function is needed to select a single action from all actions that are proposed, or possible none if there is no acceptable option.

Now the formal specification for deliberation systems in our framework is introduced. This definition is taken from [10], with the appropriate additions and revisions.

Definition 1 (Deliberation system). A dialogue system for deliberation dialogues is defined by:

- A *topic language* L_t is a logical language closed under classical negation.
- An *argumentation logic* $\mathcal{L}$ as defined in [11]. It is an instance of the Dung [4] argumentation model in which arguments can be formed using inference trees of strict and defeasible rules. Here, an argument will be written as $A \Rightarrow p$ where A is a set of premises and sub-arguments, $\Rightarrow$ is the top inference rule and p is the conclusion of the argument. Such an argument can be attacked by rebutting the conclusion or a sub-argument, by undermining some premise it uses or by undercutting one of the used inference rules.
- A *communication language* L_c, which is a set of locutions $\mathcal{S}$ and two binary relations R_a and R_s of attacking and surrendering reply on $\mathcal{S}$. Every $s \in \mathcal{S}$ is of the form $p(l)$ where p is a performative and $l \in L_t$, $l \subseteq L_t$ or l is an argument in $\mathcal{L}$. R_a and R_s are disjunct and irreflexive. Locutions cannot attack one locution and surrender to another. Finally, every surrendering locution has an *attacking counterpart*, which is an attacking locution in L_c.
- The set $\mathcal{A}$ of agents.
- The set of *moves* M defined as $\mathbb{N} \times \mathcal{A} \times L_c \times \mathbb{N}$ where each element of a move m respectively is denoted by:
  - `id`(m), the move identifier,
  - `player`(m), the agent that played the move,
  - `content`(m), the speech act, or content, of the move,
  - `target`(m), the move target.

- The set of *dialogues* $M^{\leq\infty}$ is the set of all sequences $m_1, \ldots, m_i, \ldots$ from M, where each i^{th} element in the sequence has identifier i and for each m_i in the sequence it holds if $\texttt{target}(m_i) \neq 0$ then $\texttt{target}(m_i) = j$ for some m_j preceding m_i in d. The set of finite dialogues $M^{<\infty}$ is the set of all those dialogues that are finite, where one such dialogue is denoted by d.
- A *dialogue purpose* to reach a decision on a single *course of action*, which is a $P \in L_t$. P is a proposition stating that some action should be done.
- A deliberation *context* consisting of the *mutual goal* $g_d \in L_t$.
- A *protocol* $\mathcal{P}$ that specifies the legal moves at each point in the dialogue. Formally a protocol on M is a function that works on a non-empty set of *legal finite dialogues* $D \subseteq M^{<\infty}$ and the mutual goal such that $\mathcal{P} : D \times L_t \longrightarrow Pow(M)$. The elements of $\mathcal{P}(d)$ are called the legal moves after d. $\mathcal{P}$ must satisfy the condition that for all legal finite dialogue d and moves m it holds that $d \in D$ and $m \in \mathcal{P}(d)$ iff $d, m \in D$.
- A *turntaking function* $\mathcal{T} : D \longrightarrow \mathcal{A}$ mapping a legal finite deliberation dialogue to a single agent.
- A *deliberation outcome* specified by a function $\mathcal{O} : D \times L_t \longrightarrow L_t$, mapping all legal finite dialogues and the mutual goal g_d to a single course of action α.

This deliberation system specification gives rise to a dialogue game with an explicit reply structure. The types of locutions of L_c that are available to the agents are enumerated in Table 1, each with the appropriate attacking and surrendering replies. The attacking counterpart for each surrendering locution is displayed in the same row. The locutions that deal with proposals (propose, reject, why-propose and prefer) are taken from McBurney et al. while the ones dealing with persuasion (argue, why, retract, concede) are adopted from Prakken's framework. Below the term *proposal move* is used when the $\texttt{content}(m) = \textit{propose}(P)$, *argue move* is used when the $\texttt{content}(m) = \textit{argue}(A \Rightarrow p)$, etc.

Argue moves have a well-formed argument in $\mathcal{L}$ as content. If it attacks some other argue move it should defeat the argument contained in that targeted move following the defeat relation of $\mathcal{L}$. All other speech acts have some well-formed formula in L_t as content. Note that for every move m where $\texttt{content} = \textit{propose}$, *prefer* or *prefer-equal* it holds that $\texttt{target}(m) = 0$ and for all other locutions $\texttt{target} \neq 0$. Specific instantiations of our framework may use a different communication language with different speech acts, as long as the reply relation is defined.

Series of moves that agents make are called turns.

Definition 2 (Turn). A *turn* T in a deliberation dialogue is a maximal sequence of moves $\langle m_i, \ldots, m_j \rangle$ where the same player is to move. A complete deliberation dialogue d can be split up in the sequence of turns $\langle T_1, \ldots, T_k, \ldots, T_n \rangle$ where $k \in \mathbb{N}$ is the turn identifier. A turn thus only has moves from a single player, defined by $\texttt{player}(T)$.

A deliberation dialogue may be represented a set of ordered directed trees.

Table 1. The available speech acts in the communication language L_c

speech act	**attacks**	**surrenders**
$propose(P)$	$why\text{-}propose(P)$ $reject(P)$	
$reject(P)$	$why\text{-}reject(P)$	
$why\text{-}propose(P)$	$argue(A \Rightarrow p)$	$drop\text{-}propose(P)$
$why\text{-}reject(P)$	$argue(A \Rightarrow \neg p)$	$drop\text{-}reject(P)$
$drop\text{-}propose(P)$		
$drop\text{-}reject(P)$		
$prefer(P, Q)$		
$prefer\text{-}equal(P, Q)$		
$skip$		
$argue(A \Rightarrow p)$	$argue(B \Rightarrow q)$ where $B \Rightarrow q$ defeats $A \Rightarrow p$ $why(q)$ where $q \in A$	$concede(p)$ $concede(q)$ where $q \in A$
$why(p)$	$argue(A \Rightarrow p)$	$retract(p)$
$concede(p)$		
$retract(p)$		

Definition 3 (Proposal tree). For each proposal move m_i in dialogue d a *proposal tree* P is defined as follows:

1. The root of P is m_i.
2. For each move m_j that is a node in P, its children are all moves m_k in d such that $\mathtt{target}(m_k) = m_j$.

This is a tree since every move in d has a single target. Now, for any move m in proposal tree P we write $\mathtt{proposal}(m) = m_i$.

An example proposal tree is displayed in Fig. 1, which represents a dialogue between three agents. A proposal is moved, questioned and being supported with an argument that in turn had several replies. For each move m_i the number i is its identifier in the dialogue and between brackets the playing agent is noted. Moves in a dotted box are *out*, those in a solid box are *in*.

4 Dialogical Status of a Move

At every point in time, the dialogical status of a move can be evaluated. The use for this is twofold. First, it helps making dialogues coherent through the notion of move relevance. Secondly, the status of proposal moves can later be used during the selection of the final dialogue outcome.

Every move in a proposal tree is always either *in* or *out*. The distinction between attacking and surrendering replies is used here to make the status of moves concrete.

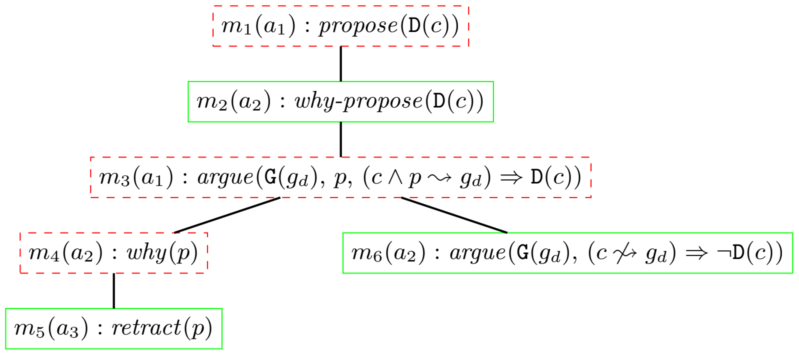

$m_1(a_1)$: $propose(\mathtt{D}(c))$
$m_2(a_2)$: $why\text{-}propose(\mathtt{D}(c))$
$m_3(a_1)$: $argue(\mathtt{G}(g_d),\ p,\ (c \wedge p \rightsquigarrow g_d) \Rightarrow \mathtt{D}(c))$
$m_4(a_2)$: $why(p)$
$m_6(a_2)$: $argue(\mathtt{G}(g_d),\ (c \not\rightsquigarrow g_d) \Rightarrow \neg\mathtt{D}(c))$
$m_5(a_3)$: $retract(p)$

Fig. 1. A small example proposal tree

Definition 4 (Move status). A move m in a dialogue d is *in*, also called *warranted*, iff:

1. m is surrendered in d by every agent $a \in \mathcal{A}$; or else,
2. m has no attacking replies in d that are *in*.

Otherwise it is *out*.

Although this definition is directly taken from [10], special attention here is required to the surrendering attacks. A move is not yet *out* until it is surrendered by every agent in the dialogue, not only by the agent that originally made the attacked move. Take for example the dialogue of Fig. 1. Although agent a_3 moved a $retract(p)$ in response to a_2's $why(p)$ this targeted move was still *in*. It is not until agent a_1 replied with a $retract(p)$ as well that the $why(p)$ move is *in* again. A surrendering move is more a statement of no commitment. This idea is made concrete in the following definition of a surrendered move.

Definition 5 (Surrendered move). A move m is *surrendered* in a dialogue d by some agent a iff:

1. m is an argue move $A \Rightarrow p$ and a has made a reply m' to m that has $\mathtt{content}(m') = concede(p)$; or else
2. a has made a surrendering reply to m in d.

Otherwise it is *out*.

The notion of relevance can now be formalised.

Definition 6 (Relevance). An attacking move m in a dialogue d is *relevant* iff it changes the move status of $\mathtt{proposal}(m)$. A surrendering move is relevant iff its attacking counterpart is.

Depending on the domain a different notion of surrendered move or relevance may be useful. Prakken describes a notion of weak relevance that may be adopted. It is weaker in the sense that an agent can contribute multiple ways to change the proposal tree root and still be relevant. This is achieved by only requiring a move to create an additional way to change the status of a proposal. A protocol with weak relevance allows an agent to make multiple attacks per turn in a proposal tree as opposed to a single one if the earlier notion is used, which we below use the term *strong relevance* for.

Definition 7 (Weak relevance). An attacking move m in a dialogue d is *weakly relevant* iff it creates a new or removes an existing *winning part* in the proposal tree P associated with $\texttt{proposal}(m)$ in d. A surrendering move is weakly relevant iff its attacking counterpart is. If the $\texttt{proposal}(m)$ is *in*, a winning part w^P for this tree P is defined as follows:

1. First include the root of P;
2. For each m of even depth, if m is surrendered by every agent $a \in \mathcal{A}$, include all its surrendering replies, otherwise include all its attacking replies;
3. For each m of even depth, include one attacking reply m' that is *in* in d;

The idea of a winning part is that it is 'a reason' why the proposal is *in* at that moment. Since this is not unique, there may be alternative attacking replies, a move is already weakly relevant if it succeeds to create an additional winning part or removes a winning part. Take for example the dialogue of Fig. 1 again. After $argue(\texttt{G}(g_d), (c \leadsto g_d) \Rightarrow \texttt{D}(c))$ was moved by agent a_1 there are no more strongly relevant moves in this proposal tree, while there exists new weakly relevant moves, for example $argue(s \Rightarrow g_d)$. This results in a more liberal deliberation process.

5 Turntaking and Termination

We have still not made concrete how agents take turns and when the dialogue terminates.

Definition 8 (Turntaking). Agents take turns in sequence and end their turns explicitly with a skip move. Formally, for a dialogue $d = \langle m_1, \ldots, m_n \rangle$ $\mathcal{T}(d) = \texttt{player}(m_n)$ unless $\texttt{content}(m_n) = skip$ in which case $\mathcal{T}(d) = \texttt{player}(m_n) + 1$.

Clearly, when there are no more legal moves besides the skip move, that is $\mathcal{P}(d) = \{skip\}$, the turn switches. Now, the dialogue terminates if all agents no longer make other moves than directly skipping.

Definition 9 (Termination). A dialogue d terminates on $|\mathcal{A}| + 1$ consecutive skip moves.

The rationale behind the termination rule is that each agent should have the opportunity to make new moves when it stills want to. However, to prevent agents from endlessly skipping until some other agent makes a beneficial move or even a mistake, the number of skip moves is limited.

6 Protocol Rules

Now various protocol rules are discussed. Depending on the domain some might or might not be desirable. First, some rules that prevent agents from playing incoherent moves are added. More precisely, these rules require the agents to be relevant, not to overflow the dialogue.

1. Agents can only reply to moves of others. Formally, for every attacking or surrendering move m in a dialogue $\mathit{player}(m) \neq \mathit{player}(\texttt{target}(m))$.
2. Every attacking and surrendering move must be relevant.
3. A turn can contain at most one proposal move.
4. A proposal must be unique in the dialogue. Formally, for every proposal move m in d it holds that $\texttt{content}(m) \notin \{p|p = \texttt{content}(n)$ of some proposal move $n \in d\}$.

The first rule may be dropped for domains where a more liberal deliberation process is appropriate. This would allow agents to attack their own proposals as well. The relevance of the second rule may be strong or weak relevance. Note that in case of strong relevance there can be at most one attacking move per proposal tree.

Not only the dialogue should be coherent. The same holds for the agents' preference statements on the proposals. A protocol rule is added to ensure that an agent is consistent in his ordering.

5. An agent may only make a prefer move if the resulting option ordering maintains transitivity and antisymmetry. This is further explained below.

The last rules are used to ensure that arguments for (and against) a proposal explain how it (fails to) achieve the mutual goal.

6. Every argue move m with $\texttt{target}(m) = m'$ and $\texttt{content}(m') = \textit{why-propose}(\texttt{D}(P))$ will contain an argument in $\mathcal{L}$ with g_d as one of its premises and $\texttt{D}(P)$ as conclusion.
7. Every argue move m with $\texttt{target}(m) = m'$ and $\texttt{content}(m') = \textit{why-reject}(\texttt{D}(P))$ will contain an argument in $\mathcal{L}$ with $\neg g_d$ as one of its premises and $\neg\texttt{D}(P)$ as conclusion.

The arguments that these protocol rules require are used to make sure that a proposal for action P will indeed (fail to) achieve the mutual goal g_d. Put differently, the proposed action needs to be *appropriate* in relation to our dialogue topic. The topic language and used logic therefore need support to express this. One option, used below, is to include an inference rule for the practical syllogism in our logic $\mathcal{L}$. Similar to [2] a practical reasoning rule will then be used that says 'if g_d is a goal and P will achieve g_d then P is an appropriate proposal for action'. Such arguments, below written as $\texttt{G}(g_d), P \rightsquigarrow g_d \Rightarrow \texttt{D}(P)$, can then be moved.

7 Dialogue Outcome

At any moment in time the outcome of the deliberation dialogue can be determined. As the outcome function dictates, this is a single course of action, or no action at all when there is a structural disagreement. To establish this, the options, which are the moved proposals, are first specified and then classified based on their status. This set of proposals is then considered over the agent preferences to determine a winner.

Definition 10 (Options). The dialogue *options* are defined by a function $O : D \longrightarrow Pow(L_t)$ mapping all legal dialogues to a subset of proposals. For any dialogue d the set of options is $O(d) = \{o | o = \texttt{content}(m)$ for each proposal move $m \in d\}$ (below written simply as O). In reverse, $\texttt{move}(o)$ is used to refer to the move in which the option o was proposed.

The proposal moves that introduced the various options have a move status, which will be used to classify the options. Such a classification is any-time and can thus not only be used in selecting the dialogue outcome, but also during the dialogue by agent strategies.

Definition 11 (Option status). An option $o \in O(d)$ for any dialogue d is:

- *justifiable* iff *move*(o) is *in*,
- *invalid* iff $\texttt{player}(\texttt{move}(o))$ played a move m such that $\texttt{target}(m) = \texttt{move}(o)$ and $\texttt{content}(m) = \textit{drop-propose}(o)$,
- otherwise it is *defensible*.

Justifiable options are proposals that were questioned but were successfully defended. None of the agents was able to build a warranted case against the proposal. Defensible options are proposals that were attacked by some move that is still warranted. These are thus options that might be reasonable alternatives albeit not being properly supported. Invalid options are those that were retracted by the proposing agent. From the perspective of the multi-agent system, the status of each option hints at its acceptability as dialogue outcome. To settle on one of the options they are first ordered according to some preference.

Definition 12 (Option preference). An *option preference* relation $\preceq$ is a partial order of O. This is defined as $o_i \prec o_j$ (strictly preferred) if $o_i \preceq o_j$ but $o_j \not\preceq o_i$ and we have $o_i \approx o_j$ (equally preferred) if $o_i \preceq o_j$ and $o_j \preceq o_i$.

A preliminary ordering on the options can be made. This captures the idea of preferring justifiable options over non-justifiable ones. This may be used during the selection of a dialogue outcome.

Definition 13 (Preliminary ordering). Using the set of all options a partition $O = O_j \cup O_i \cup O_d$ is created such that

- $O_j = \{o | o \in O$ where o is justifiable $\}$,
- $O_d = \{o | o \in O$ where o is defensible $\}$,

– $O_i = \{o | o \in O$ where o is invalid $\}$.

Now $\preceq_p$ is the total *preliminary ordering* over O such that:

– for every two options $o_k, o_l \in O_j$, O_d or O_i it holds that $o_k \approx_p o_l$,
– for every $o_j \in O_j$ and $o_d \in O_d$ it holds that $o_j \prec_p o_d$,
– for every $o_d \in O_d$ and $o_i \in O_i$ it holds that $o_d \prec_p o_i$.

Justifiable proposals are in principle preferred as dialogue outcome over defensible proposals, which in turn are preferred over invalid ones. However, justifiable options should not always be selected as winner over defensible ones. For one, the preferences as moved by the agents using prefer and prefer-equal moves may be taken into account.

Definition 14 (Agent option ordering). Every agent a has a partial *agent option ordering* $\preceq_a$ over O such that for any two options $o_i, o_j \in O$:

– $o_i \prec_a o_j$ if the agent played some move m where $\texttt{content}(m) = \textit{prefer}(o_j, o_i)$,
– $o_i \approx_a o_j$ if the agent played some move m where $\texttt{content}(m) = \textit{prefer-equal}(o_j, o_i)$.

The protocol forces an agent to be consistent in its preference utterances with relation to the strict ordering of options.

When the dialogue terminates, the deliberation dialogue outcome should be selected from the set of options. How this final selection is achieved is totally dependent on the domain and the purpose of the system. For example, there may be an agent authority that gets to choose the winner, an additional phase may be introduced in which agents vote on the outcome or a function may be used to aggregate all (preliminary and agent-specific) preference orderings. In any case we need to leave open the option for *mutual disagreement* [5].

Preference aggregation is extensively studied in the field of social choice theory and is out of the scope of the present paper. [9] It is interesting to note, though, that when maximum social welfare is desirable it may be good to incorporate the notion of our option status in the winner selection. The valuable information obtained during the deliberation dialogue can be used with a public calculus. This would decide on the outcome in a way similar to the use of public semantics and would not need to rely on agents considering these notions in their voting strategies. For single agents, this is already studied in [1]. How to make use of this is left as future research.

8 An Example

To further explain how the different notions work together, consider an example of three agents $\mathcal{A} = \{a_1, a_2, a_3\}$ participating in a deliberation dialogue with mutual goal g_d. We will use all the protocol rules discussed above and adopt a weak form of move relevance. The turns are as follows:

- T_1 by a_1
  m_1 : $\mathit{propose}(\mathtt{D}(z))$ where $z = \mathit{goToPizzeria}$
- T_2 by a_2
  m_2 : $\mathit{why\text{-}propose}(\mathtt{D}(b))$ where $b = \mathit{goToBistro}$
  m_3 : $\mathit{propose}(\mathtt{D}(z))$
- T_3 by a_3
  m_4 : *skip*
- T_4 by a_1
  m_5 : $\mathit{argue}(P \Rightarrow \mathtt{D}(z))$ where
  $P = \{\mathtt{G}(enjoyFood), tastyPizza, goToPizzaria \wedge tastyPizza \rightsquigarrow enjoyFood\}$
  m_6 : $\mathit{why\text{-}propose}(\mathtt{D}(b))$
- T_5 by a_2
  m_7 : $\mathit{argue}(T \Rightarrow \neg\mathtt{D}(z))$ where
  $T = \{\mathtt{G}(enjoyFood), dullTopping, goToPizzaria \wedge dullTopping \rightsquigarrow \neg enjoyFood\}$
  m_8 : $\mathit{argue}(S \Rightarrow \mathtt{D}(b))$ where
  $S = \{\mathtt{G}(enjoyFood), bestSteaks, goToBistro \wedge bestSteaks \rightsquigarrow enjoyFood\}$
- T_6 by a_3
  m_9 : $\mathit{argue}(D \Rightarrow \mathtt{D}(b))$ where
  $D = \{\mathtt{G}(enjoyFood), greatSeafood, goToBistro \wedge greatSeafood \rightsquigarrow enjoyFood\}$
- T_7 by a_1
  m_{10} : $\mathit{argue}(\neg m \Rightarrow \neg s)$ where $m = steakOnMenu$
- T_8 by a_2
  m_{11} : *skip*
- T_9 by a_3
  m_{12} : $\mathit{why}(d)$ where $d = dullTopping$
- T_{10} by a_1
  m_{13} : *skip*
- T_{11} by a_2
  m_{14} : $\mathit{argue}(n \Rightarrow d)$ where $m = menuNeverChanged\}$
  m_{15} : $\mathit{propose}(\mathtt{D}(p))$ where $b = goToPub$
  m_{16} : $\mathit{prefer}(b, p)$ m_{17} : $\mathit{prefer}(p, z)$
- T_{12} by a_3
  m_{18} : $\mathit{prefer}(b, p)$
- T_{13} by a_1
  m_{19} : $\mathit{reject}(p)$
  m_{20} : $\mathit{prefer}(z, b)$
  m_{21} : $\mathit{prefer\text{-}equal}(b, p)$
- T_{14} by a_2
  m_{22} : *skip*
- T_{15} by a_3
  m_{23} : *skip*
- T_{16} by a_1
  m_{24} : *skip*
- T_{17} by a_2
  m_{25} : *skip*

At that point, the proposal trees of the dialogue will look as represented Fig. 2. To see how the dialogical status and protocol rules affected the agents, consider turn T_5, in which agent a_2 tries to refute the proposal for $do(goToPizzeria)$ as made by agent a_1 and support its own proposal for $do(goToBistro)$.

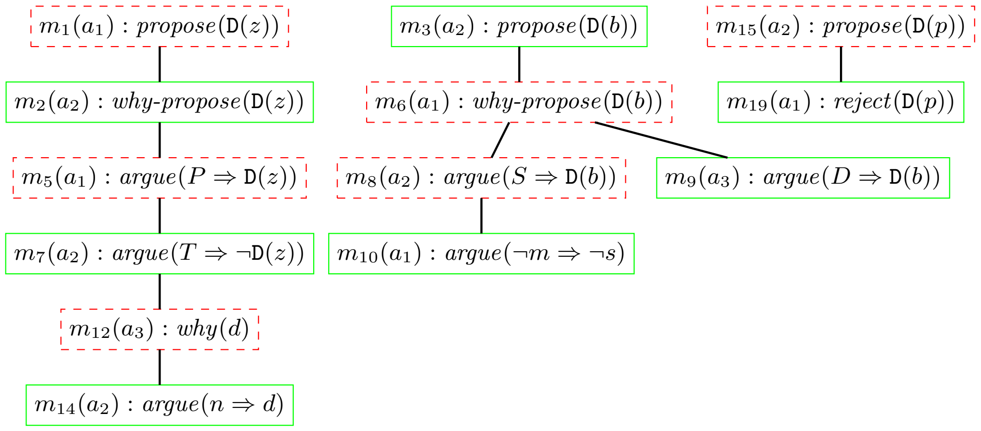

$m_1(a_1) : propose(\mathtt{D}(z))$
$m_2(a_2) : why\text{-}propose(\mathtt{D}(z))$
$m_5(a_1) : argue(P \Rightarrow \mathtt{D}(z))$
$m_7(a_2) : argue(T \Rightarrow \neg\mathtt{D}(z))$
$m_{12}(a_3) : why(d)$
$m_{14}(a_2) : argue(n \Rightarrow d)$
$m_3(a_2) : propose(\mathtt{D}(b))$
$m_6(a_1) : why\text{-}propose(\mathtt{D}(b))$
$m_8(a_2) : argue(S \Rightarrow \mathtt{D}(b))$
$m_9(a_3) : argue(D \Rightarrow \mathtt{D}(b))$
$m_{10}(a_1) : argue(\neg m \Rightarrow \neg s)$
$m_{15}(a_2) : propose(\mathtt{D}(p))$
$m_{19}(a_1) : reject(\mathtt{D}(p))$

Fig. 2. The proposal trees of the example

To somehow attack proposal $\mathtt{D}(goToPizzeria)$ the agent needs to find a point of attack, which should always be a relevant move. Within this proposal branch, the only points of attack are to attack m_5 or to move another reply to m_1. A relevant move to m_5 can be both an argue (rebuttal, undercutter or underminer) or a why move. Since the proposal move m_1 was already questioned with a *why-propose* the only remaining valid reply there is to move a *reject*($\mathtt{D}(goToPizzeria)$). The agent chooses to rebut the conclusion of m_5 with some argument $T \Rightarrow \neg\mathtt{D}(goToPizzeria)$.

Within the same turn, the agent also decides to give support to its own proposal $\mathtt{D}(goToBistro)$. To make this proposal *in*, it will have to find a relevant attack move. In this case the only legal attacking move is to forward an argument with conclusion $\mathtt{D}(goToBistro)$ in reply to m_6, which it does in the form $S \Rightarrow \mathtt{D}(goToBistro)$.

Weak relevance is displayed in turn T_6 where agent a_3 make the move $argue(D \Rightarrow \mathtt{D}(goToBistro))$. Although at that point a winning part for the proposal tree of $\mathtt{D}(goToBistro)$ already existed, specifically $\{m_3, m_6, m_8\}$, a new winning part $\{m_3, m_6, m_9\}$ is created. If instead strong relevance is used, then move m_9 is not relevant and thus illegal. In turn, the move m_{10} by agent a_1 is only weakly relevant because it removed one winning part in the proposal tree without changing the status of $\mathtt{proposal}(m_8)$.

The dialogue terminates after turn T_{25}, when agent a_2 was the first to skip twice in a continuous series of skips. The proposal moves of *goToPizzeria* and *goToPub* are *out* so those options are defensible. The proposal move of *goToBistro* on the other hand is *in* and so this option is justifiable. Partitioning the options set O according to the option status results in $O_j = \{goToBistro\}$ and $O_d = \{goToPizzeria, goToPub\}$. This gives a preliminary ordering $goToPizzeria \approx_p goToPub \prec_p goToBistro$. The agent orderings follow directly from the prefer moves they made. The agent option ordering for a_1 is $goToBistro \approx_{a_1} goToPub \prec_{a_1} goToPizzeria$, while that of a_2 is $goToPizzeria \prec_{a_2} goToPub \prec_{a_2} goToBistro$ and the ordering of a_3 is $goToBistro \prec_{a_3} goToPub$.

9 Basic Fairness and Efficiency Requirements

McBurney et al. [6] have proposed a set of 13 desiderata for argumentation protocols. These are criteria which dialogue game protocols need to adhere for basic fairness and efficiency. Each of the desiderata can be verified against our deliberation framework and protocol.

1. **Stated Dialogue Purpose** The protocol is explicitly designed to decide on a course of action.
2. **Diversity of individual purpose** Agents are allowed to have personal goals that possibly conflict with the stated mutual goal.
3. **Inclusiveness** Many agents can join the deliberation dialogue and no roles are enforced upon them.
4. **Transparency** The rules of our framework are fully explained, but it is up to an implementation to make sure every agents knows these rules and knows how to play the game.
5. **Fairness** Every agent has equal rights in the dialogue and the framework allows for fair winner selection methods. Since an agent may always choose not to move (any more) at all, it is never forced to adopt or drop some belief or goal.
6. **Clarity of Argumentation Theory** The reply structure and notion of relevance in our framework are not hidden implicitly in a protocol, but made explicit. Moreover, the structure of arguments is formalised in an explicitly defined argumentation logic and topic language.
7. **Separation of Syntax and Semantics** The communication language is separately defined from the protocol. Also, dialogues in the framework are independent of the agent specification while their public behaviour can still be monitored.
8. **Rule Consistency** We have not studied the rule consistency in detail, but the protocol will never lead to deadlocks; agents can always skip their turn or make a new proposal and within a proposal tree there is always a way to make a new contribution, as long as the top argue move was not conceded.
9. **Encouragement of Resolution** Agents are encouraged to stay focussed on the dialogue topic through the notion of relevance. If agents still have something to say, there is always the opportunity to do so.

10. **Discouragement of Disruption** Disruption is discouraged through the definition of legal speech acts, which are separated in attacks and replies. This restricts the available moves, for example agents cannot attack their own moves. However, it is still possible for aggressive agents to question everything that is claimed and no agent is compelled to accept any claim.
11. **Enablement of Self-Transformation** Agents are allowed to adjust their beliefs or goals depending on the arguments that are moved and preferences that are expressed. Moreover, they are allowed to drop proposals and to retract or concede claims.
12. **System Simplicity** Simplicity of the system is hard to prove or disprove. However, it is highly modular; communication and topic languages are separated and various alternative protocol rules may be adopted or dropped. The winner function is left unspecified, but this may range from a dictator agent to a social welfare-based function.
13. **Computational Simplicity** The computational implications of our framework have not yet been studied. However, the separation of agent and framework designs is at least one step towards simplifying the complexity.

Conforming to these guidelines does not yet mean that every dialogue will be fair and effective. A better understanding is needed of what fair and efficient deliberation dialogues are. Indeed, future work will need to assess how the deliberation process and outcome can be evaluated in relation to the initial situation. In contrast to beliefs, actions will never have an actual truth value but are rather more or less applicable in a specific situation. [5]

New research will also focus on more complete fairness and effectiveness results. For example it is interesting to see how agent attitudes [8] are influential in deliberation dialogues. Moreover, additional formal properties are interesting to study such as the correspondence between the dialogue outcome and the underlying logic of [10].

10 Related Work

The literature on argumentation theory for multi-agent systems includes several attempts at designing systems for deliberation dialogues. Earlier we already briefly discussed the most important work on argumentation in deliberation, i.e. that of McBurney et al. [5] They propose a very liberal protocol for agents to discuss proposals restricted by the advancement of a series of dialogue stages. The used speech acts are very similar to that of our framework, although no explicit logic is used to construct and evaluate arguments. Proposals can be forwarded or rejected, claims and arguments are made, questioned or retracted and preferences are expressed. The resulting commitments of agents are determined, but as in our model they are not used to restrict the legal moves.

Specific support is built into their system for discussion of different perspectives about the problem at hand. Perspectives are influential factors such as moral implications and costs. These perspectives can be integrated in our framework as well though the adopted topic language and logic. One model that could be adopted is proposed in [15].

Agents in the framework of McBureny et al. are constrained in their utterances only by preconditions of the different speech acts. For example, they may not state a preference on two actions before they are asserted. Our model accomplishes this through the explicit reply structure of moves rather than using preconditions. Moreover, our model can enforce dialogical coherence through the notion of move relevance.

To decide on a winning proposal agents need to unanimously accept some proposal or a voting system may be used. This way any knowledge of the arguments on proposals is discarded. In contrast, our model may utilise this knowledge on the multi-agent level to decide on a fair winner without the need for a consensus.

A dialogue protocol on proposals for action is introduced in the work of Atkinson et al. [2] They list all the possible ways to attack a proposal for action, including the circumstances, the goal it achieves and the values it promotes. In our framework, both the goal and action itself are explicitly stated, while the circumstances appear within the arguments that are moved in our deliberation dialogues. As explained earlier, support for values, which are similar to the perspectives of McBurney et al. [5], will be added later.

Many locutions are available to attack proposals, like 'deny goal exists' or 'ask circumstances'. These are needed because no explicit reply structure is present. This also means that no direct relation between the attacks and the resolution of conflicting statements can be made. It is assumed that agents eventually agree on the subject at hand, agree to disagree or use a separate argumentation framework to establish the validity of the proposal. Moreover, the complete work only covers dialogues on a single proposal for action, which makes it persuasion rather than deliberation, albeit being about actions instead of beliefs.

A practical application of multi-agent deliberation dialogues was developed by Tolchinsky et al. [13] A model for discussion on proposals is coupled to a dialogue game. In the model, agents are proponents or opponents of some proposal for action, while a mediator agent determines the legal moves and evaluates moved arguments to see if they are appropriate and how they support or criticize the proposal for action. Although the paper focusses on the translation and application of argument schemes, it is interesting to see how their work can be modelled inside our framework. The number of proposals is limited to a single action, namely to transplant some organ to some recipient, with a mutual goal to find the best organ donor. A dialogue has to start with propose, reject and why-reject moves after which agents can play argue moves. Whether the proposal is also the winner is determined an the authoritative mediator agent.

11 Conclusions

In this paper a framework for multi-agent deliberation dialogues has been proposed. The contribution is twofold.

The general framework for persuasion-type dialogues of Prakken [10] has been altered to provide support for multi-party deliberation dialogues. Consequently, non-trivial modifications have been made to the framework. First, support for moving, criticizing and preferring proposals for action was added. By reusing the explicit reply structure we represent deliberation dialogues as directed multiple trees. Second, the notions of dialogical status and relevance have been adapted for multiple agents. In particular, surrendering replies in a multi-agent context are studied and how strong and weak relevance can still be maintained.

Our framework also improves on the existing work on deliberation dialogues. In contrast with McBurney et al. [5], conflicts of interest are handled through a persuasion-style explicit move status. This allows for varying ways to impose coherence on the deliberating agents. Moreover, the status of proposals is used to define a classification so a preliminary ordering on them can be made. This, together with the agents' explicit preferences, may be used to select a winning proposal.

The framework was checked against the desiderata for multi-agent argumentation protocols. Deliberation systems in our framework will adhere to those basic standards for efficiency and effectiveness. A more rigid study on formal properties of the framework will be valuable here as well as a study on how different agent strategies can affect fairness and effectiveness.

As an extension of our framework, we could allow agents to discuss not only beliefs but also goals, values and preferences. For example, attacking of prefer moves could be allowed, by which a new argument tree is started. A preference-based argumentation framework [7] may be used to in turn evaluate the effect on the dialogical status of proposals. To support discussion on values the topic and communication languages can be extended. One option is to incorporate the work of Black and Atkinson [3], who explicitly allow discussion on promoted values.

Acknowledgments This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.066.823.

References

1. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence 173(3-4), 413–436 (2009)
2. Atkinson, K., Bench-Capon, T.J.M., McBurney, P.: A dialogue game protocol for multi-agent argument over proposals for action. Autonomous Agents and Multi-Agent Systems 11(2), 153–171 (2005)

3. Black, E., Atkinson, K.: Dialogues that account for different perspectives in collaborative argumentation. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems. pp. 867–874. Budapest, Hungary (2009)
4. Dung, P.M.: 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)
5. McBurney, P., Hitchcock, D., Parsons, S.: The eightfold way of deliberation dialogue. International Journal of Intelligent Systems 22(1), 95–132 (2007)
6. McBurney, P., Parsons, S., Wooldridge, M.: Desiderata for agent argumentation protocols. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems. pp. 402–409. Bologna, Italy (2002)
7. Modgil, S.: Reasoning about preferences in argumentation frameworks. Artificial Intelligence 173(9-10), 901–934 (2009)
8. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
9. Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Aggregating Partially Ordered Preferences. Journal of Logic and Computation 19(3), 475–502 (2008)
10. Prakken, H.: Coherence and Flexibility in Dialogue Games for Argumentation. Journal of Logic and Computation 15(6), 1009–1040 (2005)
11. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation 1(2) (2010 (to appear))
12. Tang, Y., Parsons, S.: Argumentation-based dialogues for deliberation. In: Proceedings of the 4th International Conference on Multi-Agent Systems. pp. 552–559. ACM (2005)
13. Tolchinsky, P., Atkinson, K., McBurney, P., Modgil, S., Cortés, U.: Agents deliberating over action proposals using the ProCLAIM model. In: Proceedings of the 5th International Central and Eastern European Conference on Multi-Agent Systems. pp. 32–41. Springer-Verlag New York Inc, Leipzig, Germany (2007)
14. Walton, D.N., Krabbe, E.C.W.: Commitment in dialogue: Basic concepts of interpersonal reasoning. State University of New York Press, New York (1995)
15. van der Weide, T.L., Dignum, F.P.M., Meyer, J.J.C., Prakken, H., Vreeswijk, G.A.W.: Practical reasoning using values: Giving meaning to values. In: Proceedings of the 6th International Workshop on Argumentation in Multi-Agent Systems. pp. 225–240. Budapest, Hungary (2009)

Towards Pragmatic Argumentative Agents within a Fuzzy Description Logic Framework

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. To bring the level of current argumentation to the expressive and flexible status expected by human agents, we introduce fuzzy reasoning on top of the classical Dung abstract argumentation framework. The system is built around Fuzzy Description Logic and exploits the integration of ontologies with argumentation theory, attaining the advantage of facilitating communication of domain knowledge between agents. The formal properties of fuzzy relations are used to provide semantics to the different types of conflicts and supporting roles in the argumentation. The usefulness of the framework is illustrated in a supply chain scenario.

1 Introduction

Abstract argumentation frameworks lack high-level conveniences such as ease of understanding, an aspect required by human agents. Many challenges still exists in order to build intelligent systems based on abstract argumentation frameworks [1].
On the one hand, humans manifest a lot of flexibility when they convey arguments from supporting and attacking a claim. One can disagree, can provide a counter example, can rebut or undercut a claim. Currently, these common patterns of attacking relations are encapsulated as argumentation schemes [2]. This informal reasoning does not exploit the formal properties of the attacking relations. The semantics of the support relation *agree* contains the transitivity property: $agree(a, b)$ and $agree(b, c)$ implies that *a agrees with c*. Similarly, the rebutting relation is symmetrical. That is, $rebut(a, \neg a)$ implies $rebut(\neg a, a)$. In this paper, we advocate to use such properties when deciding on the status of an argument. We provide software agents with description logic based reasoning capabilities to exploit the formal properties of the attacking relations.
On the other hand, people do not express their arguments precisely in their daily life. Such vague notions as: strongly, moderately, don't fully agree, tend to disagree are used during an argumentative dialog. Real arguments are also a mixture of fuzzy linguistic variables and ontological knowledge. Arguments conveyed by people are incomplete, normally *enthymemes* [3], where the opponent of the arguments assumes that his

Operation		Lukasiewicz Logic	Gödel Logic
intersection	$\alpha \otimes_S \beta$	$max\{\alpha+\beta-1,0\}$	$min\{\alpha,\beta\}$
union	$\alpha \oplus_S \beta$	$min\{\alpha+\beta,1\}$	$max\{\alpha,\beta\}$
negation	$\ominus_S \alpha$	$1-\alpha$	$1,\ \mathit{if}\ \alpha=0,\ 0,\ \mathit{otherwise}$
implication	$\alpha \Rightarrow_S \beta$	$min\{1,1-\alpha+\beta\}$	$1, \mathit{if}\, \alpha \leqslant \beta, \beta, \mathit{otherwise}$

Table 1. Operators in Fuzzy Logics.

partner understands the missing part. Thus, a common knowledge on the debate domain is assumed by the agents. We introduce Fuzzy Description Logic on top of the argumentation theory, as the adequate technical instrumentation needed to model real-life debates.

2 Preliminaries

2.1 Fuzzy Sets and Relations

A fuzzy relation R between two set A and B has degree of membership whose value lies in [0, 1]: $\mu_R : A \times B \to [0,1]$. $\mu_R(x,y)$ is interpreted as strength of relation R between x and y. When $\mu_R(x,y) \geq \mu_R(x',y')$, (x,y) is more strongly related than (x',y'). A fuzzy relation R over X × X is called:

- *transitive*: $\forall\, a,b \in X, R(a,c) \geq sup_{b\in X}\{\otimes_S(R(a,b),R(b,c))\}$
- *reflexive*: $\forall\, a \in X, R(a,a) = 1$
- *irreflexive*: $\exists\, a \in X, R(a,a) \neq 1$
- *antireflexive*: $\forall\, a \in X, R(a,a) \neq 1$
- *symmetric*: $\forall\, a,b \in X, R(a,b) \to R(b,a)$
- *antisymmetric*: $\forall\, a,b \in X, R(a,b) \to \neg R(b,a)$
- *disjoint*: $\forall\, a,b \in X, \otimes_S(R(a,b),S(a,b)) = 0$

The *inverse* of a fuzzy relation $R \subseteq X \times Y$ is a fuzzy relation $R^- \subseteq Y \times X$ defined as $R^-(b,a) = R(a,b)$. Given two fuzzy relations $R_1 \subseteq X \times Y$ and $R_2 \subseteq Y \times Z$ we define the *composition* as $[R_1 \circ R_2](a,c) = sup_{b\in Y}\{\otimes_S(R(a,b),R(b,c))\}$ (table 1). The composition satisfies the following properties: $(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$, and $(R_1 \circ R_2)^- = (R_2^- \circ R_1^-)$. Due to the associativity property we can extend the composition operation to any number of fuzzy relations: $[R_1 \circ^t R_2 \circ^t ... \circ^t R_n](a,b)$. If a relation is reflexive, antisymmetric, and transitive it is called *order relation*.

2.2 Fuzzy Description Logic

In the following paragraphs the differences introduced by fuzzy reasoning on top of classical description logic are presented. The complete formalization of the fuzzy description logic can be found in [4]. The *syntax* of fuzzy *SHIF* concepts [4] is as follows:

$$\begin{aligned} C, D &= \top \mid \bot \mid A \mid C \sqcap_S D \mid C \sqcup_S D \mid C \sqsubseteq_S D \mid \neg_L C \mid \\ &\quad \forall R.C \mid \exists R.C \mid \forall T.d \mid \exists T.d \mid \leq nR \mid \geq nR \mid m(C) \mid \{a_1, ... a_n\} \\ d &= crisp(a, b) \mid triangular(a, b, c) \mid trapezoidal(a, b, c, d) \end{aligned}$$

where S=$\{L, G, C\}$, L comes from Lukasiewicz semantics, G from Gödel semantics, and C stands for classical logic (see table 1). The modifier $m(C) = linear(a) \mid triangular(a, b, c)$ can be used to alter the membership functions of the fuzzy concepts. Fuzzy modifiers such as *very*, *more-or-less*, *slightly* can be applied to fuzzy sets to change their membership functions. They are defined in terms of linear hedges. For instance, one can define *very=linear (0.8)*. A functional role S can always be obtained by means of the axiom $\top \sqsubseteq (\leq 1S)$.

Example 1. The definition of junk food is applied to some food which has little nutritional value, or to products with nutritional value but which also have ingredients considered unhealthy: $JunkFood = Food \sqcap (\exists\, hasNutritionalValue.Little \sqcup \exists\, hasIngredients.Unhealthy)$. In this definition, there are two roles which point to the fuzzy concepts *Little* and *Unhealthy*, which could be represented as $Little = triangular(10, 20, 30)$, or $Unhealthy = \exists\, hasSalt. \geq 2mg \sqcup hasAdditive. > 0.5mg$.

$\bot^I(x) = 0$	$(\forall R.C)^I(x) = inf_{y \in \Delta^I} R^I(x, y) \Rightarrow_S C^I(y)$
$\top^I(x) = 1$	$(\exists R.C)^I(x) = sup_{y \in \Delta^I} R^I(x, y) \otimes_S C^I(y)$
$(\neg C)^I = \ominus C^I(x)$	$(\forall T.d)^I(x) = inf_{y \in \Delta^I} R^I(x, v) \Rightarrow_S d^I(y)$
$(C \sqcap_S D)^I(x) = C^I(x) \otimes_S D^I(x)$	$(\exists R.d)^I(x) = sup_{y \in \Delta^I} R^I(x, v) \otimes_S d^I(y)$
$(C \sqcup_S D)^I(x) = C^I(x) \oplus_S D^I(x)$	$(x : C)^I = C^I(x^I)$
$(C \rightarrow_S D)^I(x) = C^I(x) \Rightarrow_S D^I(x)$	$((x, y) : R)^I = R^I(x^I, y^I)$
$(m(C))^I(x) = f_m(C^I(x))$	$(C \sqsubseteq D)^I(x) = inf_{x \in \Delta^I} C^I(x) \Rightarrow_S D^I(x)$

Fig. 1. Semantics of fuzzy concepts.

The main idea of *semantics* of FDL is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain [4]. A fuzzy interpretation $I = (\Delta^I, \bullet^I)$ consists of a non empty set Δ^I (the domain) and a fuzzy interpretation function $\bullet^I$. The mapping $\bullet^I$ is extended to roles and complex concepts as specified in figure 1.

3 Fuzzy Argumentation Systems

3.1 Fuzzy Resolution Argumentation Base

An argumentation framework [5] consists of a set of arguments, some of which attack each other. In our approach, the arguments represent instances of concepts, while different types of attack relations are instantiations of roles defined on these concepts. Both, the concepts and the roles can be fuzzy.

Definition 1. *A fuzzy resolution argumentation base is a tuple $FRA = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, consisting of a fuzzy Abox $\mathcal{A}$, representing argument instances and their attacking relations, a fuzzy Tbox $\mathcal{T}$ representing concepts, and a fuzzy Rbox $\mathcal{R}$ encapsulating attack-like and support-like relations.*

Definition 2. *A fuzzy Abox $\mathcal{A}$ is a tuple $\prec Arg, Attacks \succ$, where Arg of a finite set of assertion axioms for fuzzy arguments $\{a_1 : C_1 \bowtie \alpha_1, a_2 : C_2 \bowtie \alpha_2, ..., a_n : C_n, \bowtie \alpha_n\}$, and Attacks is a set of fuzzy roles $\subseteq Arg \times Arg$ of the form $\{(a_i, a_j) : R_k \bowtie \alpha_l\}$, where $\alpha_l \in [0, 1]$, C_i are concepts, R_k are attack and support-like relations, and $\bowtie = \{<, \leq, >, \geq\}$.*

Example 2. Let $\mathcal{A} = \prec \{funghi : CheapPizza \geq 0.8\}, \{(funghi, vegetarian) : Attack \geq 0.7\} \succ$ states that *funghi* is a *CheapPizza* with degree at least 0.8, and it attacks the *vegetarian* argument with degree at least 0.7.

If α is omitted, the maximum degree of 1 is assumed. We use $\bowtie^-$ as the reflection of inequalities $\leq^- = \geq$ and $<^- = >$.

Definition 3. *A fuzzy Tbox $\mathcal{T}$ is a finite set of inclusion axioms $\{C_i \sqsubseteq_S D_i, \geq \alpha_i\}$, where $\alpha_i \in [0, 1]$, C_i, D_i are concepts, and S specifies the implication function (Lukasiewicz, Gödel) to be used. The axioms state that the subsumption degree between C and D is at least α.*

Example 3. Let's take the common example of pizza. Can it be categorized as junk food or nutrition food? Associated with some food outlets, it is labeled as "junk", while in others it is seen as being acceptable and trendy. Rather, one can consider that it belongs to both concepts with different degree of truth, let's say 0.7 for *JunkFood* and 0.5 to *NutritionFood*, encapsulated as $\mathcal{T} = \{Pizza \sqsubseteq_L JunkFood \geq 0.7, Pizza \sqsubseteq_L NutritionalFood \geq 0.5, FreshFruits \sqsubseteq_L NutritionalFood, CandyBar \sqsubseteq_L JunkFood\}$. Note the subconcept *CandyBar* is subsumed by the concept *JunkFood* with a degree of 1.

Definition 4. *The argumentation core $\mathcal{R}^k$ of the fuzzy Rbox $\mathcal{R}$ consists of two relations Attack and Support (noted by $\bar{A}$, respectively $\bar{S}$), having the properties: i) dis($\bar{A}$,$\bar{S}$), meaning that $\forall\, a, b \in Arg, \otimes((a, b) : \bar{A}, (a, b) : \bar{S}) = 0$. Formally, $\mathcal{R}^k = \{\bar{A},\ \bar{S},\ dis(\bar{A},\bar{S})\}$.*

Under the Gödel semantics, the disjoint property of the *Attack* and *Support* relations states that $\otimes_G((a, b) : \bar{A}, (a, b) : \bar{S}) = min((a, b) : \bar{A}, (a, b) : \bar{S}) = 0 \Leftrightarrow$ if $(a, b) : \bar{S} \geq 0$ then $(a, b) : \bar{A} \leq 0$ and if $(a, b) : \bar{A} \geq 0$ then $(a, b) : \bar{S} \leq 0$. In other words, if a attacks b there is no support relation from a to b, and similarly if a supports b there is no attack relation from a to b. The Lukasiewicz semantics leads to a more flexible interpretation, given by $\otimes_L((a, b) : \bar{A}, (a, b) : \bar{S}) = max((a, b) : \bar{A} + (a, b) : \bar{S} - 1, 0) = 0 \Leftrightarrow (a, b) : \bar{A} + (a, b) : \bar{S} \leq 1$. Thus, if a attacks b to a certain degree α, there exists the possibility that also a supports b with a maximum degree of $1 - \alpha$. While the Gödel interpretation fits perfectly to the general case of argumentative debates, some special examples lay under the Lukasiewicz semantics.

Definition 5. *The fuzzy Rbox $\mathcal{R}$ consists of i) the argumentation core $\mathcal{R}^k$; ii) an hierarchy of disjoint attack and support-like relations R, defined by role inclusion axioms: $R \sqsubseteq Attack$, or $R \sqsubseteq Support$; and iii) a set of role assertions of the form: $(fun\ R)$, $(trans\ R)$, $sym(R)$, $(inv\ R\ R^-)$, stating that the role R is functional, transitive, symmetric, respectively its inverse relation is R^-.*

There are two types of relations in $\mathcal{R}$: *supporting roles* (denoted by $\mathcal{R}^S$), opposite to *attacking roles* (denoted by $\mathcal{R}^A$), where $\mathcal{R}^S \cap \mathcal{R}^A = \varnothing$. We note that a_1 supports a_2 by $a_1 \rightarrow a_2$ and a_1 attacks a_2 by $a_1 \nrightarrow a_2$.

Example 4. Let $\mathcal{R} = \mathcal{R}^k \cup \{Defeat, Disagree, Agree, Defeat \sqsubseteq \bar{\text{A}}, Disagree \sqsubseteq \bar{\text{A}}, Agree \sqsubseteq \bar{\text{S}}, tra(Agree), sym(Agree), ref(Agree)\}$. The two hierarchies are $\mathcal{R}^A = \{\bar{\text{A}}, Defeat, Disagree\}$, respectively $\mathcal{R}^S = \{\bar{\text{S}}, Agree\}$. Note that *Support* relation is transitive, while *Attack* role is not a transitive one; *Agree* is a particular instance of *Support* relation, while *Disagree* and *Defeat* relations are *Attack*-type relations. The following properties hold:

Proposition 1. *Attack and Support-like relations are not functional, i.e. the same argument a can attack two different arguments $b_1 \neq b_2$: $(a, b_1) : Attack$ and $(a, b_2) : Attack$.*

Proposition 2. *The inverse of the Attack relation is an attack-like relation (inv Attack $\sqsubseteq$ Attack).*

Proposition 3. *An argument a agrees to itself $(a, a) : Agree$, given by the the reflexivity property of the Agree relation.*

Example 5. Consider $FRA = \langle \prec \{a : A, b : B, c : C\}, \{(a, b) : Agree \geq 0.9, (b, c) : Agree \geq 0.8\}, \{A, B, C\}, \mathcal{R}^k \cup \{Agree \sqsubseteq \bar{\text{S}}, tra(Agree)\}$. *Agree* being a transitive relation, the argument a also agrees to c with a degree of $\alpha \geq sup_{b \in Arg}\{\otimes_S((a, b) : Agree, (b, c) : Agree)\}$, which gives $max(0.9 + 0.8 - 1, 0) = 0.7$ under Lukasiewicz semantics and $min(0.9, 0.8) = 0.8$ in Gödel interpretation.

Proposition 4. *The relation S is complement of the relation R if $(inv\ R\ S)$ and $(x, y) : R \bowtie \alpha \rightarrow (y, x) : S \bowtie^- (1-\alpha)$. Here, Agree and Disagree are complement relations, $(a, b) : Agree \geq \alpha$ implies $(b, a) : Disagree \leq 1-\alpha$. Informally, if a and b agree each other with at least α, the disagreement degree between them should be less then $1 - \alpha$.*

Definition 6. *(Argumentation Chains) An argument b is supported by the argument a if their is a finite path $p = (a, x_1) : R_1, (x_1, x_2) : R_2, ..., (x_{n-1}, b) : R_n, \forall R_i, R_i \sqsubseteq Support$. An argument b is attacked by the argument a if their is a finite path $p = (a, x_1) : R_1, (x_1, x_2) : R_2, ..., (x_{n-1}, b) : R_n, \forall R_i, R_i \sqsubseteq Support \sqcup Attack$, and the number of attack relations $\mid \mathcal{R}^A \mid$ is odd.*

Proposition 5. *(Indirect Support) By composing an even number of attack relations, one gets an indirect support relation. Formally, $R_1 \sqsubseteq Attack$, $R_1 \sqsubseteq Attack$, implies $R_1 \circ^t R_2 \sqsubseteq Support$. The norm used to compute the strength of the attack is $\circ^t_{\bar{A}} = \otimes^2_S$, where the power 2 models the fact that an indirect attack should be smaller than a direct one.*

Example 6. Consider $FRA = \langle \prec \{a : A, b : B, c : C\}, \{(a, b) : Undercut \geq 0.9, (b, c) : Attack \geq 0.7\} \succ, \{A, B, C\}, \mathcal{R}^k \cup \{Disagree, Undercut, Disagree \sqsubseteq Attack, Undercut \sqsubseteq Attack\}$. By applying complex role inclusion we obtain $Attack \circ^t Disagree \sqsubseteq Support$. In other words, if the argument a attacks b and b disagrees with c we say that there is a support-like relation between a and c: $(a, c) : R > 0, R \in \mathcal{R}^S$. The degree of support is given under the Lukasiewicz semantics as $Undercut \otimes_L^2 Disagree = (sup_{b:B}\{max(0, 0.9 + 0.7 - 1)\})^2 = (0.6)^2 = 0.36$ and $Undercut \otimes_G^2 Disagree = (sup_{b:B}\{min(0.9 + 0.7)\})^2 = 0.49$, under Gödel semantics.

3.2 Aggregation of Arguments

Several issues are raised by merging description logic and fuzzy argumentation: What happens when there is more than one attack-like relation between two concepts? What happens when one argument belongs with different membership functions to several concepts, which are linked by different attack-like relations with the opposite argument? What happens when two independent arguments attack the same argument? Should one take into consideration the strongest argument, or both of them may contribute to the degree of truth of that concept? Given an argumentation system, a semantic attaches a status to an argument. Different semantics may lead to different outputs [6].
One advantage of fuzzy logic is that it provides technical instrumentation (Lukasiewicz semantics, Gödel semantics) to handle all the above cases in an argumentative debate. The interpretation of Gödel operators maps the *weakest link principle* [7] in argumentation, which states that an argument supported by a conjunction of antecedents α and β, is as good as the weakest premise $\otimes_G = min(\alpha, \beta)$. The reason behind this principle is the fact that the opponent of the argument will attack the weakest premise in order to defeat the entire argumentation chain. When two reasons supporting the same consequent are available, having the strength α and β, the strongest justification is chosen to be conveyed in a debate, which can be modeled by the Gödel union operator $\oplus_G max\{\alpha, \beta\})$.
The Lukasiewicz semantics fits better to the concept of *accrual of arguments*. In some cases, *independent* reasons supporting the same consequent provide stronger arguments in favor of that conclusion. Under the Lukasiewicz logic, the strength of the premises (α,β) contributes to the confidence of the conclusion, given by $\oplus_L = min\{\alpha + \beta, 0\}$. For instance, the testimony of two witnesses is required in judicial cases. Similarly, several reasons against a statement act as a form of collaborative defeat [7].
One issue related to applying Lukasiewicz operators to argumentation regards the difficulty to identify independent reasons. Thus, an argument presented in different forms contributes with all its avatars to the alteration of the current degree of truth.
Thus, the description logic provides the technical instrumentation needed to identify independent justifications, whilst the Lukasiewicz semantics offers a formula to compute the accrual of arguments. The accrual of dependent arguments is not necessarily useless. Changing the perspective, this case can be valuable in persuasion dialogs, where an agent,

by repeatedly posting the same argument in different forms, will end in convincing his partner to accept it.

3.3 Resolution Schemes

The key limitation of conventional systems is that, even if they guarantee to compute a solution for consistent sets, admissible or preferred extensions, it is possible that the only answer to be the empty set.

Definition 7. *The preference relation Pref* $\subseteq Arg \times Arg$ *is a fuzzy role having the following properties: (ref P), (tran P), and (antysim P). The Rbox* $\mathcal{R}$ *extended with preferences is given by* $\mathcal{R}^P = \mathcal{R} \cup \{Pref\}$*. An argument a is preferred to b* ($a \succ b$) *based on the preference role Pref with a degree* α *if* $(a, b) : Pref \bowtie \alpha$*.*

Example 7. Consider the task to classify a compound according to potential toxicity. In the guidelines of U.S. Environmental Protection Agency for the assessment the health impacts of potential carcinogens, an argument for carcinogenicity that is based on human epidemiological evidence is considered to outweigh arguments against carcinogenicity that are based only on animal studies. The corresponding FRA (figure 2) augmented with preferences is
$FRA^{\mathcal{P}} = \langle \prec (h : HumanStudy, a : A, b : B, c : Carcinogenicity, d : AnimalStudy, (a, h) : BasedOn, (a, c) : For, (b, d) : BasedOn, (b, c) : Against, (a, b) : Outweigh \succ, \{HumanStudy, AnimalStudy, Carcinogenicity, A, B\}, \mathcal{R}^k \cup \{For \sqsubseteq Support, Against \sqsubseteq Attack, (inv\ BasedOn \sqsubseteq Support)\} \cup \{Outweigh \sqsubseteq Pref\}\rangle$. Observe that if a is based on h then there exists a support-like relation from h towards a. Formally ($inv\ BasedOn \sqsubseteq Support$).

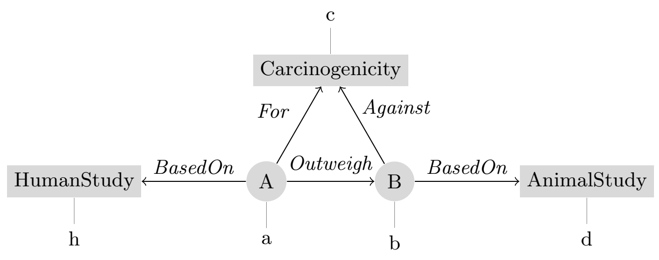


Fig. 2. Explicit preference among arguments

In the above example, the preference assertion *Outweigh* between the arguments a and b was explicitly given, stating that a clearly outweighs b ($\alpha = 1$). Note that any preference role can be a fuzzy one. When this explicitness does not exist, FDL offers the possibility to infer preference relations among arguments based on various *conflict resolution strategies*, like the following ones.

- *Fuzzy membership value (M).* The status of an argument is assessed by comparing the membership degrees of the arguments to their concepts. Prior information is usually provided by an expert or knowledge engineer.
- *Specificity (S).* This heuristic can be applied both on concepts, in which case the most specific argument dominates, and roles, where the most specific relation in the hierarchy prevails (figure 3).
- *Value based argumentation (V).* The argument which promotes the highest value according to some strict partial ordering on values will defeat its counter-argument. In a FRA, arguments can promote (or demote) values to a given degree, so that if the arguments a and b promote the same value v, we consider that a successfully attacks b if it promotes v to a greater degree than b. In the current framework, values can be used from an ontology of values, providing a reasoning mechanism over values.

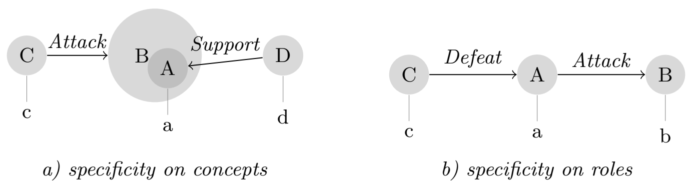


Fig. 3. Conflict Resolution Strategies

Example 8. (Specificity on concepts) $FRA^{\mathcal{P}} = \langle \prec (a : A, c : C, d : D, A \sqsubseteq B \succ, \{A, B, C, D\}, \mathcal{R}^k\{(C, B) : \mathit{Attack}, (D, A) : \mathit{Support}\} \rangle$. So, a as an element of $A \sqsubseteq B$ is supported by d, and attacked by c (figure 3a). In this case the specificity principles says that the support relation will prevail.

Example 9. Now consider the case in which $(a, b) : \mathit{Defeat}, (b, c) : \mathit{Attack}$ and $\mathit{Defeat} \sqsubseteq \mathit{Attack}$ (figure 3b). Based on the specificity heuristic on roles, the *Defeat* relation is stronger (more specific) than *Attack*. Consequently, the only admissible set is $\{a, c\}$.

The specificity preference is also illustrated by the following dialog:
A: *I am very hungry. Let's go eat something.*
B: *I am a little hungry too. I agree.*
A: *I don't have too much time. Let's have a pizza.*
B: *It's not healthy. I prefer something else. What about fish and wine.*
A: *It's too expensive.*
Here, the agent B accepts the argument *Food*, defined as $\mathit{Food} = \exists\, \mathit{canEat}.\mathit{Self}$ and supported by the argument *hungry*. Observe that the support is stronger from the agent A, given by the fuzzy modifier *very*, and not so

convincing as denoted by the modifier *little*. However, the more specific argument *pizza*, which is a kind of food (*Pizza* $\sqsubseteq$ *JunkFood* $\sqsubseteq$ *Food*) is rejected by the agent B. Similarly, the argument *fish* conflicts with the argument *expensive*. In many real life discussions, people have agreements at certain level of generality, while they manifest divergent opinions starting with a given level of specificity. Description logic is particularly useful to define the edge between agreement and disagreement. In this particular case, the agreed concept would be $\mathit{NutritionalFood} \sqcap \forall\, \mathit{hasPrice}.\neg \mathit{Expensive}$. The instance that belongs to this concept with the highest degree will best satisfy the both agents. If such an assertion does not exist, the agreement is reached by a preference relation over the common constraints: *healthy*, *not expensive*, *quick*.
Note that preferences are fuzzy relations, meaning that linguistic scale can be defined on them. Preferences like: *just equal, weakly more important, fairly strongly preferred, absolutely outweighed* are accepted in a $FRA^{\mathcal{P}}$.

Proposition 6. *By composing two preference relations we get a preference relation. In order not to breach the transitivity property, the composition function that we use is* $\circ^t_{Pref} = \otimes_S^{1/2}$.

Example 10. Let $(a, b) : \mathit{Outweigh} \geq 0.9$ and $(b, c) : \mathit{Pref} \geq 0.7$. The preference degree between a and c is given by $\mathit{Outweigh}(a, b) \circ^t \mathit{Pref}(b, c) = \mathit{min}(0.9, 0.7)^{1/2} = 0.83$.

3.4 Semantic Inconsistency

An important aspect is that inconsistency is naturally accommodated in fuzzy logic: the intersection between the fuzzy concept A and its negation is not 0 ($A \sqcap \neg A \neq 0$). Similarly, the disjoint property of an attack $A_1 \sqsubseteq \mathit{Attack}$ and support $S_1 \sqsubseteq \mathit{Support}$ relation $\otimes_S((a, b) : A_1 \geq \alpha, (a, b) : S_1 \geq \beta) = 0$, under the Lukasiewicz interpretation leads to an inconsistent argumentation base if $\alpha + \beta > 1$. Consider the concept *InternallyConsistentArguments* (ICA) defined in FDL as: $\mathit{ICA} \equiv \mathit{Argument} \sqcap \neg \exists\, \mathit{Attack}.\mathit{ICA}$. Based on the $(\forall\, R.C)^I = (\neg \exists\, R.\neg C)^I$, which holds under the Lukasiewicz semantics [8], follows that: $\mathit{ICA} \equiv \mathit{Argument} \sqcap \forall\, \mathit{Attack}.\neg \mathit{ICA}$. The semantics of $\forall\, R.C$ being $(\forall\, R.C)^I = \mathit{inf}_{y \in \Delta^I} = R^I(x, y) \rightarrow C^I(y)$ implies in FRA that if $(x, y) : \mathit{Attack} \bowtie \alpha \rightarrow_L y : \neg \mathit{ICA} \bowtie \beta$ The implication holds if $1 - \alpha + \beta \geq 1$ (recall table 1), or $\alpha \leq \beta$. In order to keep the argumentation base semantically consistent the following constraints exist, where $\gamma = 1 - \beta$, represents the degree of y to the ICA concept:

- $(x, y) : \mathit{Attack} \leq \alpha \Rightarrow \alpha \leq \beta \Leftrightarrow \gamma < 1 - \alpha$: If the attack relation between x and y is maximum α, the knowledge base remains consistent as long as y belongs to the concept ICA no more than $1 - \alpha$.
- $(x, y) : \mathit{Attack} \geq \alpha \Rightarrow \beta = 1 \Leftrightarrow \gamma = 0$: If the attack relation between x and y is at least α, the knowledge base is guaranteed to remain consistent if y does not belong to ICA at all.
- $y : \neg \mathit{ICA} \leq \beta \Rightarrow \alpha = 0$: If y belongs to the concept $\neg \mathit{ICA}$ with maximum β, it means that it should belong to the opposite concept

IAC at least $1 - \beta$. Consequently, no attack relation should exist between x and y.
– $y : \neg ICA \geq \beta \Rightarrow \alpha \leq \beta \Leftrightarrow \gamma \leq 1 - \alpha$.

The notion of indirect support in combination with the disjoint property of the attack and support relation may help to signal semantic inconsistencies in an argument bases. If A attacks B attacks C and A attacks C, then A indirectly both supports C and attacks C.

Example 11. Consider the situation in which $(A, B) : attack_{0.6}$, $(B, C) : attack_{0.9}$, and also $(A, C) : attack_{0.7}$. Under the Lukasiewicz semantics, the indirect support from A to C equals $max(0.6 + 0.9 - 1, 0)^2 = 0.25$. The disjoint property of attack and support holds because $max(0.25 + 0.7 - 1, 0) = 0$. If the strength of the attack from A to C increases to 0.7 the disjoint property is violated. In this case, the framework signals to the human agent that the argument base is semantically inconsistent. In other words, the alert means that on the these particular argumentation chains the strengths of the attacks or supports might be incorrect stated and the initial facts should be reconsidered.

There is no need to explicitly define simple negation on roles, as it exists in FDL systems by mean of assertions that use the inequalities, $\leq$ and $<$ [9]. For instance, the assertion *x does not attack y* can be defined as $(x, y) : Attack \leq 0$

4 Argumentative Agents in FRA

Definition 8. *A preference scheme specifies the order in which the conflict resolution strategies are applied. An example of preference scheme is MSV (fuzzy membership value, specificity, value-based).*

Preference schemes come from the cognitive system of the agents. A $FRA^{\mathcal{P}}$ still allows a lack of complete transitive preference.

Definition 9. *An agent is a tuple* $Ag = [PreferenceScheme, (\oplus, \otimes, \ominus, \rightarrow), (\circ_{Pref}, \circ_{\bar{A}}, \circ_{\bar{S}})]$*, where* $\oplus, \otimes, \ominus, \rightarrow$ *represent the union, intersection, negation and implication operators, and* $\circ_{Pref}, \circ_{\bar{A}}, \circ_{\bar{S}}$ *the norm used for composing preferences, attacks, and support relations. KB encapsulates the private domain information of the agent.*

We assume that agents acting within the same $FRA^{\mathcal{P}}$ share a common vocabulary of attacking, supporting, and outranking fuzzy relations and common understanding of their formal properties. However, they have their own order of preferences and own functions for aggregating arguments. The inconsistency budget of each agent emerges from the combination of these functions. The personality of the agents can be encapsulated also by the above combination

Example 12. An agent $Judge = [MSV, (\oplus_L, \otimes_L, \ominus_L, \rightarrow_L), (\otimes_L^{1/2}, \otimes_L^2, \otimes_L^2)]$, by aggregating arguments under the Lukasiewicz semantics takes into consideration all the existing facts. It acts based on a hierarchy of values derived from the hierarchy of laws. In case of conflict, the most specific norm, which in general refers to exceptions, will be applied, then the argument from the most recent case (in case based law) or most recent norm (in norm based law). Afterward, the fuzziness of some linguistic terms from the law, will be considered in the decision.

For modeling practical scenarios we follow the steps.
1. Identify the relevant concepts and their attacking and supporting relations.
2. Define the membership functions.
3. Formalize the FRA: define the class hierarchy, define the role hierarchy, define the role properties, define the membership of arguments to their concepts, define the known strengths of the attacking and supporting relations among arguments.
4. Build the argumentation network
5. The agents start reasoning on the knowledge within the FRA based on their own preference schemes and aggregating functions. The reasoning process includes: i) identify semantic inconsistencies, ii) identify indirect attacks and supports, iii) reduce the argumentation network by composing the fuzzy relations, iv) aggregate arguments, v) compute the defeat status based on the active preference scheme.

5 A Case for Food Supply Chains

ISO 22000 is a recent standard designed to ensure safe food supply chains worldwide. Its main component is the HACCP (Hazard Analysis at Critical Control Points) system, which is a preventive method of securing food safety. Its objectives are the identification of consumer safety hazards that can occur in the supply chain and the establishment of a control process to guarantee a safer product.

5.1 Technology drivers for HACCP

HACCP is based on the following steps and principles [10]. In the first step, the business entities within the supply chain determine the *food safety hazards* and identify the preventive measures for controlling them. Then, the *critical control points* (CCP) are identified. They represent point steps in a food process at which a specific action can be applied in order to prevent a safety hazard, or to reduce it to an acceptable level. Afterward, *critical limits* are established, representing criteria which separate acceptability from unacceptability. Criteria often used include measurements of time, temperature, moisture level, pH, Aw, available chlorine, and sensory parameters such as visual appearance and texture. Critical limits must be specified for each CCP and they should be justified. A *monitor process* is followed by the *establishment of critical actions* in order to deal with deviations when they occur. Then procedures for verifications are needed to confirm that the HACCP system is working effectively. Finally, the documentation is needed to encapsulate justifications for all the decisions which have been taken. The main goal of the standard is to build confidence between suppliers and customers. It demands that business entities follow specific well-documented procedures, in which the quality of the items should be demonstrated by different types of justifications, and not only by attaching a quality label to the product. The technical requirements for building an HACCP system lay around the need to integrate support for argumentative debates.

Structured Argumentation. The technical support for argumentation is needed during the HACCP development for various tasks.

Justifying hazards. The HACCP standard explicitly requests that arguments pro and against should be provided in order to justify all decisions to classify hazards as critical or not critical ($Hazard = \exists\, hasJustification.Argument$). Based on the above definition in DL, the reasoner can check that a justification is attached to both significant or not significant hazard.

Justifying control options. For each hazard which is considered significant, a control measure should be defined ($SignificantHazard = Hazard \sqcap hasControlMeasure.\top$). The absence of the control measure is signaled as an inconsistency by the reasoner. The advantages and disadvantages of each available option should be backed by supporting arguments, respectively counter-arguments.

Justifying associated critical limits. The recommended sources of information for justifying the chosen critical limits are: norms, experts, scientific publications, or experimental studies. The rationale and the reference material should become part of the HAACP plan [10].

Ontological knowledge. When implementing the HACCP standard, the human experts need ontological knowledge in the following activities.

Hazard identification. The user can query hazard ontologies and their possible connections with ingredients and processing steps. Also, food and pathogen ontologies may be used to compare different risks which may stem from the production system.

Automatic verification of the safety conditions. Having formal descriptions about what a safety device, process, or service represent (encapsulated as TBoxes) and by having the current situation (encapsulated as ABoxes) the system can automatically point out possible contradictions with the norms in use.

Fuzzy Reasoning. It is used as a tool for qualitatively assessing during the following activities:

Assessment of critical control points. For each step of the production process, one should decide whether that stage will be a CCP. The decision depends on the hazard possibility of occurrence (terms such as rarely, often, sometimes, always are used in practice by the experts) and on its severity (usually assessed as low, medium, high).

Supply chain integration. An important source of hazards appears when receiving the input items. Depending on the potential risk, the company should decide to rely on the information from the product label or to conduct its own measurements of the product characteristics. This qualitative decision is based on fuzzy assessments. Also, the feedback received from the buyers, representing their preferences and perceptions is fuzzy. The costumers subjective evaluation can refer to attributes such as: color, smell, taste. Moreover, the company decides if it is able to deal with all the identified hazards or to outsource this task. For instance, the presence of rodents, insects, birds or other pests is unacceptable. The hazards are related both to the direct effects of these pests, and to the risks coming from the substances used to eliminate them. A good option is to contract a specialized company to handle these hazards.

Process adjustment. Actions need to be taken to bring the CCP under control before the critical limit is exceeded. The point where the operators take such action is called the operating limit. The process should be adjusted when the operating limit is reached to avoid violating the critical limit.

Modeling fuzzy critical limits. Consider some microbiological data. One rule can say: *"The product is safe if it is kept no longer than 48 hours at a temperature below* 10^0C*"*. What happens if the product is kept 47 hours at the temperature of 9^0C ? Is it safer comparing with the situation in which it is kept for 1 hour at a temperature of 12^0C? According to the above rule, the second item is not considered safe. As the alteration of product features is gradual, fuzzy membership functions being able to model these cases.

5.2 An HACCP Scenario

The framework is exemplified by for a cooked shrimp company. The control measure has emerged after an argumentation process.
The first step is two identify the main concepts and the relations among them. From figure 4, we have three options, each supported by its advantages and attacked by the disadvantages that it brings (figure 5, step 1). In the second step the agents should agree on the fuzzy membership functions for each concept in the domain. In the figure 5 three such membership functions are shown. The *ExpensiveToMonitor* concept is defined as a trapezoidal number $ExpensiveToMonitor = \exists\, hasCost.trapezoidal(10, 20, 30, 30)$ Consider a particular limit of pathogens d_1 which has the estimated cost at least 18 to be validated. The degree of membership to the concept *ExpensiveToMonitor* will be 0.8 ($d_1 : ExpensiveToMonitor \geq 0.8$), reflected in the strength of the attack between d_1 and the first option o_1. We consider the agents E_1, E_2 (decision makers) needing to be involved with other agents E_3, B_1 (consultants) in the process.

$$E_1 = [M, (\oplus_G, \otimes_G, \ominus_L, \rightarrow_G), (\otimes_G^{1/2}, \otimes_G^2, \otimes_G^2)]$$
$$E_2 = [MS, (\oplus_L, \otimes_L, \ominus_L, \rightarrow_L), (\otimes_L^{1/2}, \otimes_L^2, \otimes_L^2)]$$

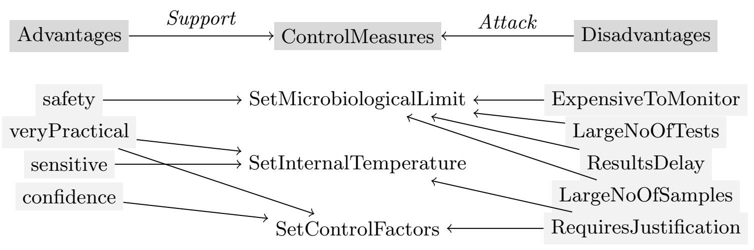
Agent	HACCP Plan: Justifying control measures
E_1	The first option for the control measure is to set a microbiological limit, under which the product is considered safe. This direct method minimizes error measurements, but I admit the monitoring process is expensive.
E_2	Several tests are necessary to determine critical limits derivations and samples may need to be large to be meaningful.
E_3	Moreover, the results are obtained in several days.
E_2	The second option is to set a minimum internal temperature at which the pathogens are destroyed. The method is practical and more sensitive.
E_1	But justification is needed to validate the chosen temperature value.
E_3	The third option is to control the factors that affect the internal temperature of the product (oil, thickness of the pane, or cooking time).
E_1	The method requires justifications between these limits and the internal temperature of food.
E_3	I agree. Nevertheless, it is very practical and it increases confidence in the measurements.
B_1	The business policy encourages practical and reliable solutions.

Fig. 4. Arguing for the adequate control measures in an HACCP plan.

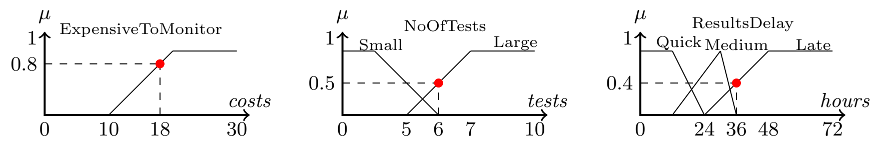
They start by aggregating the direct attack and support relations. The aggregation of d_1, d_2, d_3 and d_4 gives an attack strength of $max(0.8, 0.5, 0.4, 0.7) = 0.8$ for the first option o_1, under the Gödel semantics (the agent E_1) and $min(min(min(0.8 + 0.5), 1) + 0.4), 1) + 0.7, 1) = 1$ (figure 6). Because both arguments a_2 and a_3 support the second option o_2 there strengths are aggregated, giving $max(0.55, 0.49) = 0.55$ for the agent E_1, respectively $min(0.55 + 0.49, 1) = 1$ for the agent E_2. Next, the indirect relations are taken into consideration. By composing the supporting relation *Enc* with *Support* we get an indirect attack between b_1 and o_2 of strength $min(0.9, 0.49)^2 = 0.24$ under Logic semantics and $max(0.9 + 0.49 - 1, 0)^2 = 0.15$. Hence, $(b_1, o_1) : Enc(\otimes_G)^2 \geq 0.24$ for E_1 and $(b_1, o_1) : Enc(\otimes_L)^2 \geq 0.15$. Note that b_1 indirectly supports the option o_3 through two intermediary nodes a_2 and a_4. The amount of indirect support provided by b_1 for o_3 is $max(min(0.9, 0.55)^2, min(0.9, 0.8)^2) = 0.64$ from the E_1's perspective. The agent E_2 computes the degree of support from b_1 towards o_3 as $min(max(0.9+0.55-1, 0)^2 + max(0.9+0.7-1, 0)^2, 1) = 0.56$. The bottom part of figure 6 depicts the aggregation of direct and indirect relations. The compound concept $a_{23}b_1$ supports the option o_2 with at least $max(0.55, 0.24) = 0.55$ from the E_1 viewpoint and with $min(1 + 0.15, 0) = 1$ from that of the expert E_2. The support given by $a_{24}b_1$ to the third option equals $max(0.64, 0.8) = 0.8$, respectively $min(1 + 0.49, 1) = 1$.

The attack on the first option is stronger then its support, from both perspectives E_1 and E_2. Hence, an agreement between the agent E_1 and E_2 exists to eliminate this option. Given the above information, the agent E_2 equally accepts the options o_3 and o_2. The degree of support is 1 and the attack 0.6 in both cases. On the other hand, the agent E_1 rejects the option o_2 but accepts option o_3. Consequently, the last control measure gets the support from both parties.

1. Identifying concepts and attacking and supporting relations.



2. Define the membership functions



4. Formalizing the FRA

(A=Advantages, D=Disadvantages, O=ControlMeasures, B=BusinessPolicy, Enc=Encourage)

$\mathcal{A} = \prec \{a_1 : Safety \geq 0.6,\ a_2 : Practical \geq 0.7,\ a_2 : very(Practical) \geq 0.49,\ a_3 : Sensitive \geq 0.3,$
$a_4 : Confidence \geq 0.5,\ d_1 : ExpensiveToMonitor \geq 0.8,\ d_2 : LargeNoOfTests \geq 0.5,$
$d_3 : ResultsDelay \geq 0.4,\ d_4 : LargeNoOfSamples \leq 0.8,$
$d_5 : RequiresJustification,\ b_1 : B,\ o_1 : O,\ o_2 : O,\ o_3 : O\} \succ$

$\mathcal{T} = \{A, D, O, B,\ Safeness \sqsubseteq A,\ Practical \sqsubseteq A,\ Sensitive \sqsubseteq A,\ Confidence \sqsubseteq A,$
$ExpensiveToMonitor \sqsubseteq D,\ LargeNoOfTests \sqsubseteq D,\ ResultsDelay \sqsubseteq D,\ LargeNoOfSamples \sqsubseteq D,$
$RequiresJustification \sqsubseteq D, (A, O) : \bar{\mathrm{A}}), (A, O) : \bar{\mathrm{S}})\}$

$\mathcal{R} = \mathcal{R}^k \cup \{Enc \sqsubseteq \bar{\mathrm{S}}\}$

3. Building the Argumentation Network

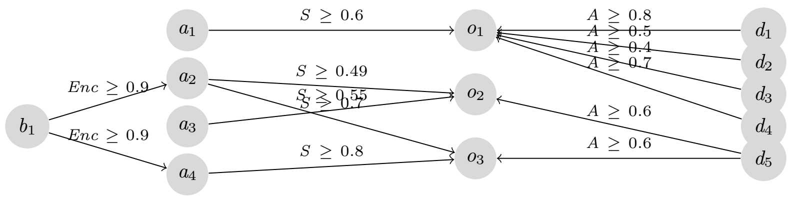


Fig. 5. A FRA for Justifying Control Measures

$E_1 = [M, (\oplus_G, \otimes_G, \ominus_L, \rightarrow_G), (\otimes_G^{1/2}, \otimes_G^2, \otimes_G^2)]$. $E_2 = [MS, (\oplus_L, \otimes_L, \ominus_L, \rightarrow_L), (\otimes_L^{1/2}, \otimes_L^2, \otimes_L^2)]$.

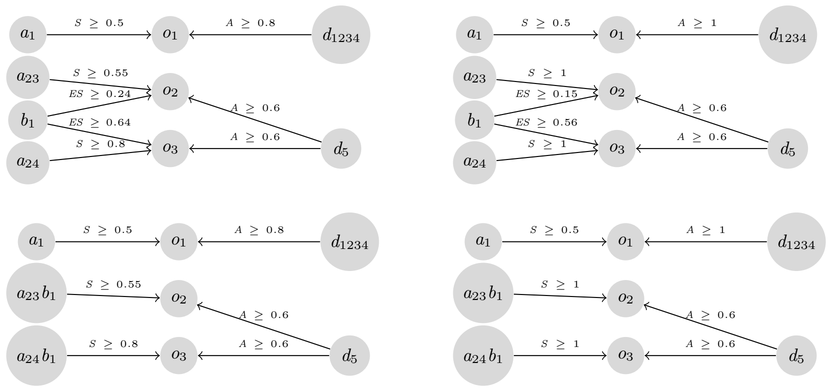


Fig. 6. Argumentative agents

6 Discussion and Related Work

Arguments supporting both a consequent and its negation co-exist in the knowledge base. To overcome this drawback, *weighted argument systems* (WAS) have been introduced, with the notion of *inconsistency budget* [11] used to characterize how much inconsistency one is prepared to tolerate in an argumentation base. A FRA framework is a particular instance of a WAS, where the degree of inconsistency is accommodated by the semantics of fuzzy reasoning.

Other approaches have investigated imprecise argumentation [12–14]. The fuzzy argumentation framework [12] is an extension of the classical Dung model, while in our approach, the fuzzy component is meant to help software agents to exploit the real arguments conveyed by humans. Compared to the defeasible logic approach [13, 14], where the ontological knowledge is embedded in the program, we have been interested in having a clear, separate representation of the ontology, allowing for transparency.

Rahwan and Banihashemi address [15] the idea of modeling argument structures in OWL, where arguments are represented in the Argument Interchange Format ontology (AIF), a current proposal for a standard to represent arguments. Meta-argumentation [16, 17] is supported by the AIF approach: one can apply a preference on preference, attack a support, or support a preference. Meta-argumentation can be handled in FRA indirectly by defining new concepts and applying roles on them. For instance, the preference relation P applied to the argument a over b can be encapsulated as the concept *a preferred to b*, which makes possible to apply further attack or support roles on it. If one wants to challenge the degree α of membership of an element a to the concept A, the new concept $A_{\geq\alpha}$ can be defined and the attack should be applied on it. A hierarchy of Dung frameworks is proposed in [18], in which level n arguments refer to level n-1 arguments, and conflict based relations and preferences between level n-1 arguments. Arguing hierarchically is handled by navigating through the concepts which are subject of dispute and which can be organized hierarchically based on description logic.

The role of ontologies to resolve conflicts among arguments based on the specificity principle appears also in [19]. The existing work has focused on concept properties only, and do not exploit the formal properties of the attack relations. Our formalism based on FDL contributes to the current vision of developing the infrastructure for the World Wide Argument Web.

7 Conclusions

The fuzzy-based approach presented in this paper makes a step towards practical applications, a fuzzy-based argumentative system being cognitively less demanding for the decision makers. Real argumentative debates implies other relations and concepts, not only attacking and support roles or arguments. This additional domain knowledge can be easily integrated into a FRA framework. The main contribution comes from the introduction of different types of attack and support roles with a specific semantics given by their formal properties, with no need to invent a new mechanism to compute the strength of the attack. We have just used the technical instrumentation provided by fuzzy logic for computing the status of argument. We advocate the merging of argumentation theory with semantic technologies, which leads to the possibility to reuse the argumentation bases among multi-agent systems. The preference schemes have already proved their success as conflict resolution strategies in expert systems. Techniques from expert systems can be used to validate them for specific domains (by checking the accuracy of the FRA frameworks against the expert evaluation of arguments).

One drawback is that we assume a common ontology of attacking and supporting roles. In the presented scenario, this limitation is partial overcome by the fact that HACCP represents a standard, which implies a common description of concepts and procedures. Also, the implication of fuzzy composition to indirect attacks needs a deeper investigation to validate the proposed semantics. Third, a methodology for modeling practical applications with argumentation theory is needed.

Acknowledgments

We are grateful to the anonymous reviewers for the very useful comments. Part of this work was supported by the grant ID_170/672 from the National Research Council of the Romanian Ministry for Education and Research.

References

1. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artif. Intell. **171**(10-15) (2007) 619–641
2. Rahwan, I., Zablith, F., Reed, C.: Laying the foundations for a World Wide Argument Web. Artif. Intell. **171**(10-15) (2007) 897–921
3. Hunter, A.: Real arguments are approximate arguments. In: AAAI. (2007) 66–71
4. Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy description logic reasoner. In: FUZZ-08, IEEE Computer Society (2008) 923–930
5. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77** (1995) 321–357
6. Madakkatel, M.I., Rahwan, I., Bonnefon, J.F., Awan, R.N., Abdallah, S.: Formal argumentation and human reasoning: The case of reinstatement. In: Proceedings of the AAAI Fall Symposium on The Uses of Computational Argumentation, Washington DC, USA. (2009)
7. Pollock, J.: Defeasible reasoning with variable degrees of justification. Artificial Intelligence **133** (2002) 233–282
8. Straccia, U.: A fuzzy description logic for the semantic web. In Sanchez, E., ed.: Capturing Intelligence: Fuzzy Logic and the Semantic Web, Elsvier (2006)
9. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research **14** (2001) 137–166
10. Food, of the United Nations World Health Organization, A.O.: Codex Alimentarus. (1997)
11. Dunne, P., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Inconsistency tolerance in weighted argument systems. In: AAMAS. (2009) 851–858
12. Janssen, J., De Cock, M., Vermier, D.: Fuzzy argumentation frameworks. In: IPMU. (2008) 513–520
13. 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) (2008) 1208–1228
14. Gomez, S.A., Chesñevar, C.I., Ricardo, G.: Reasoning with inconsistent ontologies through argumentation. Applied Artificial Intelligence **24** (2010) 102–148
15. Rahwan, I., Banihashemi, B.: Arguments in OWL: A progress report. In: COMMA. (2008) 297–310
16. Boella, G., Gabbay, D.M., van der Torre, L., Villata, S.: Meta-argumentation modelling i: Methodology and techniques. Studia Logica **93**(2-3) (2009) 297–355
17. Modgil, S.: Reasoning about preferences in argumentation frameworks. Artif. Intell. **173**(9-10) (2009) 901–934
18. Modgil, S.: Hierarchical argumentation. In: JELIA. (2006) 319–332
19. Bench-Capon, T.J.M., Gordon, T.F.: Isomorphism and argumentation. In: ICAIL. (2009) 11–20

Dynamic Argumentation in Abstract Dialogue Frameworks [⋆]

M. Julieta Marcos, Marcelo A. Falappa, and Guillermo R. Simari

National Council of Scientific and Technical Research (CONICET)
Artificial Intelligence Research & Development Laboratory (LIDIA)
Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
{mjm, mfalappa, grs}@cs.uns.edu.ar

Abstract. In this work we present a formal model for collaborative argumentation-based dialogues by combining an abstract dialogue framework with a formalism for dynamic argumentation. The proposed model allows any number of agents to interchange and jointly build arguments in order to decide the justification status of a given claim. The model is customizable in several aspects: the argument attack relation and acceptability semantics, the notion of relevance of contributions, and also the degree of collaboration are selectable. Important properties are ensured such as dialogue progress step by step, completeness of the sequence of steps, and termination. Under the higher degree of collaboration, the dialogue constitutes a sound and complete distributed argumentation process.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Coherence and coordination

General Terms

Theory, Design

Keywords

Collective intelligence, Dialogue, Argumentation

1 Introduction and Motivation

Multi-agent systems (MAS) provide solutions to problems in terms of autonomous interactive components (agents). A *dialogue* is a kind of interaction in which a sequence of messages, over the same topic, is exchanged among a group of agents, with the purpose of jointly drawing some sort of conclusion. There is a subset of dialogues, which we call *collaborative*, in which the agents are willing to share any relevant knowledge to the topic at issue, having no other ambition than achieving the right conclusion on the basis of all the information they have.

Argumentation-based dialogues usually consist of interchanging arguments for and against certain claim. Mostly in the literature, these dialogues are held between two

⋆ This research is partially supported by Sec. Gral. de Ciencia y Tecnología (Univ. Nac. del Sur), CONICET and Agencia Nac. de Prom. Científica y Técnica (ANPCyT).

agents, one of them putting the arguments 'for' and the other putting the arguments 'against'. In order to achieve collaborative (in the sense described above) behavior, all the participants should contribute with both kinds of arguments, and also they should be able to jointly build new arguments. Even as part of non-collaborative dialogues (*e.g.* persuasion) it may be useful to build arguments in conjunction.

Classical *abstract argumentation* [1] assumes a static set of already built arguments, resulting insufficient for modeling collaborative dialogues. The set of arguments involved in a dialogue is, in contrast, dynamic: new arguments jointly constructed by the agents may arise, and also arguments may be invalidated (note this is not the same as defeated) at the light of new information. The argument construction step cannot be performed separately from the dialogue.

Recently, a *dynamic abstract argumentation framework (DAF)* has been proposed by Rotstein *et al.* [2], which extends the work done on acceptability of arguments, by taking into consideration their construction and their validity with respect to a varying set of evidence. This approach results, hence, very suitable for the modeling of collaborative dialogues. The main elements of the DAF are summarized in Sect. 2.

In [3] we have defined an *abstract dialogue framework ($\mathfrak{DF}$)* together with a set of *collaborative semantics* which characterize different levels of collaboration in dialogues, in terms of a given reasoning model and a given notion for the relevance of contributions. Under certain natural conditions, the proposed semantics ensure important properties of collaborative dialogues, such as termination and outcome-determinism.

The aim of this work is to show how the abstract dialogue framework and semantics [3] can be applied to dynamic argumentation [2]. As will be seen, the agents will interchange both arguments (in Rotstein's sense) and evidence, achieving the joint construction of arguments in the usual sense. A particular framework for argumentation-based dialogues will be obtained, which inherits the semantics and properties defined for the abstract framework. In sections 4 through 6, we will reintroduce the abstract concepts that constitute the $\mathfrak{DF}$ showing how they can be instantiated in terms of the DAF.

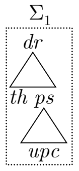
2 Background

Next we summarize an abstract argumentation framework capable of dealing with dynamics through the consideration of a varying set of evidence [2]. Depending on a particular situation (given by the content of the set of evidence), an instance of the framework will be determined, in which some arguments hold and others do not.

The formalization is coherent with classical abstractions [1], however arguments play a smaller role: they are aggregated in structures. These argumental structures can be thought as if they were arguments (in the usual sense), but they will not always guarantee their actual achievement of the claim.

A language $\mathbf{L}$ will be assumed for the representation of evidence, premisses, and claims. An *argument* $\mathcal{A}$ is a pair $\langle\{s_1 \ldots s_n\}, \delta\rangle$ consisting of a consistent set of *premisses*, noted $\texttt{supp}(\mathcal{A})$, and a *claim*, noted $\texttt{cl}(\mathcal{A})$. These basic premisses are considered the argument *support*. A *supporting argument* is one that claims for the premise of another argument. The language of all the possible arguments built from $\mathbf{L}$ will be noted $\mathbf{L}_{\mathrm{A}}$. Consider for instance the argument $\mathcal{A}_1 = \langle\{th, ps\}, dr\rangle$ which assumes a route to be dangerous because there are known thieves in that area and the security there is poor. Consider also the supporting argument $\mathcal{A}_2 = \langle \{upc\}, ps \rangle$ saying that underpaid cops might provide poor security [2].

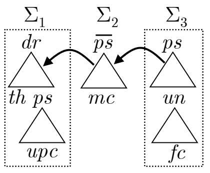
An argument is *coherent*, wrt. a set of evidence E, if its claim does not contradict, nor coincides with, any evidence in E. Then, a coherent argument is *active* if each of its premisses is either evidence or a claim of an active argument. Following the above example, the argument $\mathcal{A}_1$ is active wrt. the set $\{th, ps\}$ and also wrt. the set $\{th, upc\}$, but it is not wrt. $\{th\}$ nor $\{th, ps, \overline{dr}\}$. Inactive arguments will be depicted in gray.



An *argumental structure* (*structure* for short), for a claim δ, is a tree of arguments where the root argument claims for δ, and every non-root argument supports the parent through a different premise (note there may be unsupported premisses). The arguments $\mathcal{A}_1$ and $\mathcal{A}_2$ from the previous example constitute an argumental structure Σ_1 , depicted on the right, for the claim 'dr'. Structures are depicted as dashed boxes. The box will be omitted when the structure consists of a unique argument.

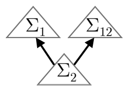
Further constraints over structures (yielding *well-formed* structures) are imposed in order to ensure a sensible reasoning chain. These avoid arguments attacking each other within a structure, infinite structures, and heterogeneous support for a premise throughout a structure (see [2] for details). A well-formed argumental structure is *active*, wrt. a set of evidence E, if every argument in it is coherent wrt. E, and every unsupported premise is evidence in E. For instance, the previous structure Σ_1 claiming a route as being dangerous, is active wrt. the set $\{th, upc\}$, but not wrt. $\{th\}$ because the premise 'upc' is not evidence nor the claim of another argument in the structure. Neither is Σ_1 active wrt. $\{th, ps\}$ since $\mathcal{A}_2$ would not be coherent: its claim 'ps' is redundant wrt. to the evidence. Inactive structures will be depicted in gray.

The *dynamic argumentation framework (DAF)* we will use is a pair $\langle \mathbf{E}, (\mathbf{W}, \mathbf{R}) \rangle$ composed by a consistent set $\mathbf{E}$ of *evidence*, a *working set of arguments* $\mathbf{W}$, and an *attack relation* $\mathbf{R} \subseteq \mathbf{L}_\mathrm{A} \times \mathbf{L}_\mathrm{A}$ [1] between arguments. We restrict the attacks to pairs of arguments with contradictory claims, and at least in one direction. That is, for every pair of arguments $\mathcal{A}_1$ and $\mathcal{A}_2$ whose claims are in contradiction, at least one of $(\mathcal{A}_1, \mathcal{A}_2)$ or $(\mathcal{A}_2, \mathcal{A}_1)$ will belong to $\mathbf{R}$. Contradictory sentences will be noted as a and $\overline{a}$.



The notion of attack over arguments has a direct correlation to argumental structures: a structure *attacks* another if the root argument of the first attacks any argument of the second. Consider, for instance, the argument $\mathcal{A}_3 = \langle \{mc\}, \overline{ps} \rangle$ which assumes the security to be good because there are many cops in the area. Consider also the arguments $\mathcal{A}_4 = \langle \{un\}, ps \rangle$ and $\mathcal{A}_5 = \langle \{fc\}, un \rangle$ saying that foreign cops might be unacquainted with the place, giving the idea of poor security [2]. Assume that $\mathcal{A}_3$ attacks $\mathcal{A}_2$ and $\mathcal{A}_4$ attacks $\mathcal{A}_3$. Then, the structure Σ_1 mentioned earlier is attacked by the structure Σ_2 (composed by $\mathcal{A}_3$), which is in turn attacked by the structure Σ_3 (composed by $\mathcal{A}_4$ and $\mathcal{A}_5$).

[1] In [2], the attack relation is defined over the working set of arguments $\mathbf{W}$. Since a unified policy for comparing arguments from different agents is needed in a collaborative dialogue setting, here we introduce a slight variation generalizing the attack relation over the universal set of arguments $\mathbf{L}_\mathrm{A}$.



At any moment, the *active instance* of a DAF is a pair $(\mathbb{S}, \mathbb{R})$, where $\mathbb{S}$ is the set of active argumental structures, and $\mathbb{R}$ is the resulting attack relation between them. This is equivalent to Dung's definition of argumentation framework [1]. Therefore, classic argumentation semantics can be applied to the active instance. From the previous examples, if we consider the set of evidence $\{th, upc, mc\}$ then the active instance, depicted on the right, consists of: the structures Σ_1 and Σ_2 mentioned earlier, and also the structure Σ_{12} composed only by argument $\mathcal{A}_2$. Note that Σ_3 is not active, and hence does not belong to the active instance. Picking grounded semantics, for instance, the only accepted structure would be Σ_2.

In this work we will assume unique-extension semantics. The arguments belonging to the extension, along with their claims, will be considered *justified* from the DAF, as well as the whole evidence set. From the previous example, the claim '$\overline{ps}$' would be justified. Multiple-extension semantics could also be used, expanding the set of possible 'justification statuses' of a claim (*e.g.* 'justified', 'not-justified', or 'undecided').

3 Informal Requirements for Collaborative Dialogue Models

We believe that an ideal collaborative behavior of dialogues should satisfy the following, informally specified, requirements:

R$_1$**:** All the **relevant** information is exposed in the dialogue.
R$_2$**:** The exchange of **irrelevant** information is avoided.
R$_3$**:** The final conclusion **follows** from all what has been said.

On that basis, we will conduct our analysis of collaborative dialogue behavior in terms of two abstract elements: a *reasoning model* and a *relevance notion* [2], assuming that the former gives a formal meaning to the word *follows*, and the latter to the word *relevant*. Both elements are domain-dependent and, as we shall see, they are not unattached concepts. It is important to mention that the relevance notion is assumed to work in a context of *complete information* (this will be clarified later).

We believe that the achievement of R$_1$-R$_3$ should lead to achieving other important requirements, listed below. Later in this work we will state the conditions under which this hypothesis actually holds.

R$_4$**:** The dialogue should always end.
R$_5$**:** Once the dialogue ends, if the agents added all their still private information, and reasoned from there, the previously drawn conclusions should not change.

In the task of simultaneously achieving requirements R$_1$ and R$_2$, in the context of a distributed MAS, a non-trivial problem arises: relevant information distributed in such a way that none of the parts is relevant by itself. For instance, considering the DAF of Sect. 2, an agent may have an argument for a certain claim but the activating evidence

[2] The term *relevance* appears in many research areas: *epistemology*, *belief revision*, *economics*, *information retrieval*, etc. In this work we intend to use it in its most general sense, which may be closer to the epistemic one: *pertinence in relation to a given question*, but it should not be tied to any particular interpretation, except for concrete examples given in this work.

resides in a different agent. This, in principle threatens R$_1$ since the whole contribution would be left unseen. Besides, any attempt to detect these 'non-self-relevant' parts threatens R$_2$ due to the risk of being mistaken. This could happen for instance, following with the previous example, if the argument is exposed but the activating evidence does not actually exist. There is a tradeoff between requirements R$_1$ and R$_2$.

Because of the nature of collaborative dialogues, we believe R$_1$ may be mandatory in many application domains, and hence we will seek solutions which achieve it, even at the expense of relegating R$_2$ a bit. As will be seen later in Sect. 6, the basic idea will be to develop a new relevance notion (which will be called a *potential relevance notion*) able to detect parts of distributed relevant contributions (under the original notion).

4 The Dialogue Framework

Three languages are assumed to be involved in a dialogue: the *Knowledge Representation Language* $\mathcal{L}$ for expressing the information exchanged by the agents, the *Topic Language* $\mathcal{L}_\mathrm{T}$ for expressing the *topic* that gives rise to the dialogue, and the *Outcome Language* $\mathcal{L}_\mathrm{O}$ for expressing the final conclusion (or *outcome*). Also assumed is a language $\mathcal{L}_\mathrm{I}$ for agent identifiers. As usual, a *dialogue* consists of a topic, a sequence of *moves*, and an outcome. In each move an agent makes a *contribution* (exposes a set of knowledge). This is a *public view of dialogue*: agents' private knowledge is not taken into account yet.

Definition 1 (Move). *A* move *is a pair* $\langle id, X\rangle$ *where* $id \in \mathcal{L}_\mathrm{I}$ *is the identifier of the* speaker*, and* $X \subseteq \mathcal{L}$ *is her* contribution*.*

Definition 2 (Dialogue). *A* dialogue *is a tuple* $\langle t, \langle m_j\rangle, o\rangle$ *where* $t \in \mathcal{L}_\mathrm{T}$ *is the* dialogue topic*,* $\langle m_j\rangle$ *is a sequence of moves, and* $o \in \mathcal{L}_\mathrm{O}$ *is the* dialogue outcome*.*

As will be seen in short, in the argumentative approach based on the DAF, the agents will expose arguments and evidence, topics will correspond to claims, and dialogue outcomes might be Yes (justified) or No (not justified).

As anticipated in Sec. 3, we will study the dialogue behavior in terms of two abstract concepts: relevance and reasoning. To that end, an *Abstract Dialogue Framework ($\mathfrak{DF}$)* is introduced, whose aim is to provide an environment for dialogues to take place, and which includes: the languages involved in the dialogue, a set of participating agents, a relevance notion and a reasoning model. An *agent* is represented by an *identifier* and a private *knowledge base (kb)*, providing in this way a *complete view* of dialogues.

Definition 3 (Agent). *An* agent *is a pair* $\langle id, K\rangle$*, noted* K_{id}*, where* $K \subseteq \mathcal{L}$ *is a private finite knowledge base, and* $id \in \mathcal{L}_\mathrm{I}$ *is an agent identifier.*

A *relevance notion* is a criterion for determining, given certain already known information and a topic, whether it would be relevant to add certain other information (*i.e.,* to make a contribution). We emphasize that this criterion works under an assumption of *complete information*, to be contrasted with the situation of a dialogue where each agent is unaware of the private knowledge of the others. This issue will be revisited in Sec. 5. Finally, a *reasoning model* will be understood as a mechanism for drawing a conclusion about a topic, on the basis of an individual knowledge base. The argumentation-based reasoning model, for instance, will determine the justification status of a claim from a given set of evidence and arguments.

Definition 4 (Abstract Dialogue Framework). *An* abstract dialogue framework ($\mathfrak{D}\mathfrak{F}$) *is a tuple* $\langle \mathcal{L}, \mathcal{L}_\mathrm{T}, \mathcal{L}_\mathrm{O}, \mathcal{L}_\mathrm{I}, \mathcal{R}, \Phi, Ag \rangle$ *where* $\mathcal{L}$, $\mathcal{L}_\mathrm{T}$, $\mathcal{L}_\mathrm{O}$ *and* $\mathcal{L}_\mathrm{I}$ *are the languages involved in the dialogue,* Ag *is a finite set of agents,* $\mathcal{R} \subseteq 2^{\mathcal{L}} \times 2^{\mathcal{L}} \times \mathcal{L}_\mathrm{T}$ *is a* relevance notion*, and* $\Phi : 2^{\mathcal{L}} \times \mathcal{L}_\mathrm{T} \Rightarrow \mathcal{L}_\mathrm{O}$ *is a* reasoning model. *The brief notation* $\langle \mathcal{R}, \Phi, Ag \rangle$ *will be also used.*

Notation 1. *If* $(X, S, t) \in \mathcal{R}$*, we say that* X *is a* t-relevant contribution to S under $\mathcal{R}$*, and we note it* $X\mathcal{R}_tS$*. When it is clear what relevance notion is being used, we just say that* X *is a* t*-relevant contribution to* S*. For individual sentences* α *in* $\mathcal{L}$*, we also use the simpler notation* $\alpha\mathcal{R}_tS$ *meaning that* $\{\alpha\}\mathcal{R}_tS$*.*

Throughout this work we will refer to the partially instantiated $\mathfrak{D}\mathfrak{F}$ $\mathfrak{F}^{\mathrm{ar}} = \langle \mathbf{L} \cup \mathbf{L}_\mathrm{A}, \mathbf{L}, \{\mathrm{Yes}, \mathrm{No}\}, \mathcal{L}_\mathrm{I}, \mathcal{R}_\delta, \Psi, Ag \rangle$. The languages $\mathbf{L}$ and $\mathbf{L}_\mathrm{A}$ are the ones of Sect. 2. Hence, in this argumentation-based dialogue framework, the knowledge representation language consists of both evidence and arguments, topics correspond to claims, and outcomes might be Yes or No. As mentioned before, the reasoning model Ψ determines the justification status of a claim from a given set of evidence and arguments. That is, $\Psi(K, \delta) = \mathrm{Yes}$ if, and only if, the claim δ is justified (under a certain argumentation semantics $\mathbf{S}$) from the DAF $\langle \mathbf{E}, (\mathbf{W}, \mathbf{R}) \rangle$, with $\mathbf{E} \cup \mathbf{W} = K$ and $\mathbf{R}$ a certain attack relation between arguments. We will sometimes use the more specific notation $\mathfrak{F}^{\mathrm{ar}}(\mathbf{S}, \mathbf{R})$. Particularly, the notation $\mathfrak{F}^{\mathrm{ar}}(\mathbf{G}, \mathbf{R})$ will be used for the instantiation with grounded semantics [1]. For the aim of simplicity, we will assume that the union of the evidence in all knowledge bases is consistent.

There are two different sets of knowledge involved in a dialogue: the *private knowledge* which is the union of the agents' knowledge bases, and the *public knowledge* which is the union of all the contributions already made, up to certain step. The former is a static set, whereas the latter grows as the dialogue progresses.

Definition 5 (Public Knowledge). *Let* d *be a dialogue consisting of a sequence* $\big\langle \langle id_1, X_1 \rangle \ldots \langle id_m, X_m \rangle \big\rangle$ *of moves. The* public knowledge *associated to* d *at step* j *(*$j \leq m$*) is the union of the first* j *contributions of the sequence and is noted* $\mathbf{PU}^j_\mathrm{d}$ *(*$\mathbf{PU}^j_\mathrm{d} = X_1 \cup \cdots \cup X_j$*).*

Definition 6 (Private Knowledge). *Let* $\mathfrak{F}$ *be a* $\mathfrak{D}\mathfrak{F}$ *including a set* Ag *of agents. The* private knowledge *associated to* $\mathfrak{F}$ *(and to any admissible dialogue under* $\mathfrak{F}$*) is the union of the knowledge bases of the agents in* Ag*, and is noted* $\mathbf{PR}_\mathfrak{F}$ *(*$\mathbf{PR}_\mathfrak{F} = \bigcup_{K_{id} \in Ag} K$*).*

In order to restrict agents' contributions to be subsets of their private knowledge, we define next the set of *admissible dialogues* under a given $\mathfrak{D}\mathfrak{F}$.

Definition 7 (Admissible Dialogues). *Let* $\mathfrak{F} = \langle \mathcal{L}, \mathcal{L}_\mathrm{T}, \mathcal{L}_\mathrm{O}, \mathcal{L}_\mathrm{I}, \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{D}\mathfrak{F}$*,* $t \in \mathcal{L}_\mathrm{T}$ *and* $o \in \mathcal{L}_\mathrm{O}$*. A dialogue* $\big\langle t, \langle m_j \rangle, o \big\rangle$ *is* admissible *under* $\mathfrak{F}$ *if, and only if, for each move* $m = \langle id, X \rangle$ *in the sequence, there is an agent* $K_{id} \in Ag$ *such that* $X \subseteq K$*. The set of admissible dialogues under* $\mathfrak{F}$ *is noted* $\mathrm{d}(\mathfrak{F})$*.*

Remark 1. For any step j of any dialogue $\mathrm{d} \in \mathrm{d}(\mathfrak{F})$, it holds that $\mathbf{PU}_{\mathrm{d}}^{j} \subseteq \mathbf{PR}_{\mathfrak{F}}$.

Returning to the notions of relevance and reasoning, it was mentioned in Sec. 3 that these were not unattached concepts: a coherent dialogue must exhibit some connection between them. Assuming a contribution to be relevant whenever its addition alters the conclusion achieved by the reasoning model, as defined below, seems to be a natural connection.

Definition 8 (Natural Relevance Notion). *Let Φ be a reasoning model. The* natural relevance notion *associated to Φ is a relation $\mathcal{N}_t^{\Phi}$ such that: $X \mathcal{N}_t^{\Phi} S$ iff $\Phi(S, t) \neq \Phi(S \cup X, t)$. When $X \mathcal{N}_t^{\Phi} S$ we say that X is a* natural t-relevant contribution to S under Φ.

Hence, in the argumentative approach, the natural relevance notion $\mathcal{N}_{\delta}^{\Psi}$ detects the change of the "justification status" for a given claim. It will be seen later that this connection can be relaxed, *i.e.,* other relevance notions which are not exactly the natural one, might also be accepted. We distinguish the subclass of $\mathfrak{DF}$s in which the relevance notion is the natural one associated to the reasoning model. We refer to them as *Inquiry* [3] *Dialogue Frameworks ($\mathfrak{IDF}$)*, and the relevance notion is omitted in their formal specification.

Definition 9 (Inquiry Dialogue Framework). *An* Inquiry Dialogue Framework ($\mathfrak{IDF}$) *is a $\mathfrak{DF}$ $\langle \mathcal{R}_t, \Phi, Ag \rangle$ where $\mathcal{R}_t = \mathcal{N}_t^{\Phi}$. The brief notation $\langle \Phi, Ag \rangle$ will be used.*

Throughout this work we will refer to the partially instantiated $\mathfrak{IDF}$ $\mathfrak{I}^{\mathrm{ar}} = \langle \Psi, Ag \rangle$. As with $\mathfrak{DF}$s, we will also use the notation $\mathfrak{I}^{\mathrm{ar}}(\mathbf{S}, \mathbf{R})$ for specifying a particular argumentation semantics and a particular argument attack relation.

5 Utopian Collaborative Semantics

A *semantics* for a $\mathfrak{DF}$ is a subset of the admissible dialogues representing a particular dialogue behavior. We are interested in specifying which, from all the admissible dialogues under a given $\mathfrak{DF}$, have an acceptable collaborative behavior. In Sec. 3 we identified three requirements, R_1-R_3, to be ideally achieved by collaborative dialogue systems. In this section, we will define an *Utopian Collaborative Semantics* which gives a formal characterization of such ideal behavior. In order to translate requirements R_1-R_3 into a formal specification, some issues need to be considered first.

In particular, the notion of *relevant contribution* needs to be adjusted. On the one hand, there may be contributions which does not qualify as relevant but it would be adequate to allow. To understand this, it should be noticed that, since relevance notions are related to reasoning models, and reasoning models may be non-monotonic, then it is possible for a contribution to contain a relevant subset, without being relevant itself.

[3] The term *Inquiry* is inspired on the popularized typology of dialogues proposed in [4], since we believe that the natural relevance notion captures the essence of this type of interaction: *collaboration to answer some question*. However, the term will be used in a broader sense here, since nothing is assumed regarding the degree of knowledge of the participants.

For instance, in the context of the $\mathfrak{I}^{\mathrm{ar}}(\mathbf{G},\mathbf{R})$ framework, an active argumental structure Σ_1 would be a natural $\mathtt{cl}(\Sigma_1)$-relevant contribution to the empty set, but if we added an active structure Σ_2 attacking Σ_1, then it would not. The possibility of some other agent having, for instance, an active structure Σ_3 attacking Σ_2, explains why it would be useful to allow the whole contribution consisting of both Σ_1 and its attacker Σ_2 (and all the supporting evidence). In these cases, we say that the relevance notion fails to satisfy *left-monotonicity* and that the whole contribution is *weakly relevant* [4]. The formal definitions are given below.

Definition 10 (Left Monotonicity). *Let $\mathcal{R}_t$ be a relevance notion. We say that $\mathcal{R}_t$ satisfies* left monotonicity *iff the following condition holds: if $X\mathcal{R}_tS$ and $X \subseteq Y$ then $Y\mathcal{R}_tS$.*

Definition 11 (Weak Contribution). *Let $\mathcal{R}_t$ be a relevance notion. We say that X is a* weak t-relevant contribution *to S iff there exists $Y \subseteq X$ such that $Y\mathcal{R}_tS$.*

On the other hand, there may be contributions which qualify as relevant but they are not *purely* relevant. For example, the argument $\langle\{b\}, a\rangle$ together with the set of evidence $\{b, e\}$ constitute a natural 'a'-relevant contribution to the empty set, although the evidence 'e' is clearly irrelevant. These impure relevant contributions must be avoided in order to obey requirement R_2. For that purpose, *pure relevant contributions* impose a restriction over weak relevant ones, disallowing absolutely irrelevant sentences within them, as defined below.

Definition 12 (Pure Contribution). *Let $\mathcal{R}_t$ be a relevance notion, and X a weak t-relevant contribution to S. We say that X is a* pure t-relevant contribution *to S iff the following condition holds for all $\alpha \in X$: there exists $Y \subset X$ such that $\alpha\mathcal{R}_t(S \cup Y)$.*

Finally, it has been mentioned that the relevance notion works under an assumption of *complete information*, and thus it will be necessary to inspect the private knowledge of the others for determining the actual relevance of a given move. Now we are able to give a formal interpretation of requirements R_1-R_3 in terms of the $\mathfrak{DF}$ elements:

Definition 13 (Utopian Collaborative Semantics). *Let $\mathfrak{F} = \langle\mathcal{R}_t, \Phi, Ag\rangle$ be a $\mathfrak{DF}$. A dialogue $\mathrm{d} = \langle t, \langle m_j\rangle, o\rangle \in \mathrm{d}(\mathfrak{F})$ belongs to the* Utopian Collaborative Semantics *for $\mathfrak{F}$ (noted Utopian($\mathfrak{F}$)) if, and only if:*

Correctness: *if m_j is the last move in the sequence, then $\Phi(\mathbf{PU}^j_{\mathrm{d}}, t) = o$.*
Global Progress: *for each move $m_j = \langle id_j, X_j\rangle$ in the sequence, there exists $Y \subseteq \mathbf{PR}_{\mathfrak{F}}$ such that $X_j \subseteq Y$ and Y is a pure t-relevant contribution to $\mathbf{PU}^{j-1}_{\mathrm{d}}$.*
Global Completeness: *if m_j is the last move in the sequence, then $\mathbf{PR}_{\mathfrak{F}}$ is not a weak t-relevant contribution to $\mathbf{PU}^j_{\mathrm{d}}$.*

Requirement R_3 is achieved by the *Correctness* condition, which states that the dialogue outcome coincides with the application of the reasoning model to the public

[4] The term *weak relevance* is used in [5] in a different sense, which should not be related to the one introduced here.

knowledge at the final step of the dialogue (*i.e.,* the outcome of the dialogue can be obtained by reasoning from all that has been said). In the case of the $\mathfrak{I}^{\mathrm{ar}}$ framework, for instance, this means that the dialogue outcome is Yes if, and only if, the claim (topic) results justified considering all the arguments and evidence exposed during the dialogue. Requirement R_2 is achieved by the *Global Progress* condition, which states that each move in the sequence is part of a distributed pure relevant contribution to the public knowledge generated so far. Finally, requirement R_1 is achieved by the *Global Completeness* condition, which states that there are no more relevant contributions, not even distributed among different knowledge bases, after the dialogue ends. Notice that the three conditions are simultaneously satisfiable by any $\mathfrak{DF}$ and topic, *i.e.,* there always exists at least one dialogue which belongs to this semantics, as stated in the following proposition.

Proposition 1 (Satisfiability). *For any $\mathfrak{DF}$ $\mathfrak{F} = \langle \mathcal{R}_t, \varPhi, Ag \rangle$, the set Utopian$(\mathfrak{F})$ contains at least one dialogue over each possible topic in $\mathcal{L}_{\mathrm{T}}$.*

Furthermore, any sequence of moves satisfying *global progress* can be completed to a dialogue belonging to the semantics. This means that a system implementation under this semantics would not need to do *backtracking*. Although this property is useless for the case of the utopian semantics which, as will be seen in short, is not implementable in a distributed system, it will be useful in the case of the two practical semantics that will be presented in Sec. 6.

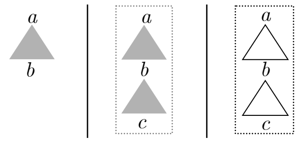
Definition 14. *A dialogue* d_2 *over a topic t is a* continuation *of a dialogue* d_1 *over the same topic t if, and only if, the sequence of moves of* d_2 *can be obtained by adding zero or more elements to the sequence of moves of* d_1.

Proposition 2 (No Backtracking). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \varPhi, Ag \rangle$ be a $\mathfrak{DF}$, and* $\mathrm{d}_1 \in \mathrm{d}(\mathfrak{F})$. *If* d_1 *satisfies* global progress *under $\mathfrak{F}$, then there exists a dialogue* $\mathrm{d}_2 \in$ *Utopian*$(\mathfrak{F})$ *which is a* continuation *of* d_1.

Note that the truth of the previous statements (regarding *satisfiability* and *no backtracking*) comes from the following facts, which can be easily proven: (1) if *global completeness* is not achieved, then there exists at least one possible move that can be added to the sequence according to *global progress*, and (2) the *correctness* condition is orthogonal to the other two. Next, an illustrative example of the dialogues generated under the Utopian Semantics is given.

Example 1. Consider an instance of the $\mathfrak{I}^{\mathrm{ar}}(\mathbf{G}, \mathbf{R})$ framework, where the set Ag is composed by $K_{\mathrm{A}} = \{\langle\{b\}, a\rangle, e\}$,

j	A	B	C	$\varPsi(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_1}, a)$
1	$\{\langle\{b\}, a\rangle\}$			No
2		$\{\langle\{c\}, b\rangle\}$		No
3			$\{c\}$	<u>Yes</u>



$K_{\mathrm{B}} = \{\langle\{c\}, b\rangle, \langle\{d\}, b\rangle, f\}$ and $K_{\mathrm{C}} = \{c, g\}$. The dialogue d_1 shown above, over topic 'a', and also all the permutations of its moves with the same topic and outcome, belong to the Utopian Semantics for the framework. The chart traces the dialogue, showing the partial results of reasoning from the public knowledge so far generated.

The last of these results (underlined) is the dialogue outcome. The evolution of the public knowledge is depicted in figures (a) through (c). At the first step of the dialogue, an inactive argument is added (a). The second step adds another inactive argument, supporting the first one (b). Finally, the supporting evidence is made available, and the whole structure becomes active (c), yielding to the justification of claim a.

An essential requirement of dialogue systems is ensuring the termination of the generated dialogues. This is intuitively related to requirement R_2 (achieved by *global progress*) since it is expected that agents will eventually run out of relevant contributions, given that their private knowledge bases are finite. This is actually true as long as the relevance notion satisfies an intuitive property, defined below, which states that a relevant contribution must add some new information to the public knowledge.

Definition 15 (Novelty). *A relevance notion $\mathcal{R}_t$ satisfies* novelty *iff the following condition holds: if $X\mathcal{R}_tS$ then $X \nsubseteq S$.*

Proposition 3 (Termination). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$ be a $\mathfrak{DF}$, and $\mathrm{d} = \big\langle t, \langle m_j\rangle, o\big\rangle \in \mathrm{d}(\mathfrak{F})$. If the notion $\mathcal{R}_t$ satisfies* novelty *and dialogue* d *satisfies* global progress *under $\mathfrak{F}$, then $\langle m_j\rangle$ is a finite sequence of moves.*

It is easy to see that any natural relevance notion satisfies novelty, since it is not possible for the conclusion achieved by the reasoning model to change without changing the topic nor the knowledge base.

Proposition 4. *For any reasoning model Φ, it holds that its associated natural relevance notion, $\mathcal{N}_t^{\Phi}$, satisfies novelty.*

Another desirable property of collaborative dialogue models is ensuring it is not possible to draw different conclusions, for the same set of agents and topic. In other words, from the entirety of the information, it should be possible to determine the outcome of the dialogue, no matter what sequence of steps are actually performed [5]. Furthermore, this outcome should coincide with the result of applying the reasoning model to the private knowledge involved in the dialogue. We emphasize that this is required for *collaborative* dialogues (and probably not for non-collaborative ones). For instance, in Ex. 1, all the possible dialogues under the semantics end up justifying the claim, which is also justified from $K_{\mathrm{A}} \cup K_{\mathrm{B}} \cup K_{\mathrm{C}}$. This is intuitively related to requirements R_1 (achieved by *global completeness*) and R_3 (achieved by *correctness*) since it is expected that the absence of relevant contributions implies that the current conclusion cannot be changed by adding more information. This is actually true as long as the relevance notion is the natural one associated to the reasoning model, or a *weaker* one, as stated below.

Definition 16 (Stronger Relevance Notion). *Let $\mathcal{R}_t$ and $\mathcal{R}'_t$ be relevance notions. We say that the notion $\mathcal{R}_t$ is* stronger or equal *than $\mathcal{R}'_t$ iff the following holds: if $X\mathcal{R}_tS$ then $X\mathcal{R}'_tS$ (i.e., $\mathcal{R}_t \subseteq \mathcal{R}'_t$). We will also say that $\mathcal{R}'_t$ is* weaker or equal *than $\mathcal{R}_t$.*

[5] This property, which we will call *outcome determinism*, has been studied in various works under different names. For instance in [6] it was called *completeness*. Notice that we use that term for another property, which is not the same but is related to the one under discussion.

Observe that here we use the term *weaker*, as the opposite of *stronger*, denoting a binary relation between relevance notions, and this should not be confused with its previous use in Def. 11 of *weak relevant contribution*.

Proposition 5 (Outcome Determinism). *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$ *and* $\mathrm{d} = \langle t, \langle m_j \rangle, o \rangle \in \mathrm{d}(\mathfrak{F})$. *If* d *satisfies* correctness *and* global completeness *under* $\mathfrak{F}$*, and* $\mathcal{R}_t$ is weaker or equal than $\mathcal{N}_t^{\Phi}$*, then* $o = \Phi(\mathbf{PR}_{\mathfrak{F}}, t)$.

For example, a relevance notion which detects the generation of new justified arguments (in the usual sense) for a given claim, would be *weaker* than the natural one. It is easy to see that this weaker relevance notion would also achieve *outcome determinism*.

The following corollaries summarize the results regarding the utopian semantics for $\mathfrak{DF}$s, and also for the particular case of $\mathfrak{IDF}$s. Clearly, these results are inherited respectively by $\mathfrak{F}^{\mathrm{ar}}$ and $\mathfrak{I}^{\mathrm{ar}}$.

Corollary 1. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$*. The dialogues in Utopian*$(\mathfrak{F})$ *satisfy* termination *and* outcome determinism*, provided that the relevance notion* $\mathcal{R}_t$ *satisfies* novelty *and is* weaker or equal *than* $\mathcal{N}_t^{\Phi}$.

Corollary 2. *Let* $\mathfrak{I}$ *be an* $\mathfrak{IDF}$*. The dialogues in Utopian*$(\mathfrak{I})$ *satisfy* termination *and* outcome determinism.

It is clear that Def. 13 of the Utopian Collaborative Semantics is not constructive, since both *global progress* and *global completeness* are expressed in terms of the private knowledge $\mathbf{PR}_{\mathfrak{F}}$, which is not entirely available to any of the participants. The following example shows that, it is not only not constructive, but also in many cases not even implementable in a distributed MAS.

Example 2. From Ex. 1, the dialogue d_2, shown on the right, does not belong to the Utopian Semantics since step 2 violates *global progress*. However, it would not be possible to design a dialogue system which allows d_1 (from Ex. 1) but disallows d_2, since agent B cannot know in advance that 'c', rather than 'd', holds.

j	A	B	C	$\Psi(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_2}, a)$
1	$\{\langle \{b\}, a \rangle\}$			No
2		$\{\langle \{d\}, b \rangle\}$		No
3		$\{\langle \{c\}, b \rangle\}$		No
4			$\{c\}$	$\underline{\text{Yes}}$

The undesired situation is caused by a relevant contribution distributed among several agents, in such a way that none of the parts is relevant by itself, leading to a tradeoff between requirements R_1 and R_2 (*i.e.,* between *global progress* and *global completeness*). In the worst case, each sentence of the contribution resides in a different agent. Thus, to avoid such situations, it would be necessary for the relevance notion to warrant that every relevant contribution contains at least one individually relevant sentence. When this happens, we say that the relevance notion satisfies *granularity*, defined next.

Definition 17 (Granularity). *Let* $\mathcal{R}_t$ *be a relevance notion. We say that* $\mathcal{R}_t$ *satisfies* granularity *iff the following holds: if* $X\mathcal{R}_tS$ *then there exists* $\alpha \in X$ *such that* $\alpha\mathcal{R}_tS$.

Unfortunately, the relevance notions we are interested in, fail to satisfy granularity. It does not hold in general for the natural notions associated to deductive inference mechanisms. In particular, it has been shown in Ex. 2 that it does not hold for $\mathcal{N}_\delta^{\Psi}$.

6 Practical Collaborative Semantics

The lack of granularity of relevance notions motivates the definition of alternative semantics which approach the utopian one, and whose distributed implementation is viable. The simplest approach is to relax requirement R_1 by allowing distributed relevant contributions to be missed, as follows.

Definition 18 (Basic Collaborative Semantics). *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$*. A dialogue* $\mathrm{d} = \langle t, \langle m_j \rangle, o \rangle \in \mathrm{d}(\mathfrak{F})$ *belongs to the* Basic Collaborative Semantics *for* $\mathfrak{F}$ *(noted Basic*$(\mathfrak{F})$*) if, and only if, the following conditions, as well as* **Correctness** *(Def. 13), hold:*

Local Progress: *for each move* $m_j = \langle id_j, X_j \rangle$ *in the sequence,* X_j *is a pure* t*-relevant contribution to* $\mathbf{PU}_{\mathrm{d}}^{j-1}$.
Local Completeness: *if* m_j *is the last move in the sequence, then it does not exist an agent* $K_{id} \in Ag$ *such that* K *is a weak* t*-relevant contribution to* $\mathbf{PU}_{\mathrm{d}}^{j}$.

In the above definition, requirement R_2 is achieved by the *local progress* condition which states that each move in the sequence constitutes a pure relevant contribution to the public knowledge generated so far. Notice that this condition implies global progress (enunciated in Sec. 5), as stated below.

Proposition 6. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$*, and* $\mathrm{d} \in \mathrm{d}(\mathfrak{F})$*. If the dialogue* d *satisfies* local progress*, then it satisfies* global progress *under* $\mathfrak{F}$*.*

Requirement R_1 is now compromised. The *local completeness* condition states that each agent has no more relevant contributions to make after the dialogue ends. Unless the relevance notion satisfies granularity, this is not enough for ensuring global completeness (enunciated in Sec. 5), since there could be a relevant contribution distributed among several agents, in such a way that none of the parts is relevant by itself.

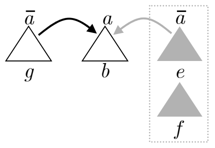
Proposition 7. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$*, and* $\mathrm{d} \in \mathrm{d}(\mathfrak{F})$*. If the dialogue* d *satisfies* global completeness*, then it satisfies* local completeness *under* $\mathfrak{F}$*. The reciprocal holds if, and only if, the relevance notion* $\mathcal{R}_t$ *satisfies* granularity.

As a result, requirement R_4 (termination) is achieved, given the same condition as in Sec. 5, whereas requirement R_5 (outcome determinism) cannot be warranted. These results are summarized in the corollary below. Clearly, these results are inherited by $\mathfrak{F}^{\mathrm{ar}}$ and $\mathfrak{I}^{\mathrm{ar}}$.

Corollary 3. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{DF}$*. The dialogues in Basic*$(\mathfrak{F})$ *satisfy* termination*, provided that the relevance notion* $\mathcal{R}_t$ *satisfies* novelty.

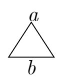
Corollary 4. *Let* $\mathfrak{I}$ *be an* $\mathfrak{IDF}$*. The dialogues in Basic*$(\mathfrak{I})$ *satisfy* termination*.*

Considering the same scenario as in Ex. 1, it is easy to see that the only possible dialogue under the Basic Semantics is the empty one (*i.e.,* no moves are performed), with outcome No. A more interesting example is shown next.

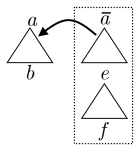


Example 3. Consider an instance of the $\mathfrak{I}^{\mathrm{ar}}(\mathbf{G}, \mathbf{R})$ framework, where the set Ag is composed by $K_{\mathrm{A}} = \{\langle\{b\}, a\rangle, b, g\}$, $K_{\mathrm{B}} = \{\langle\{e\}, \overline{a}\rangle, \langle\{f\}, e\rangle, f, g\}$, and $K_{\mathrm{C}} = \{\langle\{g\}, \overline{a}\rangle, \overline{e}\}$. Also consider that both $\langle\{e\}, \overline{a}\rangle$ and $\langle\{g\}, \overline{a}\rangle$ attack $\langle\{b\}, a\rangle$, but not viceversa. The private knowledge is depicted on the right. The dialogue d_3 traced below, over topic a, belongs to the Basic Semantics for the $\mathfrak{IDF}$ instantiated above. The evolution of the public knowledge is depicted in figures (a) through (c). At the first step, an active argument for 'a' is added (a). At the second step, an attacking structure is added (b). Finally, the attacking structure is deactivated due to a supporting argument becoming inconsistent wrt. new evidence (c). Note that *global completeness* is not achieved, since there still exists a distributed relevant contribution when the dialogue ends: $\{\langle\{g\}, \overline{a}\rangle, g\}$. Consequently, outcome determinism is not achieved: the outcome is Yes whereas the result of reasoning from the private knowledge is No.

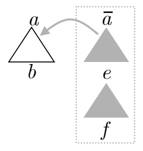
j	A	B	C	$\Psi(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_3}, a)$
1	$\{\langle\{b\}, a\rangle, b\}$			Yes
2		$\{\langle\{e\}, \overline{a}\rangle, \langle\{f\}, e\rangle, f\}$		No
3			$\{\overline{e}\}$	<u>Yes</u>



(a)



(b)



(c)

In Sec. 3 we argued that requirement R_1 may be mandatory in many domains, but the Basic Semantics does not achieve it unless the relevance notion satisfies granularity, which does not usually happen. In order to make up for this lack of granularity, we propose to build a new notion (say $\mathcal{P}$) based on the original one (say $\mathcal{R}$) which ensures that, in the presence of a distributed relevant contribution under $\mathcal{R}$, *at least one* of the parts will be relevant under $\mathcal{P}$. We will say that $\mathcal{P}$ is a *potential relevance notion for* $\mathcal{R}$, since its aim is to detect contributions that could be relevant within certain *context*, but it is uncertain whether that context actually exists or not. Observe that the context is given by other agents' private knowledge, which has not been exposed yet.

Below we define the binary relation (*"is a potential for"*) between relevance notions, and also its propagation to $\mathfrak{DF}$s. Clearly, if a relevance notion already satisfies granularity then nothing needs to be done. Indeed, it would work as a potential relevance notion for itself.

Definition 19 (Potential Relevance Notion). *Let $\mathcal{R}_t$ and $\mathcal{P}_t$ be relevance notions. We say that $\mathcal{P}_t$ is a* potential (relevance notion) for *$\mathcal{R}_t$ iff the following conditions hold: (1) $\mathcal{R}_t$ is* stronger or equal *than $\mathcal{P}_t$, and (2) if $X\mathcal{R}_tS$ then there exists $\alpha \in X$ such that $\alpha\mathcal{P}_tS$. If $X\mathcal{P}_tS$ and $\mathcal{P}_t$ is a potential for $\mathcal{R}_t$, we say that X is a* potential t-relevant contribution to S under $\mathcal{R}_t$.

Definition 20 (Potential Dialogue Framework). *Let $\mathfrak{F} = \langle\mathcal{R}_t, \Phi, Ag\rangle$ and $\mathfrak{F}^* = \langle\mathcal{P}_t, \Phi, Ag\rangle$ be $\mathfrak{DF}$s. We say that $\mathfrak{F}^*$ is a* potential (framework) *for $\mathfrak{F}$ if, and only if, $\mathcal{P}_t$ is a potential relevance notion for $\mathcal{R}_t$.*

Proposition 8. *If the relevance notion $\mathcal{R}_t$ satisfies* granularity*, then $\mathcal{R}_t$ is a potential relevance notion for itself.*

Now we will show a more interesting potential relevance notion, in the context of the $\mathfrak{I}^{\mathrm{ar}}$ framework. The basic idea is to detect contributions that would be relevant given a certain *situation* (*i.e.,* a certain set of evidence). To that end, we first introduce the concept of *abduction set* associated to a given claim δ and a given set K. This abduction set reflects how the current situation (represented by the evidence in K) could be minimally expanded in order to change the justification status of δ.

Definition 21 (Abduction Set). *Let $K \subseteq \mathbf{L} \cup \mathbf{L}_{\mathrm{A}}$ and $\delta \in \mathbf{L}$. The* abduction set *of δ from K is defined as: $\mathcal{AB}(K, \delta) = \{E \subseteq \mathbf{L}$: E is consistent wrt. the evidence in K, and E is a minimal natural δ-relevant contribution to $K\}$.*

Example 4. Consider the $\mathfrak{I}^{\mathrm{ar}}$ framework. In the chart on the right, the second column shows the abduction set of claim "a", from the set K on the first column. In the last case, assume that the argument $\langle\{e\}, \overline{a}\rangle$ attacks the argument $\langle\{b\}, a\rangle$, but not viceversa.

K	$\mathcal{AB}(K, a)$
$\{\}$	$\{\{a\}\}$
$\{\ \langle\{b\}, a\rangle\ \}$	$\{\{a\}\{b\}\}$
$\{\ \langle\{b\}, a\rangle,\ b\ \}$	$\{\{\}\}$
$\{\ \langle\{b\}, a\rangle,\ b,\ \langle\{e\}, \overline{a}\rangle,$ $\langle\{f\}, e\rangle,\ f\ \}$	$\{\{a\}\{\overline{e}\}\}$

Now we are able to introduce an *abductive relevance notion* $\mathcal{A}^{\Psi}_{\delta}$. Under this notion, a set X is considered an δ-relevant contribution to K if, and only if, its addition generates a new element in the abduction set of δ from K. This means that a new potential situation in which the justification status of δ would change has arisen. It can be shown (proof is omitted due to space reasons) that $\mathcal{A}^{\Psi}_{\delta}$ is a potential relevance notion for $\mathcal{N}^{\Psi}_{\delta}$.

Definition 22 (Abductive Relevance). *Let $K \subseteq \mathbf{L} \cup \mathbf{L}_{\mathrm{A}}$ and $\delta \in \mathbf{L}$. A set $X \subseteq \mathbf{L} \cup \mathbf{L}_{\mathrm{A}}$ is an δ-relevant contribution to K under $\mathcal{A}^{\Psi}_{\delta}$ iff there exists $E \subseteq \mathbf{L}$ such that: (1) $E \in \mathcal{AB}(K \cup X, \delta)$ and (2) $E \notin \mathcal{AB}(K, \delta)$.*

Proposition 9. *The notion $\mathcal{A}^{\Psi}_{\delta}$ is a potential relevance notion for $\mathcal{N}^{\Psi}_{\delta}$.*

Returning to the semantics definition, the idea is to use the potential framework under the Basic Semantics, resulting in a new semantics for the original framework. Next we introduce the *Full Collaborative Semantics*, which is actually a family of semantics: each possible potential $\mathfrak{DF}$ defines a different semantics of the family.

Definition 23 (Full Collaborative Semantics). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$ be a $\mathfrak{DF}$. A dialogue $\mathrm{d} = \langle t, \langle m_j\rangle, o\rangle \in \mathrm{d}(\mathfrak{F})$ belongs to the* Full Collaborative Semantics *for $\mathfrak{F}$ (noted Full($\mathfrak{F}$)) iff $\mathrm{d} \in$ Basic($\mathfrak{F}^*$) for some $\mathfrak{DF}$ $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, Ag\rangle$ which is a potential for $\mathfrak{F}$. We will also use the more specific notation $\mathrm{d} \in$ Full($\mathfrak{F}, \mathcal{P}_t$).*

In this way, each agent would be able to autonomously determine that she has no more potential relevant contributions to make, ensuring there cannot be any distributed relevant contribution when the dialogue ends, and hence achieving R_1. In other words, achieving local completeness under the potential relevance notion implies achieving global completeness under the original one, as stated below.

Proposition 10. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$ and $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, Ag\rangle$ be $\mathfrak{DF}$s such that $\mathfrak{F}^*$ is a potential for $\mathfrak{F}$, and $\mathrm{d} \in \mathrm{d}(\mathfrak{F})$. If dialogue d satisfies* local completeness *under $\mathfrak{F}^*$, then it satisfies* global completeness *under $\mathfrak{F}$.*

Requirement R_2 is now compromised, since the context we have mentioned may not exist. In other words, achieving local progress under the potential relevance notion does not ensure achieving global progress under the original one. The challenge is to design *good* potential relevance notions which considerably reduce the amount of cases in which a contribution is considered potentially relevant but, eventually, it is not. Observe that a relevance notion which considers any sentence of the language as relevant, works as a potential for any given relevance notion, but it is clearly not a good one.

Next we summarize the results for the dialogues generated under the Full Collaborative Semantics. By achieving global completeness these dialogues achieve outcome determinism under the same condition as before. Although global progress is not achieved under the original relevance notion, it is achieved under the potential one, and thus termination can be ensured as long as the latter satisfies novelty. Clearly, these results are inherited by $\mathfrak{F}^{\mathrm{ar}}$ and $\mathfrak{I}^{\mathrm{ar}}$.

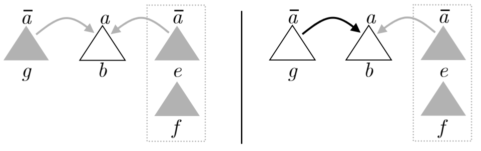
Corollary 5. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag \rangle$ *be a* $\mathfrak{D}\mathfrak{F}$*, and* $\mathcal{P}_t$ *a potential for* $\mathcal{R}_t$*. The dialogues in Full*$(\mathfrak{F}, \mathcal{P}_t)$ *satisfy* termination *and* outcome determinism*, provided that* $\mathcal{P}_t$ *satisfies* novelty *and* $\mathcal{R}_t$ *is* weaker or equal *than* $\mathcal{N}_t^{\Phi}$.

Corollary 6. *Let* $\mathfrak{I} = \langle \Phi, Ag \rangle$ *be an* $\mathfrak{I}\mathfrak{D}\mathfrak{F}$*, and* $\mathcal{P}_t$ *a potential for* $\mathcal{N}_t^{\Phi}$*. The dialogues in Full*$(\mathfrak{I}, \mathcal{P}_t)$ *satisfy* termination *and* outcome determinism*, provided that* $\mathcal{P}_t$ *satisfies* novelty.

Example 5. Both dialogues d_1 and d_2, presented in Ex. 1 and Ex. 2 respectively, belong to Full$(\mathfrak{I}^{\mathrm{ar}}, \mathcal{A}_{\delta}^{\mathrm{ar}})$. Also belongs to this semantics the dialogue which results from d_2 by interchanging steps **2** and **3**, or by merging these two steps together in a single one. Note that all these dialogues achieve *global completeness*, although *global progress* is achieved only by dialogue d_1.

Example 6. The dialogue d_3 from Ex. 3 can be completed according to Full$(\mathfrak{I}^{\mathrm{ar}}, \mathcal{A}_{\delta}^{\mathrm{ar}})$, as shown on the right. The fifth column of the chart shows the evolution of the abduction set of the claim “a” from the generated public knowledge. An additional step **0** is added, in order to show the initial state of this abduction set. At step **4** an attacking, for the meantime inactive, argument is added (a). This generates a new potential situation in which the claim ‘a’ would not be justified any more. At step **5** the previous situation is realized, activating the attack and leaving the claim ‘a’ not justified (b). Note that other dialogues also belong to the Full Collaborative Semantics, since the first three steps do not actually need to be natural relevant contributions. For instance, agent A could expose the argument $\langle \{b\}, a \rangle$ and then, in the next step, the supporting evidence.

j	A	B	C	$\mathcal{AB}(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_4}, a)$	$\Psi(\mathbf{PU}^{\mathrm{step}}_{\mathrm{d}_4}, a)$
0				$\{\{a\}\}$	No
1	$\{\langle \{b\}, a \rangle, b\}$			$\{\{\overline{a}\}\}$	Yes
2		$\{\langle \{e\}, \overline{a} \rangle, \langle \{f\}, e \rangle, f\}$		$\{\{a\}, \{\overline{e}\}\}$	No
3			$\{\overline{e}\}$	$\{\{\overline{a}\}\}$	Yes
4			$\{\langle \{g\}, \overline{a} \rangle\}$	$\{\{\overline{a}\}, \{g\}\}$	Yes
5		$\{g\}$		$\{\{a\}\}$	<u>No</u>



(a) (b)

Moreover, agent B could make her attack while agent A's argument is still inactive. In that moment, the element $\{b, \overline{e}\}$ would be added to the abduction set.

It is important to note the existence of alternative potential relevance notions which may be also adequate, and which may cause variations in the behavior of the dialogue. For instance, a variant of the abductive relevance notion defined earlier is to consider a contribution as relevant if its addition either adds or deletes an element of the abduction set. The latter case, deletion, could be seen as discarding a possible explanation before it is actually realized (or activated). For instance, assume from Ex. 6 that agent A exposes just the argument $\langle\{b\}, a\rangle$ without the activating evidence. Before the activation, it would be possible for agent C to make an attack by exposing the argument $\langle\{g\}, \overline{a}\rangle$ together with the supporting evidence $\{g\}$. Observe that that exact sequence of steps is not allowed under the abductive notion defined earlier, in Def. 22, since no element is added to the abduction set, instead the element $\{b\}$ is deleted. Under that notion, it would be necessary for agent A to activate her argument before agent C can attack it. It is easy to see that the alternative notion which considers not only the expansion but also the reduction of the abduction set, may in some cases lead to shorter dialogues.

Results regarding *satisfiability* and *no-backtracking* also hold under the two practical semantics we have presented in this section, as stated below.

Proposition 11. *For any $\mathfrak{DF}$ $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$, each one of the sets Basic$(\mathfrak{F})$ and Full$(\mathfrak{F}, \mathcal{P}_t)$, contains at least one dialogue over each possible topic in $\mathcal{L}_{\mathrm{T}}$.*

Proposition 12. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$ and $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, Ag\rangle$ be $\mathfrak{DF}$s such that $\mathfrak{F}^*$ is a potential for $\mathfrak{F}$, and let $\mathrm{d}_1 \in \mathrm{d}(\mathfrak{F})$. If d_1 satisfies* local progress *under $\mathfrak{F}$ ($\mathfrak{F}^*$), then there exists a dialogue $\mathrm{d}_2 \in$ Basic$(\mathfrak{F})$ ($\mathrm{d}_2 \in$ Full$(\mathfrak{F}, \mathcal{P}_t)$) which is a* continuation *of* d_1.

Finally, a result showing the relation among the three collaborative semantics, for the case in which the relevance notion satisfies *granularity*, is stated.

Proposition 13. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, Ag\rangle$ be a $\mathfrak{DF}$. If the relevance notion $\mathcal{R}_t$ satisfies* granularity*, then it holds that: Basic$(\mathfrak{F})$ = Full$(\mathfrak{F}, \mathcal{R}_t) \subseteq$ Utopian$(\mathfrak{F})$.*

To sum up, we have defined three collaborative semantics for a $\mathfrak{DF}$. The Utopian Semantics describes an idealistic, in most cases impractical behavior of a collaborative dialogue. Its usefulness is theoretical. It is approximated, in different ways, by the other two practical semantics. The Basic Semantics, on the other side, describes a straightforward implementable behavior of a collaborative dialogue. The weak point of this semantics is not ensuring *global completeness* (neither *outcome determinism*, thus). The Full Collaborative Semantics is actually a family of semantics: each potential relevance notion $\mathcal{P}_t$ associated to $\mathcal{R}_t$ defines a semantics of the family. Thus, the constructiveness of these semantics is reduced to the problem of finding a potential relevance notion for $\mathcal{R}_t$. These semantics succeed in achieving *global completeness*, at the price of allowing moves which may not be allowed by the Utopian Semantics. The goodness of a given potential relevance notion increases as it minimizes the amount of such moves.

7 Related Work

There are some works particulary related to our proposed approach, due to any of the following: (a) an explicit treatment of the notion of relevance in dialogue, (b) the search of the *global completeness* property, as we called it in this work, or (c) a tendency to examine general properties of dialogues rather than designing particular systems.

Regarding category (a), in [7], [8] and [5], the importance of a precise relevance notion definition is emphasized. However, these works focus on argumentation-based persuasion dialogues (actually a subset of those, which the author called *disputes*), which belong to the non-collaborative class, and thus *global completeness* is not pursued. Instead, the emphasis is put on properties with similar spirit to our properties of *correctness* and *local progress* (*i.e.,* only the *public knowledge* involved in the dialogue is given importance). In [8] the author considers dynamic disputes in which two participants (proponent and opponent) interchange arguments and counter-arguments, and studies two properties of protocols (namely *soundness* and *fairness*) regarding the relation between the generated public knowledge and the conclusion achieved (in this case, the *winner* of the dispute). The author also gives a natural definition of when a move is relevant: *"iff it changes the status of the initial move of the dispute"* whose spirit is similar to our definition of *natural relevance notion* but taken to the particular case in which the reasoning model is a logic for defeasible argumentation. In [5] the author considers more flexible protocols for disputes, allowing alternative sets of locutions, such as *challenge* and *concede*, and also a more flexible notion of relevance.

Another work in which *relevance* receives an explicit treatment is [6], where the authors investigate the relevance of utterances in an argumentation-based dialogue. However, our *global completeness* property is not pursued, so they do not consider the problematic of distributed contributions (distributed arguments in this case). They study three notions of relevance showing how they can affect the dialogue outcome.

Regarding category (b), in [9] an inquiry dialogue protocol which successfully pursues our idea of *global completeness* is defined. However, the protocol is set upon a particular argumentative system, with the design methodology implicit. They take a simplified version of the *DeLP* system [10], and define an *argument inquiry dialogue* which allows exactly two agents to jointly construct arguments for a given claim. In the present work, we not only explicitly and abstractly analyze the distributed relevance issue, but also consider the complete panorama of collaborative dialogue system behavior, including *correctness* and *progress* properties.

Regarding category (c), different measures for analyzing argumentation-based persuasion are proposed in [11]: measures of the quality of the exchanged arguments, of the behavior of each agent, and of the quality of the dialogue itself in terms of the relevance and usefulness of its moves. The analysis is done from the point of view of an external agent (*i.e., private knowledge* is not considered), and it is focused in a non-collaborative dialogue type, so they are not concerned with our main problematic.

8 Conclusions

We have shown how an existent abstract dialogue framework can be instantiated for modeling argumentation-based dialogues. This new instance, in contrast with a previous one in terms of Propositional Logic Programming [3], naturally deals with possible differences of opinion that can emerge among participants in a dialogue. Also the versatility of the abstract framework is shown through this new instantiation based on a non-monotonic reasoning model.

The obtained framework instance is capable of modeling collaborative argumentation-based dialogues among any number of participants, each of them exposing indifferently either type of argument ('for' and 'against'), and also building arguments together. The model inherits the chance of parametrization and properties from the abstract framework. The most appropriate relevance notion can be chosen according to the dialogue purpose, *e.g.* all the possible justifications for a given claim could be searched, or just one. Also the degree of collaboration is selectable by picking a certain semantics, either basic or full collaborative, according to domain requirements.

In particular, by picking the natural relevance notion and the full collaborative semantics, a model for argumentation-based inquiry has been provided, which ensures a sound and complete (in the usual sense) distributed reasoning. This model is still parametrizable, since different potential relevance notions could be investigated, for instance trying to enhance efficiency. This last issue has been left for future research.

Finally, another item left as future work is the consideration of the case in which the agents can disagree also about evidence. This would imply redefining the reasoning model in order to deal with inconsistencies in the set of evidence.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2) (1995) 321–357
2. Rotstein, N., Moguillansky, M., García, A., Simari, G.: An Abstract Argumentation Framework for Handling Dynamics. In: NMR. (2008) 131–139
3. Marcos, M.J., Falappa, M.A., Simari, G.R.: Semantically characterizing collaborative behavior in an abstract dialogue framework. In: FoIKS. (2010) 173–190
4. Walton, D., Krabbe, E.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY (1995)
5. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. J. Log. Comput. (2005)
6. Parsons, S., McBurney, P., Sklar, E., Wooldridge, M.: On the relevance of utterances in formal inter-agent dialogues. In: AAMAS'2007, Honolulu, Hawai'i. (2007)
7. Prakken, H.: On dialogue systems with speech acts, arguments, and counterarguments. In: JELIA. (2000)
8. Prakken, H.: Relating protocols for dynamic dispute with logics for defeasible argumentation. Synthese (2001)
9. Black, E., Hunter, A.: A generative inquiry dialogue system. In: AAMAS'2007, Honolulu, Hawai'i. (2007)
10. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative aproach. Theory and Practice of Logic Programming **4**(1) (2004) 95–138
11. Amgoud, L., de Saint-Cyr, F.D.: Measures for persuasion dialogs: A preliminary investigation. In: COMMA'2008, Toulouse - France. (2008)

Towards a dialectical approach for conversational agents in selling situations

Maxime Morge, Sameh Abdel-Naby, and Bruno Beaufils

Laboratoire d'Informatique Fondamentale de Lille
Université Lille 1
Bât M3 - F-59655 Villeneuve d'Ascq
{maxime.morge,sameh.abdel-naby,bruno.beaufils}@lifl.fr

Abstract. The use of virtual agents to intelligently interface with online customers of e-commerce businesses is remarkably increasing. Most of these virtual agents are designed to assist online customers while searching for information related to a specific product or service, while few agents are intended for promoting and selling a product or a service. Within the later type, our aim is to provide proactive agents that recommend a specific item and justify this recommendation to a customer based on his purchases history and his needs. In this paper, we propose a dialectical argumentation approach that would allow virtual agents that have sales goals to trigger persuasions with e-commerce's customers. Then, we illustrate the proposed idea through its integration with an example from real-life.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Intelligent Agents.

General Terms
Algorithms.

Keywords
Argumentation, E-commerce, Agents, Language Processors.

1 Introduction

Within the last twelve years, precisely from 1998 wherein the dot-coms' boom first made an impact, e-commerce has succeeded to pursue a massive number of shoppers to change their idea of buying [1]. Several existing businesses have taken an advantage of this boom by adding a virtual presence to their physical one by means of an e-commerce website, these companies are now called *brick and mortar* businesses (e.g., Barnes & Noble). Additionally, new companies that exist only through the web, called *bricks and clicks* businesses, have also appeared (e.g., Amazon). Although the online presence of companies is cost-efficient, yet the lack of a persuading salesman affects the transformation ratio (sales vs. visits).

Apart from the *Business*'s reaction to the boom, in Computer Science, several research efforts were made to study, analyze, and better shape the processes of assisting customers while being present in an e-commerce space [2, 3]. In Artificial Intelligence, a considerable amount of the research conducted in the area of Software Agents [4] focus on the enhancement and the proper provision of online Embodied Conversational Agents (ECAs) [5].

Whether these agents sell, assist, or just recommend, it is now clear that such autonomous agents are capable of engaging in verbal and non-verbal dialogues with e-commerce's customers. However, the ability of these agents to transform an ordinary visitor of an e-commerce who needs assistance to an actual buyer is yet of no notable weight. For an overview of the issues encountering the development of virtual sales agents refer to [6].

In this paper, we propose the use of dialectical argumentation technologies as a step on the way to increase the sales-oriented negotiation skills of software agents in the business-to-consumer (B2C) segment of e-commerce. For this purpose, we suggest the exploitation of existing argumentation tools, such as those found in [7–9]. Using these tools we intend to build a sales-driven dialogue system that is capable of leading a virtual seller agent to influence the decision of a potential buyer in an e-commerce setting. Then, we illustrate the proposed idea through its integration with an example from real-life.

This paper is organized as follows. In section 2 we give an overview of the existing dialogue systems while pointing out their limitations. In section 3 we adopt a different approach for dialogue management based upon argumentation. Section 4 illustrates this approach using an intuitive scenario. Section 5 briefly describes the CSO language processor on which our dialogue system is based. The rest of the paper overviews the dialectical argumentation technology we consider. Section 6 outlines the dialogue-game protocol we use. Section 7 presents our realization of the dialogue strategy. We then conclude this paper by discussing some of the related work and, providing a summary of our future work.

2 Dialogue systems

A dialogue system is a computer system that is capable of interacting with humans using the language they understand - *natural language*. Similar to that we can find TRAINS-93 [10], Collagen [11] and Artemis Agent Technology [12], which are mixed-initiative dialogue systems for collaborative problem solving. These dialogue systems can respond to initiatives made by users and, they also take initiatives themselves, which is required to support a selling process.

TRAINS-93 [10], Collagen [11] and Artemis Agent Technology [12] are adopting the same approach of focusing on the dialogue modelling itself besides the dialogue management that is based on intentions recognition. For example, out of the following utterance of a user, "I want to purchase a quilt", there can be three possible interpretations:

1. It can be a direct report of a need;

2. It can be a statement of a goal that a user is pursuing independently;
3. It can be a proposal to adopt this joint goal.

Particularly, the discourse structure considered by Collagen in [11] is based on a comprehensive axiomatization of SharedPlans [13], while TRAINS-93 and Artemis Agent Technology are based upon a BDI approach [14]. The semantics of utterances is specified with the help of a first order modal logic language using operators as Beliefs, Desires and Intentions. The notions of persistent goal is a composite mental attitude which is defined from the previous operators in order to formalize the intention expressed by utterances. According to the semantic language of FIPA-ACL [15] adopted by the Artemis Agent Technology, an agent i has p as a persistent goal, if i has p as a goal and is self-committed toward this goal until i comes to believe that the goal is achieved or, this goal is unachievable. Here, an intention is defined as a persistent goal imposing the agent to act, which accordingly generates a planning process.

The process of inferring intentions from actions is needed to constraint and reduce the amount of communications exchanged. Also, it is worth noticing here that it is hard to incorporate this process into practical computer systems due to the complexities encountered while facilitating natural intractability. Therefore, it is then required to develop a heuristic mechanism for software agents in a collaborative setting.

For this purpose, dialogue systems are required to recognize the intention of the user and reason about it. The implementation of this theory is problematic due to its computational complexity [16]. Moreover, the specification of the semantics for the speech acts in terms of mental states is not adapted for resolving the conflicts which can appear during a selling process. For instance, an information that is received by a virtual seller agent must be adopted even if this information is contradictory with its beliefs. Those are the reasons why we consider an alternative approach based upon dialectical argumentation.

3 Dialectical approach

Our approach for dialogue modelling considers the exchange of utterances as an argumentation process regulated by some normative rules that we call *dialogue-game protocol.* Our approach is inspired by the notion of dialectical system that Charles L. Hamblin introduced in [17]. A **dialectical system** is a family of regulated dialogue, (i.e., a system through which a set of participants communicate in accordance with some rules).

From this perspective, Walton and Krabbe in [18] define a **dialogue** as a coherent and structured sequence of utterances aiming at moving from an initial state to reach the goals of the participants. These are the dialogue's *goals* that can be shared by the participants or they can be also each of the participants' individual goals. Based on this definition, Walton and Krabbe have distinguished between five main categories of dialogues depending on the initial situation and goals. These categories are: information seeking, persuasion, negotiation, enquiry and deliberation [18].

Table 1: Systemic overview of dialogue categories

Initial situation → Goal ↓	Conflict	Open problem	Ignorance of a participant
Stable agreement i.e., Resolution	persuasion	enquiry	information seeking
Practical settlement i.e., Decision	negotiation	deliberation	$\emptyset$

Table 1 represents the analysis grid for dialogues proposed by Walton and Krabbe. An **information seeking** appears when a participant aims at catching knowledge from its interlocutor. The goal is to spread knowledge. In a **persuasion** dialogue, the initial situation is disagreement, (i.e., a conflict of opinion). The goal consists of solving the conflict by verbal means. In a **negotiation** dialogue, the initial situation is a conflict of interest mixed with a need for collaboration. The goal consists of a deal, i.e. an agreement attracting all participants to maximizing their gains. An **enquiry** dialogue aims at establishing (or demonstrating) the truth of a predicate. This one must answer to an open question and a stable agreement emerges. Each participant aims at extending their knowledge. A **deliberation**, as an enquiry, begins with an open problem rather than a conflict. The discussion is about the means and ends of a future action. It is worth noticing that, in real world, the nature of dialogues can be mixed. A dialogue can be composed of different sub-dialogues with different natures as we will see in our scenario.

4 Dialogue: Phases & Purposes.

In this section, we explain the different phases of the overall online sales process that we are attempting to tackle in our research. Within these phases, we expect our virtual agent to rely on a specific language processor - *explained further ahead* - to handle online one-to-one conversations, related misspelling, and the use of diverse languages. Since the existing language processor is already capable of handling what is known to us as After-Sales, (i.e., assisting online users while searching for problems' answers), we then became extra interested to increase the salesability of this agent.

- **BEFORE-SALE**: in this phase we distinguish between two different processes that are possibly interleaved: a) the process of needs identification and, b) the process of product selection.

  The **Needs Identification** can be performed with the help of an information seeking dialogue shifting from an initial asymmetric situation to a final one where both of the players share the user requirements.

  The **Product Selection** allows the participants to constraint and to reduce the amount of communication by considering only relevant products later in the selling process. This task, in overall, also supports the information seeking dialogue where the virtual seller agent asks discriminatory questions in order to narrow its focus into a single product.

Both of these dialogues can be interleaved. The aim of the virtual seller agent here is to spread information about the products, while the aim of the user is expected to be the spreading of information about his needs.

- **SALE**: here, the aim for all dialogues' parties is to bargain over their interests and, eventually, "*make a deal*". For this purpose, the participants play a role in a **negotiation** dialogue. The simplest dialogue is: the virtual seller agent makes an offer and the user accepts or refuses this proposal.

  If there is no single product corresponding to the user needs, then the participants attempt to maximize their benefits by conceding some aspects while insisting on others. If no product is matching user's needs, the user's high-ranked features of the products, (i.e., top priority conditions), are then altered to a lower ranked ones. On the other hand, if more than one product is corresponding to user's needs, the virtual seller agent picks the ones with the highest gross margin.

  Later to that, the virtual seller agent can suggest - *after a single sale* - additional sub-items or an offer which invokes more of the user needs. These later approaches are well-known marketing techniques, (i.e., cross-selling and up-selling). In both cases, the new items aim at persuading the customer.

- **AFTER-SALE**: The initial condition of this phase is not coming out of a sales conflict, however, it comes out of the user's unawareness of the problem related to the concerned product. Therefore, participants here aim at extending their knowledge about each foreseen encounter and, eventually, decide about their subsequent action, (e.g., product replacement). As a result, users here play in an enquiry dialogue or a deliberation fashion.

The dialogue in Fig 1 consists of a specific case run which embraces the before-sale and sale phases between a user and a virtual seller agent (VSA). While the utterances #3 and #5 are related to the *Needs Identification*, the utterances #1,#6 and #8 are of the *Product Selection.* The rest of utterances are directly related to the sales process. The utterance #9 is a single-sale offer, the utterance #11 captures an up-selling offer and the utterance #13 is a cross-selling proposal.

It is also worth highlighting here that relying on argumentation will make the virtual seller agent more credible and, then, persuasive, since its appearance or involvement will yield from the fulfillment of a set of predefined conditions, (e.g., user X is wandering in the quilt section).

The After-Sale phase, (i.e., assistance), is out of the scope of this paper since the existing query answering systems, such as the CSO Language Processor we present in the next section, have been practically proven to be efficient for such tasks.

1. `VSA:` Your moves through our website indicates that you might be interested in Quilts, do you need help to select one?
2. `User:` Nice.
3. `VSA:` Are you allergic?
4. `User:` Yes.
5. `VSA:` Do you sweat?
6. `User:` Sure. Is there any organic quilts ?
7. `VSA:` Definitely, but which dimensions are you looking for?
8. `User:` 200x200. Is there any one within a budget of €350?
9. `VSA:` Yes, it cost €239 since the dimensions are 200x200, it is nonallergenic and organic.
10. `User:` I buy it.
11. `VSA:` Since it is still within your budget, I may also suggest you to buy a "four seasons" one that cost €279?
12. `User:` Good idea.
13. `VSA:` What about a quilt cover that cost €99 only? still in your budget.
14. `User:` No, thanks.

Fig. 1: Virtual Seller Agent Vs. User: Before-Sale & Sale scenarios

5 CSO Language Processor

The CSO Language Processor [9] is the technology upon which our dialogue system is built.

The CSO Language Processor provided by Artificial Solutions[1] allows to produce virtual dialogue assistants such as Anna[2] and numerous other systems in more than 20 different languages. Elbot, which has won the Loebner Prize[3] in 2008, is built upon this technology.

This technology manages users' sessions, handles misspellings and it also contains a language dependent preprocessing feature. In accordance with the dialogue state, it selects and carries out the best dialogue move. Additionally, this technology is able to interact with a back-end system, (e.g., databases), to hand out answer document for requesting application/front end and to write log files for analysis.

The inputs of the language processor are the user queries, (i.e., the user's identity and his text inputs). After the identification of the session, the inputs are divided in sentences and words and the spelling is corrected. Another phase is carried out wherein an interpretation of the inputs is made: an answer retrieval for each sentences of the user's inputs based on some **interaction rules in a knowledge base**. Finally, the answer is selected and generated by replacing some template variables.

[1] http://www.artificial-solutions.com
[2] http://www.ikea.com
[3] http://www.loebner.net/Prizef/loebner-prize.html

The interaction rules combine the meaning of the user's inputs and the dialogue state to define the conditions under which a dialogue move may be uttered. A given move can only be performed if the conditions are completely fulfilled.

However, the core of the CSO Language Processor is an inference engine that implements forward-chaining and so reactionary. Therefore, the language processor only makes it possible to respond to a user's queries and not to initiate or lead a sales-driven conversations. consequently, in order for us to make CSO proactive and enable it to go through sales-driven encounters, we introduce in the next section a formal framework for possible sale-driven dialogue management that can be adapted by virtual agents.

6 Dialectical system

A dialogue is a social interaction amongst self-interested parties intended to reach a common goal. In this section, we present how our game-based social model [8] handles the forseen conversation between a user and a virtual seller agent (VSA).

A dialectical system is a formal system that regulate persuasion dialogue, (See [19] for an overview). According to the game metaphor for social interactions, the parties are players which utter moves according to social rules.

Definition 1 (Dialectical system). *Let us consider $\mathcal{L}$ a common object language and $\mathcal{ACL}$ a common agent communication language. A **dialectical system** is a tuple $\mathtt{DS}=\langle \mathtt{P}, \Omega_M, \mathtt{H}, \mathtt{T}, \mathtt{proto}, \mathtt{Z}\rangle$ where:*

- *$\mathtt{P}$ is a set of participants called players;*
- *$\Omega_M \subseteq \mathcal{ACL}$ is a set of well-formed moves;*
- *$\mathtt{H}$ is a set of histories, the sequences of well-formed moves s.t. the speaker of a move is determined at each stage by the turn-taking function $\mathtt{T}$ and the moves agree with the dialogue-game protocol $\mathtt{proto}$;*
- *$\mathtt{T}$: $\mathtt{H} \rightarrow \mathtt{P}$ is the turn-taking function;*
- *$\mathtt{proto}$: $\mathtt{H} \rightarrow 2^{\Omega_M}$ is the function determining the legal moves which are allowed to expand an history;*
- *$\mathtt{Z}$ is the set of dialogues, i.e. the terminal histories.*

Here, $\mathtt{DS}$ reflects the formalization of social interactions between players uttering moves during a dialogue. Each dialogue is a maximally long sequence of moves. Later to that, we specify informally the elements of $\mathtt{DS}$ for bilateral negotiation and information-seeking.

In our scenario, there are two players: the initiator `init` and, the responder `resp`, which utter moves each in turn. Since we address a proactive dialogue system, we consider the initiator to be a VSA. The **syntax** of moves is in conformance with a common **agent communication language**, $\mathcal{ACL}$. A move at time t: has an identifier, $\mathtt{mv}_t$; is uttered by a speaker ($\mathtt{sp}_t \in \mathtt{P}$) and the speech act is composed of a locution $\mathtt{loc}_t$ and a content $\mathtt{content}_t$.

The possible locutions are: `question`, `assert`, `unknow`, `introduce`, `request`, `accept` and `reject`. The content consists of all instances of the following schemata

1.	VSA:	question(is(product, quilt) because search(user, product)).
2.	User:	assert(is(product, quilt)).
3.	VSA:	question(is(user, allergic)).
4.	User:	assert(is(user, allergic)).
5.	VSA:	question(is(user, sweat)).
6.	User:	assert(is(user, sweat)).
		question(is(product, organic)).
7.	VSA:	assert(is(product, organic)).
		question(dimension(product, 200, 200)).
8.	User:	assert(dimension(product, 200, 200)).
		question(budget(product, 350)).
9.	VSA:	introduce(is(product, quilt) because budget(product, 239) and is(product, nonallergenic)) and is(product, organic)).
10.	User:	accept(is(product, quilt).
11.	VSA:	introduce(is(product, quilt) because budget(product, 279) and is(product, nonallergenic)) and is(product, fourseasons)).
12.	User:	accept(is(product, quilt).
13.	VSA:	introduce(is(product, quilt)) and is(product, quiltcovers) because budget(product, 333.90) and is(product, nonallergenic)) and is(product, fourseasons)).
14.	User:	reject(is(product, quiltcovers).

Fig. 2: A Possible Scenario Formalization

"S (because S')" where S (eventually S') is a set of sentences in the common object language, $\mathcal{L}$. Actually, natural language utterances are interpreted/generated by the language dependent preprocessing of CSO (See Section 5). A move is an abstract representation of natural language utterances.

The dialogue in Fig 2 depicts a possible formalization of the natural language dialogue of Fig 1. It is worth noticing here that each utterance can contain more than one move.

In Fig. 3, we present our dialogue-game protocols by means of a deterministic finite-state automaton. An information-seeking dialogue begins with a question. The legal responding speech acts are `assert` and `unknow`. Two possible cases can occur: i) the dialogue is a failure if it is closed by an `unknow`; ii) the dialogue is a success if it is closed by an `assert`. A negotiation dialogue either begins with an offer from the VSA through the speech act `introduce` or the offer is suggested by the user through the speech act `request`. The legal responding speech acts are `accept` and `reject`. Here, the possibly occurring cases are: i) the dialogue is a failure if it is closed by a `reject`; ii) the dialogue is a success if it is closed by an `accept`.

The strategy interfaces with the dialogue-game protocol through the condition mechanism of utterances for a move. For example, at a certain point in the dialogue the VSA is able to send `introduce` or `question`. The choice of which locution and which content to send is depending on the VSA's strategy. For instance, the VSA is **benevolent** in the dialogue represented in Fig 2 since he first

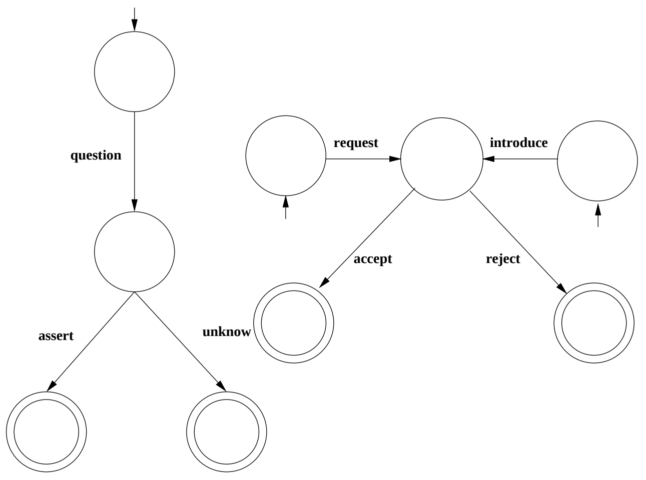


Fig. 3: Dialogue-game protocol for information-seeking (on the left), and negotiation (on the right)

attempts to identify the dialogue's party needs, he continues with the product selection phase and then it terminates with the sale dialogue. An **aggressive** agent would consider the sale prior to anything whether the before-sale tasks have been performed or not.

7 Arguing over utterances

In this section, we present how our computational model of argumentation for decision making [7] handles the dialogue strategy in order to generate and evaluate utterances.

In our framework, the knowledge is represented by a logical theory built upon an underlying logic-based language. In this language we distinguish between several different categories of predicate symbols. We use *goals* to represent the possible objectives of the decision making process (e.g. the dialogue to perform), *decisions* an agent can adopt (e.g. the move to utter) and a set of predicate symbols for *beliefs* (e.g. the previous utterance).

Assumptions here are required to carry on the reasoning process with incomplete knowledge, (e.g. some information about user's needs are missing), and we need to express *preferences* between different goals (e.g. some dialogues are prior depending on the agent's strategy). Finally, we allow the representation of explicit *incompatibilities* between goals, decisions and beliefs.

Definition 2 (Decision framework). *A* decision framework *is a tuple* $DF = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P} \rangle$*, where:*

- $\mathcal{DL} = \mathcal{G} \cup \mathcal{D} \cup \mathcal{B}$ *is a set of predicate symbols called the* ***decision language****, where we distinguish between goals (*$\mathcal{G}$*), decisions (*$\mathcal{D}$*) and beliefs (*$\mathcal{B}$*);*

– *$\mathcal{A}sm$ is a set of atomic formulae built upon predicates in $\mathcal{DL}$ called* ***assumptions****;*
– *$\mathcal{I}$ is the* ***incompatibility relation****, i.e. a binary relation over atomic formulae in $\mathcal{G}$, $\mathcal{B}$ or $\mathcal{D}$. We require $\mathcal{I}$ to be asymmetric;*
– *$\mathcal{T}$ is a logic* ***theory*** *built upon $\mathcal{DL}$; statements in $\mathcal{T}$ are clauses, each of them has a distinguished* name*;*
– *$\mathcal{P} \subseteq \mathcal{G} \times \mathcal{G}$ is the* ***priority*** *relation, namely a transitive, irreflexive and asymmetric relation over atomic formulae in $\mathcal{G}$.*

In our framework, we consider multiple objectives which may or not be fulfilled by a set of decisions under certain circumstances. Additionally, we explicitly distinguish *assumable* (respectively *non-assumable*) literals which can (respectively cannot) be assumed to hold, as long as there is no evidence to the contrary. Decisions as well as some beliefs can be assumed. In this way, `DF` can model the incompleteness of knowledge.

The most natural way to represent conflicts in our object language is throughout some forms of logical negation. We consider two types of negation, as usual, (e.g., in extended logic programming), namely *strong negation* $\neg$ (also called *explicit* or *classical negation*), and *weak negation* $\sim$, also called *negation as failure.* As a consequence we will distinguish between strong literals, i.e. atomic formula possibly preceded by strong negation, and weak literals, i.e. literals of the form $\sim L$, where L is a strong literal. The intuitive meaning of a strong literal $\neg L$ is "L is definitely not the case", while $\sim L$ intuitively means "There is no evidence that L is the case". The set $\mathcal{I}$ of incompatibilities contains some *default* incompatibilities related to negation on the one hand, and to the nature of decision predicates on the other hand. Indeed, given an atom A, we have $A \ \mathcal{I} \ \neg A$ as well as $\neg A \ \mathcal{I} \ A$. Moreover, $L \ \mathcal{I} \sim L$, whatever L is, representing the intuition that L is evidence to the contrary of $\sim L$. Notice, however, that we do not have $\sim L \ \mathcal{I} \ L$, as in the spirit of weak negation.

Other default incompatibilities are related to decisions, since different alternatives for the same decision predicate are incompatible with one another. Hence, $D(a_1) \ \mathcal{I} \ D(a_2)$ and $D(a_2) \ \mathcal{I} \ D(a_1)$, D being a decision predicate in $\mathcal{D}$, and a_1 and a_2 being different constants representing different[4] alternatives for D. Depending on the particular decision problem being represented by the framework, $\mathcal{I}$ may contain further non-default incompatibilities. For instance, we may have $g \ \mathcal{I} \ g'$, where g, g' are different goals.

To summarize, the incompatibility relation captures the conflicts, either default or domain dependent, amongst decisions, beliefs and goals. The incompatibility relation can be easily lifted to set of sentences. We say that two sets of sentences Φ_1 and Φ_2 are *incompatible* (still denoted by $\Phi_1 \ \mathcal{I} \ \Phi_2$) if there is a sentence ϕ_1 in Φ_1 and a sentence ϕ_2 in Φ_2 such that $\phi_1 \ \mathcal{I} \ \phi_2$.

A theory gathers the statements about the decision problem.

Definition 3 (Theory). *A* theory *$\mathcal{T}$ is an extended logic program, i.e a finite set of rules R: $L_0 \leftarrow L_1, \ldots, L_j, \sim L_{j+1}, \ldots, \sim L_n$ with $n \geq 0$, each L_i (with*

[4] Notice that in general a decision can be addressed by more than two alternatives.

$i \geq 0$) being a strong literal in $\mathcal{L}$. R, called the unique name *of the rule, is an atomic formula of $\mathcal{L}$. All variables occurring in a rule are implicitly universally quantified over the whole rule. A rule with variables is a scheme standing for all its ground instances.*

To simplify, we assume that names of rules are neither in the bodies nor in the head of the rules thus avoiding self-reference problems. We assume that the elements in the body of rules are independent. Besides, we suppose the decisions do not influence the beliefs and the decisions have no side effects.

In order to evaluate the relative importance of goals, we consider the *priority* relation $\mathcal{P}$ over the goals in $\mathcal{G}$, which is transitive, irreflexive and asymmetric. $G_1 \mathcal{P} G_2$ can be read "G_1 has priority over G_2". There is no priority between G_1 and G_2, either because G_1 and G_2 are *ex æquo* (denoted $G_1 \simeq G_2$), or because G_1 and G_2 are not comparable.

We consider the dialogue formalized in Section 6. The generation and the evaluation of utterances by the VSA are captured by a *decision framework* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, \mathcal{P} \rangle$ where:

- the decision language $\mathcal{DL}$ distinguishes,
  - a set of **goals** $\mathcal{G}$. This set of literals identifies various motivations for driving the possible dialogues, negotiation ($\mathtt{negotiating}(\mathtt{product})$) and information-seeking ones for product selection ($\mathtt{infoseeking}(\mathtt{product})$) or need identification ($\mathtt{infoseeking}(\mathtt{user})$),
  - a set of **decisions** $\mathcal{D}$. This set of literals identifies the possible utterances (e.g. $\mathtt{send}(\mathtt{question}(\mathtt{is}(\mathtt{user}, \mathtt{allergic})))$),
  - a set of **beliefs**, i.e. a set of literals identifying various situations identifying the possible queries of the user, (e.g. $\mathtt{receive}(\mathtt{question}(\mathtt{is}(\mathtt{product}, \mathtt{nonallergenic})))$, behavior through the website (e.g. $\mathtt{search}(\mathtt{user}, \mathtt{quilt})$) or the knowledge about the product/needs information (e.g. $\mathtt{is}(\mathtt{user}, \mathtt{allergic})$);
- the set of assumptions $\mathcal{A}sm$ contains the possible decisions and the missing information about the user, (e.g. $\sim \mathtt{is}(\mathtt{user}, \mathtt{allergic})$), or the product, (e.g. $\sim \mathtt{is}(\mathtt{product}, \mathtt{nonallergenic})$);
- the incompatibility relation $\mathcal{I}$ is trivially defined. For instance,
  $\mathtt{send}(x)\ \mathcal{I}\ \mathtt{send}(y)$, with $x \neq y$
  $\mathtt{infoseeking}(topic_1)\ \mathcal{I}\ \mathtt{infoseeking}(topic_2)$, with $topic_1 \neq topic_2$
  $\mathtt{negotiating}(topic_1)\ \mathcal{I}\ \mathtt{infoseeking}(topic_2)$ whatever $topic_1$ and $topic_2$ are
- the theory $\mathcal{T}$ contains the rules in Table 2;
- If the VSA is benevolent, then the priority is defined such that:
  $\mathtt{infoseeking}(\mathtt{user}) \mathcal{P} \mathtt{infoseeking}(\mathtt{product})$ and
  $\mathtt{infoseeking}(\mathtt{product}) \mathcal{P} \mathtt{negotiating}(\mathtt{product})$.
  If the VSA is aggressive, then the priority is defined such that:
  $\mathtt{negotiating}(\mathtt{product}) \mathcal{P} \mathtt{infoseeking}(\mathtt{product})$ and
  $\mathtt{infoseeking}(\mathtt{product}) \mathcal{P} \mathtt{infoseeking}(\mathtt{user})$.

Table 2: The rules of a Virtual Seller Agent (VSA)

$\mathtt{r}_{11}$: infoseeking(user)	$\leftarrow$ send(question(is(user, allergic))), $\sim$ is(user, allergic), is(product, quilt), $\sim$ receive(x)
$\mathtt{r}_{12}$: infoseeking(user)	$\leftarrow$ send(question(is(user, sweat))), $\sim$ is(user, sweat), is(product, quilt), $\sim$ receive(x)
$\mathtt{r}_{21}$: infoseeking(product)	$\leftarrow$ send(question(is(product, quilt))), search(user, quilt), $\sim$ is(product, quilt)
$\mathtt{r}_{22}$: infoseeking(product)	$\leftarrow$ send(question(is(product, nonallergenic))), $\sim$ is(product, nonallergenic), $\sim$ receive(x)
$\mathtt{r}_{23}$: infoseeking(product)	$\leftarrow$ send(question(is(product, organic))), $\sim$ is(product, organic), $\sim$ receive(x)]
$\mathtt{r}_{24}$: infoseeking(product)	$\leftarrow$ send(question(dimension(product, x, y))), $\sim$ dimension(product, x, y), $\sim$ receive(z)
$\mathtt{r}_{25}$: infoseeking(product)	$\leftarrow$ send(question(budget(product, x))), $\sim$ budget(product, x), $\sim$ receive(y)
$\mathtt{r}_{26}$: infoseeking(product)	$\leftarrow$ send(assert(is(x, y)), receive(question(is(x, y))), is(x, y)
$\mathtt{r}_{27}$: infoseeking(product)	$\leftarrow$ send(assert($\neg$is(x, y)), receive(question(is(x, y))), $\neg$is(x, y)
$\mathtt{r}_{28}$: infoseeking(product)	$\leftarrow$ send(unknow(is(x, y)), receive(question(is(x, y))), $\sim$ is(x, y)
$\mathtt{r}_{29}$: negotiating(product)	$\leftarrow$ send(introduce(product)), budget(product, y)
$\mathtt{r}_{31}$: budget(product, 350)	$\leftarrow$ is(product, nonallergenic), *is*(product, organic), dimension(product, 200, 200)
$\mathtt{r}_{32}$: is(product, nonallergenic)	$\leftarrow$ is(user, allergic)
$\mathtt{r}_{33}$: is(product, organic)	$\leftarrow$ is(user, sweat)

Our formalization allows to capture the incomplete representation of a decision problem with assumable beliefs. It provides a knowledge base on top of which arguments are built in order to reach decisions. We adopt here a tree-like structure for arguments.

Definition 4 (Argument). *Let $DF = \langle \mathcal{DL}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, \mathcal{P}, \mathcal{RV} \rangle$ be a decision framework. An* ***argument*** *$\bar{a}$ deducing the* ***conclusion*** *$c \in \mathcal{DL}$ (denoted $\mathtt{conc}(\bar{a})$) supported by a set of* ***assumptions*** *A in $\mathcal{A}sm$ (denoted $\mathtt{asm}(\bar{a})$) is a tree where the root is c and each node is a sentence of $\mathcal{DL}$. For each node :*

- *if the node is a leaf, then it is either an assumption in A or $\top$[5];*
- *if the node is not a leaf and it is $\alpha \in \mathcal{DL}$, then there is an inference rule $\alpha \leftarrow \alpha_1, \dots, \alpha_n$ in $\mathcal{T}$ and,*
  - *either $n = 0$ and $\top$ is its only child,*
  - *or $n > 0$ and the node has n children, $\alpha_1, \dots, \alpha_n$.*

The sentences *of $\bar{a}$ (denoted $\mathtt{sent}(\bar{a})$) is the set of literals of $\mathcal{DL}$ in the bodies/heads of the rules including the assumptions of $\bar{a}$. We write $\bar{a} : A \vdash \alpha$ to denote an argument $\bar{a}$ such that $\mathtt{conc}(\bar{a}) = \alpha$ and $\mathtt{asm}(\bar{a}) = A$. The set of arguments built upon DF is denoted by $\mathcal{A}(DF)$.*

[5] $\top$ denotes the unconditionally true statement.

Arguments are built by reasoning backwards. Additionally, arguments interact with one another, and consequently, we reach to define the following attack relation.

Definition 5 (Attack relation). *Let* $\texttt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P} \rangle$ *be a decision framework, and* $\bar{a}, \bar{b} \in \mathcal{A}(\texttt{DF})$ *be two arguments.* $\bar{a}$ $\texttt{attacks}$ $\bar{b}$ *iff* $\texttt{sent}(\bar{a})\ \mathcal{I}\ \texttt{sent}(\bar{b})$.

This relation encompasses both the direct (often called *rebuttal*) attack due to the incompatibility of the conclusions, and the indirect (often called *undermining*) attack, (i.e., directed to a "subconclusion").

Since the goals promoted by arguments have different priorities, the arguments interact with one another. For this purpose, we define the strength relation between concurrent arguments. Arguments are *concurrent* if their conclusions are identical or incompatible.

Definition 6 (Strength relation). *Let* $\texttt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P} \rangle$ *be a decision framework and* $\bar{a}_1, \bar{a}_2 \in \mathcal{A}(\texttt{DF})$ *be two arguments which are concurrent.* $\bar{a}_1$ *is stronger than* $\bar{a}_2$ *(denoted* $\bar{a}_1 \mathcal{P} \bar{a}_2$*) iff* $\texttt{conc}(\bar{a}_1) = g_1 \in \mathcal{G}$, $\texttt{conc}(\bar{a}_2) = g_2 \in \mathcal{G}$ *and* $g_1 \mathcal{P} g_2$.

Due to the definition of $\mathcal{P}$ over $\mathcal{T}$, the relation $\mathcal{P}$ is transitive, irreflexive and asymmetric over $\mathcal{A}(\texttt{DF})$.

The attack relation and the strength relation can be combined to adopt Dung's calculus of opposition as in [20]. We distinguish between one argument attacking another, and that attack succeeding due to the strength of arguments.

Definition 7 (Defeat relation). *Let* $\texttt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P} \rangle$ *be a decision framework and* $\bar{a}$ *and* $\bar{b}$ *be two structured arguments.* $\bar{a}$ defeats $\bar{b}$ *iff:*

1. $\bar{a}$ $\texttt{attacks}$ $\bar{b}$*;*
2. *and it is not the case that* $\bar{b} \mathcal{P} \bar{a}$*.*

Similarly, we say that a set S *of structured arguments defeats a structured argument* $\bar{a}$ *if* $\bar{a}$ *is defeated by one argument in* S*.*

Let us consider this example:

Example 1 (Defeat relation). Let us consider the situation after the second move in the dialogue represented in Fig. 1.

The arguments ā concludes `infoseeking(user)` since the VSA can ask to the user if he is allergic, (i.e. `question(is(user, allergic))`), the VSA is not yet aware about it, (i.e. $\sim$ `is(user, allergic)`), the user is looking for a quilt,(i.e. `is(product, quilt)`), and the user did not query the VSA, (i.e. $\sim$ `receive`(x)). The argument b̄ concludes `infoseeking(product)` since the VSA can ask to the user if the product must be nonallergenic,
(i.e. `send(question(is(product, nonallergenic)))`), the VSA is not yet aware about it (i.e. $\sim$ `is(product, nonallergenic)`) and the user did not query the VSA ($\sim$ `receive`(x)) . While ā is built upon $\texttt{r}_{11}$, b̄ is built upon $\texttt{r}_{22}$. Since these

arguments suppose different decisions, they attack each others. If the VSA is benevolent, it is not the case that $\texttt{infoseeking}(\texttt{product})\mathcal{P}\texttt{infoseeking}(\texttt{user})$ and so $\bar{\text{a}}$ defeats $\bar{\text{b}}$. If the VSA is aggressive, it is not the case that $\texttt{infoseeking}(\texttt{user})\mathcal{P}\texttt{infoseeking}(\texttt{product})$ and so $\bar{\text{a}}$ defeats $\bar{\text{b}}$.

In our argumentation-based approach for dialogue strategy, arguments motivate decisions and they can also be defeated by other arguments. Formally, our argumentation framework (AF for short) is defined as follows.

Definition 8 (AF). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, \mathcal{P}\rangle$ *be a decision framework. The* argumentation framework for decision making *built upon* $\mathtt{DF}$ *is a pair* $\mathtt{AF} = \langle \mathcal{A}(\mathtt{DF}), \ \mathtt{defeats} \ \rangle$ *where* $\mathcal{A}(\mathtt{DF})$ *is the finite set of arguments built upon* $\mathtt{DF}$ *as defined in Definition 8, and* $\mathtt{defeats} \subseteq \mathcal{A}(\mathtt{DF}) \times \mathcal{A}(\mathtt{DF})$ *is the binary relation over* $\mathcal{A}(\mathtt{DF})$ *as defined in Definition 7.*

We adapt Dung's extension-based semantics in order to analyze whenever a set of arguments can be considered as subjectively justified with respect to the agent's priority.

Definition 9 (Semantics). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, \mathcal{P}\rangle$ *be a decision framework and* $\mathtt{AF} = \langle \mathcal{A}(\mathtt{DF}), \ \mathtt{defeats} \ \rangle$ *be our argumentation framework for decision making. For* $\mathtt{S} \subseteq \mathcal{A}(\mathtt{DF})$ *a set of arguments, we say that:*

- $\mathtt{S}$ *is* conflict-free *iff* $\forall \bar{a}, \bar{b} \in \mathtt{S}$ *it is not the case that* $\bar{a}$ *defeats* $\bar{b}$*;*
- $\mathtt{S}$ *is* admissible *iff* $\mathtt{S}$ *is conflict-free and* $\mathtt{S}$ *defeats every argument* $\bar{a}$ *such that* $\bar{a}$ *defeats some argument in* $\mathtt{S}$*;*

Here, we only consider admissibility but other Dung's extension-based semantics [21] can easily be adapted.

Formally, given an argument $\bar{\text{a}}$, let

$$\texttt{dec}(\bar{\text{a}}) = \{D(a) \in \texttt{asm}(\bar{\text{a}}) \mid D \text{ is a decision predicate}\}$$

be the set of decisions supported by the argument $\bar{\text{a}}$.

The decisions are *suggested* to reach a goal if they are supported by an argument concluding this goal and this argument is a member of an admissible set of arguments.

Definition 10 (Credulous decisions). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \mathcal{I}, \mathcal{T}, \mathcal{P}\rangle$ *be a decision framework,* $g \in \mathcal{G}$ *be a goal and* $\mathtt{D} \subseteq \mathcal{D}$ *be a set of decisions. The decisions* $\mathtt{D}$ ***credulously argue for*** g *iff there exists an argument* $\bar{a}$ *in an admissible set of arguments such that* $\texttt{conc}(\bar{a}) = g$ *and* $\texttt{dec}(\bar{a}) = \mathtt{D}$*. We denote* $\texttt{val}_c(\mathtt{D})$ *the set of goals in* $\mathcal{G}$ *for which the set of decisions* $\mathtt{D}$ *credulously argues.*

It is worth noticing here that the decisions that credulously argue for a goal cannot contain mutual exclusive alternatives for the same decision predicate. This is due to the fact that an admissible set of arguments is conflict-free.

If we consider the arguments $\bar{\text{a}}$ and $\bar{\text{b}}$ supporting the decisions $D(a)$ and $D(b)$ respectively where a and b are mutually exclusive alternatives, we have $D(a) \ \mathcal{I} \ D(b)$ and $D(a) \ \mathcal{I} \ D(b)$ and so, either $\bar{\text{a}}$ $\mathtt{defeats}$ $\bar{\text{b}}$ or $\bar{\text{b}}$ $\mathtt{defeats}$ $\bar{\text{a}}$ or both of them depending on the strength of these arguments.

Proposition 1 (Mutual exclusive alternatives). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P} \rangle$ *be a decision framework,* $g \in \mathcal{G}$ *be a goal and* $\mathtt{AF} = \langle \mathcal{A}(\mathtt{DF}), \ \mathtt{defeats}\ \rangle$ *be the argumentation framework for decision making built upon* $\mathtt{DF}$*. If* $\mathtt{S}$ *be an admissible set of arguments such that, for some* $\bar{a} \in \mathtt{S}$*,* $g = \mathtt{conc}(\bar{a})$ *and* $D(a) \in \mathtt{asm}(\bar{a})$*, then* $D(b) \in \mathtt{asm}(\bar{a})$ *iff* $a = b$*.*

However, it is worth highlighting here the fact that mutual exclusive decisions can be suggested for the same goal through different admissible set of arguments. This case reflects the credulous nature of our semantics.

Definition 11 (Skeptical decisions). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{P}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P}, \mathcal{RV} \rangle$ *be a decision framework,* $g \in \mathcal{G}$ *be a goal and* $\mathtt{D} \subseteq \mathcal{D}$ *be a set of decisions. The decisions* $\mathtt{D}$ ***skeptically argue for*** *g iff for all admissible set of arguments* $\mathtt{S}$ *such that for some arguments* $\bar{a}$ *in* $\mathtt{S}$ $\mathtt{conc}(\bar{a}) = g$*, then* $\mathtt{dec}(\bar{a}) = \mathtt{D}$*. We denote* $\mathtt{val}_s(\mathtt{D})$ *the set of goals in* $\mathcal{G}$ *for which the set of decisions* $\mathtt{D}$ *skeptically argues.*

Due to the uncertainties, some decisions satisfy goals for sure if they skeptically argue for them, or some decisions can possibly satisfy goals if they credulously argue for them. While the first case is required for convincing a risk-averse agent, the second case is enough to convince a risk-taking agent. Since some ultimate choices amongst various justified sets of alternatives are not always possible, we will consider in this paper only risk-taking agents.

Since agents can consider multiple objectives which may not be fulfilled all together by a set of non-conflicting decisions, high-ranked goals must be preferred to low-ranked goals.

Definition 12 (Preferences). *Let* $\mathtt{DF} = \langle \mathcal{DL}, \mathcal{A}sm, \ \mathcal{I}\ , \mathcal{T}, \mathcal{P}, \mathcal{RV} \rangle$ *be a decision framework. We consider* $\mathtt{G}$*,* $\mathtt{G}'$ *two set of goals in* $\mathcal{G}$ *and* $\mathtt{D}$*,* $\mathtt{D}'$ *two set of decisions in* $\mathcal{D}$*.* $\mathtt{G}$ *is* ***preferred*** *to* $\mathtt{G}$ *(denoted* $\mathtt{G}\mathcal{P}\mathtt{G}'$*) iff*

1. $\mathtt{G} \supseteq \mathtt{G}'$*, and*
2. $\forall g \in \mathtt{G} \setminus \mathtt{G}'$ *there is no* $g' \in \mathtt{G}'$ *such that* $g'\mathcal{P}g$*.*

$\mathtt{D}$ *is* ***preferred*** *to* $\mathtt{D}'$ *(denoted* $\mathtt{D}\mathcal{P}\mathtt{D}'$*) iff* $\mathtt{val}_c(\mathtt{D})\mathcal{P}\mathtt{val}_c(\mathtt{D}')$*.*

Formally, let
$\mathcal{AD} = \{\mathtt{D} \mid \mathtt{D} \subseteq \mathcal{D}$ such that $\forall \mathtt{D}' \subseteq \mathcal{D}$ it is not the case that $\mathtt{val}_c(\mathtt{D}')\ \mathcal{P}\ \mathtt{val}_c(\mathtt{D})\}$
be the decisions which can be accepted by the agent. Additionally, let
$\mathcal{AG} = \{\mathtt{G} \mid \mathtt{G} \subseteq \mathcal{G}$ such that $\mathtt{G} = \mathtt{val}_c(\mathtt{D})\ \}$
be the goals which can be reached by the agent.

Let us consider now the VSA's decision problem after the second move.

Example 2 (Semantics). The argument $\bar{\mathrm{a}}$ (respectively $\bar{\mathrm{b}}$) (described in Example 1), concludes `infoseeking(user)` (respectively `infoseeking(product)`). Actually, the decisions {`send(question(is(user, allergic)`} credulously argue for `infoseeking(user)` and the decisions
{`send(question(is(product, nonallergenic)`} credulously argue for `infoseeking(product)`. If the VSA is benevolent, then
{`send(question(is(user, allergic)`} is an acceptable set of decisions. If the VSA is aggressive, then {`send(question(is(product, nonallergenic)`} is an acceptable set of decisions.

8 Related Works

Amgoud & Prade in [22] are presenting a general and abstract argumentation framework for multi criteria decision making. This framework captures the mental states (goals, beliefs and preferences) of the decision makers. Therefore, in their framework the arguments are prescribing actions to reach goals if theses actions are feasible while certain circumstances are true. These arguments - *that eventually conflict* - are balanced according to their strengths. The argumentation framework we proposed earlier in this paper is conforming with this approach while being more specific and concrete.

The argumentation-based decision making process envisaged in [22] is divided into different steps where the arguments are successively constructed, weighted, confronted and evaluated. However, the computations we proposed earlier in this paper go through the construction of arguments, the construction of counterarguments, the evaluation of the generated arguments and the relaxation of preferences for making concessions. It is also worth noticing here that: a) the model we propose is unique in making it finally possible to concede, b) Our argumentation-based decision process suggest some decisions even if low-ranked goals cannot be reached.

Bench-Capon & Prakken formalize in [23] defeasible argumentation for practical reasoning. As in [22], they select the best course of actions by confronting and evaluating arguments. Bench-Capon & Prakken focus on the abductive nature of practical reasoning which is directly modelled within in our framework.

Kakas & Moraits propose in [24] an argumentation-based framework for decision making of autonomous agents. For this purpose, the knowledge of the agent is split and localized in different modules representing different capabilities. Whereas [24] is committed to one argumentation semantics, we can still deploy our framework for a number of such semantics by relying on assumption-based argumentation.

Finally, to the best of our knowledge, few implementation of argumentation over actions exist. CaSAPI[6] [25] and DeLP[7] [26] are restricted to the theoretical reasoning. GORGIAS[8] [27] implements an argumentation-based framework to support the decision making of an agent within a modular architecture.

9 Conclusions

In this paper, we have presented a dialogue management system that applies argumentation for generating and evaluating utterances. The agent start the conversation with the prior task which can consist of the need identification, the product selection or the negotiation depending on its strategy. During the dialogue, a proactive agent can query the user. Additionally, it can introduce a

[6] http://www.doc.ic.ac.uk/∼dg00/casapi.html
[7] http://lidia.cs.uns.edu.ar/DeLP
[8] http://www.cs.ucy.ac.cy/∼nkd/gorgias/

product to sell and justify this choice depending on the information collected in the previous steps.

In order for us to implement an agent's reasoning method we are considering MARGO[9] (A Multiattribute ARGumentation framework for Opinion explanation), which is an argumentation-based mechanism for decision-making [7]. We are currently rewritting MARGO in Java so that issues related to improving its performance, (i.e., the response time), and its scalability, (i.e., the number of rules which can be managed), are better tackled. This work is required to provide an industrial application rather than a research prototype. Besides, we need to interface this argumentation-based engine with the CSO Artificial Solutions' Language Processor in order to build conversational agents which are proactive in different selling situations.

Although the negotiation dialogue model we proposed allows single-sellings through the exchange of proposals and counter-proposals. However, we are currently working on an extension that will address cross-selling and up-selling.

Acknowledgements

This work is supported by the Ubiquitous Virtual Seller (VVU) project that was initiated by the Competitivity Institute on Trading Industries (PICOM).

References

1. Hof, R., Green, H., Himmelstein, L.: Now it's YOUR WEB. BusinessWeek (October 1998) 68–75
2. Poong, Y., Zaman, K.U., Talha, M.: E-commerce today and tomorrow: a truly generalized and active framework for the definition of electronic commerce. In: Proc. of the 8th international conference on Electronic commerce (ICEC), Fredericton, New Brunswick, Canada, ACM (2006) 553–557
3. Palopoli, L., Rosaci, D., Ursino, D.: Agents' roles in B2C e-commerce. AI Communications **19**(2) (2006) 95–126
4. Wooldridge, M., Jennings, N.: Intelligent agents: Theory and practice. Knowledge Engineering Review **10** (1995) 115–52
5. Isbister, K., Doyle, P.: Design and evaluation of embodied conversational agents: A proposed taxonomy. In: Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Budapest, Hungary (2002)
6. Rist, T., Andr, E., Baldes, S., Gebhard, P., Klesen, M., Rist, P., Schmitt, M.: A review of the development of embodied presentation agents and their application fields. In: Life-Like Characters Tools, Affective Functions, and Applications, Springer (2003) 377–404
7. Morge, M., Mancarella, P.: The hedgehog and the fox. An argumentation-based decision support system. In: Proc. of the Fourth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS). (2007) 55–68
8. Morge, M., Mancarella, P.: Assumption-based argumentation for the minimal concession strategy. In: Proc. of the 6th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS), Budapest, Hungary (2009)

[9] http://margo.sourceforge.net

9. Roberts, F., Gülsdorff, B.: Techniques of dialogue simulation. In: Proc of the 7th International Conference on Intelligent Virtual Agents. Volume 4722 of Lecture Note in Computer Science., Paris, France, Springer (2007) 420–421
10. George Ferguson, J.A.: Mixed-initiative systems for collaborative problem solving. AI Magazine **28**(2) (2007) 23–32
11. Rich, C., Sidner, C.L., Lesh, N.: COLLAGEN applying collaborative discourse theory to human-computer interaction. AI Magazine **22**(4) (2001) 15–25
12. Sadek, D.: Artemis Rational Dialogue Agent Technology: An Overview. In: Multi-Agent Programming. Springer-Verlag (2005) 217–225
13. Grosz, B.J., Sidner, C.L.: Plans for Discourse. In: Intentions and Plans in Communication and Discourse. Cohen, Morgan and Pollack edn. MIT press (1990)
14. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR), Morgan Kaufmann publishers Inc.: San Mateo, CA, USA (1991) 473–484
15. C, F.T.: FIPA ACL Communicative Act Library Specification. Component, Foundation for Intelligent Physical Agents (6-12 2002) http://fipa.org/specs/fipa00037/.
16. Breiter, P.: La communication orale coopérative : contribution à la modélisation et à la mise en œuvre d'un agent rationnel dialoguant. PhD thesis, Université de Paris Nord (1992)
17. Hamblin, C.L.: Fallacies. Methuen (1970)
18. Walton, D., Krabbe, E.: Commitment in Dialogue. SUNY Press (1995)
19. Prakken, H.: Formal systems for persuasion dialogue. The Knowledge Engineering Review **21** (2006) 163–188
20. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: Proc. of UAI, Madison, Wisconsin, USA., Morgan Kaufmann (1998) 1–7
21. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2) (1995) 321–357
22. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence Journal **173**(3-4) (2009) 413–436
23. Bench-Capon, T., Prakken, H.: Justifying actions by accruing arguments. In: Proc. of the 1st International Conference on Computational Models of Argument, IOS Press (2006) 247–258
24. Kakas, A., Moraitis, P.: Argumentative-based decision-making for autonomous agents. In: Proc. of AAMAS, ACM Press (2003) 883–890
25. Gartner, D., Toni, F.: CaSAPI: a system for credulous and sceptical argumentation. In: Proc. of ArgNMR. (2007) 80–95
26. García, A.J., Simari, G.R.: Defeasible logic programming: an argumentative approach. Theory and Practice of Logic Programming **4**(2) (2004) 95–138
27. Demetriou, N., Kakas, A.C.: Argumentation with abduction. In: Proc. of the 4th Panhellenic Symposium on Logic. (2003)

Argumentation System Allowing Suspend/Resume of an Argumentation Line

Kenichi OKUNO[1] * and Kazuko TAKAHASHI[2]

School of Science&Technology, Kwansei Gakuin University,
2-1, Gakuen, Sanda, 669-1337, JAPAN
`o.kenichi@gmail.com, ktaka@kwansei.ac.jp`

Abstract. This paper discusses an argumentation system that treats argumentation dynamically. We previously proposed a model for dynamic treatment of argumentation in which all lines of argumentation are executed in succession, with the change of the agent's knowledge base. This system was designed for grasping the behaviour of actual argumentation, but it has several limitations. In this paper, we propose an extended system in which these points are revised so that the model can more precisely simulate actual argumentation. In addition, we provide a simpler algorithm for judgement of given argumentation, which can be applied to make a strategy to win.
categories:D3.2 Prolog
keywords: computational model for argumentation, belief change, agent communication
general terms: algorithm, theory

1 Introduction

Argumentation is a model that evaluates arguments. It was originally investigated in legal reasoning. Dung's work on constructing a logical framework for argumentation and showing the relationships with nonmonotonic reasoning and logic programming [8] enlarged the application area of argumentation in the field of artificial intelligence (AI). As a result, formal models of argumentation have given much attention by AI researchers [4, 21]. These works include applications for defeasible logic programming [10, 6, 17, 16], belief revision [9, 18, 22] and so on. Argumentation is considered to be a powerful tool for logically analysing significant phenomena that appear in multiagent systems, such as negotiations, agreements and persuasion [14, 1], and for making computational models of the behaviour of multiagents [12]. Generally, argumentation proceeds between two agents by giving arguments that attack the opponent's argument in turn until one of them cannot attack any more. Finally, the loser accepts the winner's proposal. This process is usually represented in tree form [1, 10]. The root node is a proposed formula and each branch corresponds to a single argumentation line, namely, a sequence of arguments. Lots of argumentation systems have been proposed so far, but they considered the evaluation of a single argumentation line,

* Currently, JSOL Corporation

and cannot handle the dynamic properties of actual argumentation. By contrast, we have proposed a system that can treat the continuous evaluation of multiple argumentation lines [19, 20].

Let us consider an example of argumentation. According to many argumentation systems, a proposer P makes the first argument and a defeater C makes counterarguments. We suppose a situation in which a murderer P tells a lie: "I did not commit murder". A policeman C argues that P's statement is a lie. P_i and C_i represent P's and C's i-th utterances, respectively.

P_1: "I did not commit murder! There is no evidence!"
C_1: "There is evidence. We found your license near the scene."
P_2: "It's not evidence! I had my license stolen!"
C_2: "It is you who killed the victim. Only you were near the scene at the time of the murder."
P_3: "I didn't go there. I was at facility A at that time."
C_3: "At facility A? No, that's impossible. Facility A does not allow a person to enter without a license. You said that you had your license stolen, didn't you?"

Figure 1 shows the structure of this argumentation.

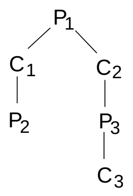


Fig. 1. Structure of argumentation

In this example, if argumentation proceeds along the left branch, and if C has no counterargument to P_2, then C continues a counterargument in the right branch which attacks P_1 from another side. Finally, C points out the contradiction between P's utterances and wins. P's utterance P_2 gives C new information and causes C to generate C_3.

To capture the behaviour in this example, we have proposed an argumentation system incorporating changes in an agent's knowledge base caused by the exchange of arguments [19, 20]. The goal of this system is to dynamically grasp argumentation by providing a model for actual argumentation. However, several points exist in which this earlier system does not reflect actual argumentation.

The first limitation is on the mechanism that brings up the settled matter in some argumentation line. Consider another argumentation:

P_1: "I did not commit murder! There is no evidence!"

C_2: "It is you who killed the victim. Only you were near the scene at the time of the murder."

P_3: "I didn't go there. I was at facility A at that time."

C_1: "There is evidence. We found your license near the scene."

P_2: "That's not evidence! I had my license stolen!"

C'_3: "That's strange. Facility A does not allow a person to enter without a license. You said that you were at facility A when the murder occurred. How did you enter?"

In this case, an argumentation first proceeds along the right branch, and then continues to the left branch, P's utterance P_2 gives C new information and causes C to generate C'_3 as a counterargument to P_3. C also points out the contradiction between P's utterances, and wins. Such a phenomenon frequently occurs in actual argumentation when each argumentation line is not so long. In our earlier system, this mechanism could not be handled. In this paper, we present a revised framework in which each argumentation line is considered as suspended but may be resumed afterward.

A second shortcoming is the equivalent rights of agents. In the earlier version, the defeater could continue an argumentation with the revised knowledge base after he/she loses one argumentation line, leading him/her to ultimately win the argumentation tree. However, the proposer loses the whole argumentation tree if he/she loses one argumentation line. In the revised version, we also allow the proposer to continue an argumentation after he/she loses one argumentation line.

The third point is also related to the equivalent rights of agents. In the earlier version, a proposer could not use disclosed information whereas a defeater could. This condition is unfair and unnatural. In the revised version, we introduce *commitment store* [13], a common knowledge base to store all the disclosed information, and both agents can use this knowledge base.

We extend the earlier system by addressing these three points so that it can more precisely simulate an actual argumentation and redefine the dynamic win/lose of an argumentation tree.

Moreover, we propose an algorithm for judging the result of an argumentation tree. This algorithm is simpler and easier to implement and it can be applied to formulate an argumentation strategy.

This paper is organised as follows. Section 2 provides the definitions of basic concepts such as argumentation and the argumentation tree. Section 3 proposes an extended model for argumentation incorporating changes in an agent's knowledge base and also presents an algorithm for the judgement of an argumentation tree. Section 4 provides an example of this algorithm. Section 5 outlines the major changes from our previous work and compares the proposed approach with related works. Finally, section 6 presents conclusions.

2 Argumentation

2.1 Argumentation Framework

We define an argumentation framework based on Dung [8].

Definition 1 (consistent) *Let Ψ be a set of formulas in propositional logic. If there does not exist ψ that satisfies both $\psi \in \Psi$ and $\neg\psi \in \Psi$, Ψ is said to be* consistent.

The knowledge base $\mathbf{K}_\mathrm{a}$ for each agent a is a finite set of propositional formulas. Note that $\mathbf{K}_\mathrm{a}$ is not necessarily consistent and may have no deductive closure; that is, a case may exist in which $\phi, \phi \rightarrow \psi \in \mathbf{K}_\mathrm{a}$ and $\psi \notin \mathbf{K}_\mathrm{a}$ hold. An agent a participates in argumentation using elements of $\mathbf{K}_\mathrm{a}$.

Definition 2 (support) *For a nonempty set of formulas Ψ and a formula ψ, if there exist ϕ, $\phi \rightarrow \psi \in \Psi$, then Ψ is said to be a* support *for ψ.*

Definition 3 (argument) *Let $\mathbf{K}_\mathrm{a}$ be a knowledge base for an agent a.* An argument of a *is a pair (Ψ, ψ) where Ψ is a subset of $\mathbf{K}_\mathrm{a}$, and $\psi \in \mathbf{K}_\mathrm{a}$ such that Ψ is the empty set or a consistent support for ψ,*

For an argument $A = (\Psi, \psi)$, Ψ and ψ are said to be *grounds* and *a sentence of* A, respectively. They are denoted by $Grounds(A)$ and $Sentence(A)$, respectively. $S(A)$ denotes $Grounds(A) \cup \{Sentence(A)\}$. If $\psi \in S(A)$, it is said that *a formula ψ is contained in an argument A.*

Similar to many argumentation systems, we adopt the concept of *preference* [3, 15]. Preferences are assigned to formulas depending on their strength, certainty and stability to avoid loops in the argumentation. Here, we assume that a formula is given a preference value based on some rules in advance regardless of the knowledge base in which it is contained, and adopt a simple definition for computing the preference of an argument. Although these definitions affect the result of the argumentation, we do not discuss the definitions here, since this aspect of argumentation is beyond the scope of this paper.

Definition 4 (preference) *Each formula is assigned a preference value. Let $\nu(\psi)$ be the preference for a formula ψ. Then, the preference of an argument A is defined by $\sum\limits_{\psi \in S(A)} \nu(\psi)$.*

Definition 5 (attack) *Let $AR_{\mathbf{K}_\mathrm{a}}$ and $AR_{\mathbf{K}_\mathrm{b}}$ be sets of all possible arguments of agents a and b, respectively.*

1. *If $Sentence(A_a) \equiv \neg Sentence(A_b)$ and $\nu(A_a) \geq \nu(A_b)$, then (A_a, A_b) is said to be* a rebut *from a to b.*
2. *If $\neg Sentence(A_a) \in Grounds(A_b)$ and $\nu(A_a) \geq \nu((\emptyset, \neg Sentence(A_a)))$, then (A_a, A_b) is said to be* an undercut *from a to b.*
3. An attack *from a to b is either a rebut or an undercut from a to b.*

When (A_a, A_b) is an attack from a to b, it is said that A_a *attacks* A_b.

Based on Dung [8], in an argumentation framework between two agents, a proposer P makes the first argument and a defeater C makes counterarguments. Hereafter, $\mathbf{K}_\mathrm{P}$ and $\mathbf{K}_\mathrm{C}$ denote their knowledge bases, respectively.

Definition 6 (argumentation framework) *Let $AR_{\mathbf{K}_\mathrm{P}}$ and $AR_{\mathbf{K}_\mathrm{C}}$ be sets of all possible arguments of P and C, respectively, with preferences ν. Let $AT_{\mathbf{K}_\mathrm{P} \rightharpoonup \mathbf{K}_\mathrm{C}}$ and $AT_{\mathbf{K}_\mathrm{C} \rightharpoonup \mathbf{K}_\mathrm{P}}$ be sets of attacks from P to C and from C to P, respectively.* An argumentation framework *between P and C, $AF(\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{C}, \nu)$ is defined as a quadruple $\langle AR_{\mathbf{K}_\mathrm{P}}, AR_{\mathbf{K}_\mathrm{C}}, AT_{\mathbf{K}_\mathrm{P} \rightharpoonup \mathbf{K}_\mathrm{C}}, AT_{\mathbf{K}_\mathrm{C} \rightharpoonup \mathbf{K}_\mathrm{P}} \rangle$.*

2.2 Argumentation Tree

Definition 7 (move) A move *is a pair of a player (an agent) P/C and an argument A in which $A \in AR_{\mathbf{K}_\mathrm{P}}/AR_{\mathbf{K}_\mathrm{C}}$. If player is P/C, then it is said to be P/C's move. For a move $M = (player, argument)$, we denote player and argument by $Ply(M)$ and $Arg(M)$, respectively.*

Definition 8 (move's attack) *M is said to be* an attack *to M', if $(Arg(M), Arg(M'))$ is an attack from $Ply(M)$ to $Ply(M')$.*

Definition 9 (argumentation line, argument set) *Let P and C denote a proposer of a formula φ and its defeater, respectively. Let $AF(\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{C}, \nu)$ be an argumentation framework between P and C.* An argumentation line *D for φ on $AF(\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{C}, \nu)$ is a finite nonempty sequence of moves $[M_1, \ldots, M_n]$ $(i = 1, \ldots, n)$ that satisfies the following:*

1. *$Ply(M_1) = P$, where $Sentence(Arg(M_1)) = \varphi$.*
2. *If i is odd, then $Ply(M_i) = P$, and if i is even, then $Ply(M_i) = C$.*
3. *M_{i+1} is an attack to M_i for each i $(1 \leq i \leq n-1)$.*
4. *No attack occurs against $Arg(M_n)$.*
5. *$M_i \neq M_j$ for each pair of i, j $(1 \leq i \neq j \leq n)$.*
6. *Both $S(Arg(M_1)) \cup S(Arg(M_3)) \cup S(Arg(M_5)) \cup \ldots \cup S(Arg(M_o))$ and $S(Arg(M_2)) \cup S(Arg(M_4)) \cup S(Arg(M_6)) \cup \ldots \cup S(Arg(M_e))$ are consistent, where o and e are the largest odd number and the largest even number less than or equal to n, respectively.*

The above $S(Arg(M_1)) \cup S(Arg(M_3)) \cup S(Arg(M_5)) \cup \ldots \cup S(Arg(M_o))$ and $S(Arg(M_2)) \cup S(Arg(M_4)) \cup S(Arg(M_6)) \cup \ldots \cup S(Arg(M_e))$ are said to be P's argument set on D *and* C's argument set on D, *and they are denoted by $S_P(D)$ and $S_C(D)$, respectively.*

This definition puts the constraints of loop-freeness and consistency of each agent's arguments on an argumentation line.

Definition 10 (win of an argumentation line) *If the last element of an argumentation line D is P's move, then it is said that P* wins *D; otherwise, P* loses *D.*

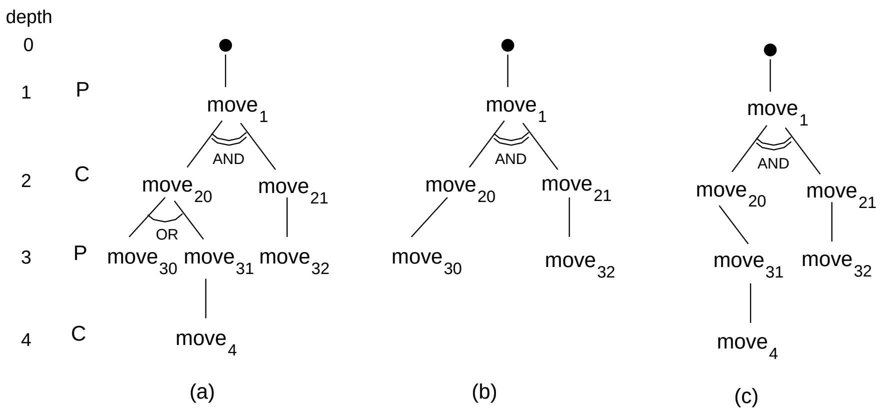


Fig. 2. An argumentation tree and its candidate subtrees

Definition 11 (argumentation tree) An argumentation tree for φ on *$AF(\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{C}, \nu)$ is a tree in which the root node at depth 0 is empty and all the branches*[1] *starting from the node of depth 1 are different argumentation lines for φ on $AF(\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{C}, \nu)$.*

Definition 12 (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 move in the original tree, and selects all child nodes for each node corresponding to P's move.*

Definition 13 (solution subtree) A solution subtree *is a candidate subtree in which P* wins *all of the argumentation lines in the tree.*

Each candidate subtree corresponds to P's selection of an argument, and the solution subtree indicates the case in which P takes a winning strategy. In Figure 2, (a) is an argumentation tree, (b) and (c) are its candidate subtrees, and (b) is the solution subtree.

In general, judgement of an argumentation tree is defined as follows.

Definition 14 (static win of an argumentation tree) *If an argumentation tree has a solution subtree, then P* statically wins *the argumentation tree; otherwise, P* statically loses *it.*

3 Argumentation with Changes in the Knowledge Base

3.1 Execution of Argumentation

We propose an argumentation system that considers the successive executions of all possible argumentation lines, whilst the usual ones consider only a sin-

[1] Here, a branch is a path from the designated node to a leaf node.

gle argumentation line. In a dynamic argumentation, we have to consider the interaction of argumentation lines.

We prepare history $\mathbf{H}_a$ for each agent a to preserve the coherence of each agent's arguments. $\mathbf{H}_a$ is a set of all the formulas contained in a's arguments in the argumentation lines in which a wins. We, however, ignore the coherence of the loser's side. This is based on the idea that the winner should be responsible for his/her arguments, but the loser can make an attack from a different side.

A dynamic argumentation line is defined by extending a static argumentation line with history.

Definition 15 (dynamic argumentation line) *Let P, C denote a proposer of a formula φ and its defeater. Let $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}}, \nu)$ be an argumentation framework between P and C.* A dynamic argumentation line $D = [M_1, \ldots, M_n]$ *for φ on $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}}, \nu)$ with histories $\mathbf{H}_P$ and $\mathbf{H}_C$ is defined as the extension of the (static) argumentation line by adding the following additional condition.*

7. $\mathbf{H}_{Ply(M_i)} \cup S(Arg(M_i))$ is consistent for each i ($1 \leq i \leq n$).

If no misleading is involved, *a dynamic argumentation line* for φ on $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}}, \nu)$ with a history $\mathbf{H}_P$ and $\mathbf{H}_C$, is said to be just *an argumentation line* on $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}}, \nu)$.

Here, we present a dynamic argumentation model. We consider the execution of an argumentation as selecting a branch, updating the commitment store and agents' histories and modifying a tree. The commitment store is a set of all the formulas contained in all arguments given so far.

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. New nodes are added to the argumentation tree if new arguments are generated. Next, another branch is selected. On 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. On the selection of a branch, the turn of an utterance should be kept. This means that if one branch suspends 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 16 (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 and $M_i, \ldots, M_n$ are unmarked, and $Ply(M_i) = t$, then the node M_i is said to be* executable.

Definition 17 (execution of a branch) *For a branch $D = [M_1, \ldots, M_n]$, histories $\mathbf{H}_P, \mathbf{H}_C$ and the commitment store $\mathbf{K}$, execution of D from i ($1 \leq i \leq n$) is defined as follows.*

1. Mark $M_i, \ldots, M_n$.

2. *Set* $\mathbf{K}=\mathbf{K}\cup\bigcup_{k=i}^{n}S(Arg(M_i))$.
3. **if** $Ply(M_n)=P$,
  then *set the current turn to* C *and* $\mathbf{H}_P=\mathbf{H}_P\cup S_P(D)$.
 if $Ply(M_n)=C$,
  then *set the current turn to* P *and* $\mathbf{H}_C=\mathbf{H}_C\cup S_C(D)$.

Definition 18 (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*.*

This Argumentation Procedure with Knowledge Change is formalised as follows.

Argumentation Procedure with Knowledge Change (*APKC2*)

Let $AF(\mathbf{K}_\mathrm{P},\mathbf{K}_\mathrm{C},\nu)$ be an argumentation framework, and φ be a proposed formula.

```
[STEP 1(initialisation)]
    Set $\mathbf{K}=\emptyset$, $\mathbf{H}_P=\emptyset$, $\mathbf{H}_C=\emptyset$, $turn=P$. Construct an initial argumentation
tree for $AF(\mathbf{K}_\mathrm{P},\mathbf{K}_\mathrm{C},\nu)$ on $\varphi$ with $\mathbf{H}_P,\mathbf{H}_C$ with all the nodes unmarked.
[STEP 2(execution of an argumentation)]
    if no branch has an executable node,
        if $turn$=P, then terminate with P's lose.
        else $turn$=C, then terminate with P's win.
    else select a branch and execute it from the executable node.
[STEP 3(modification of a tree)]
    Reconstruct an argumentation tree for $AF(\mathbf{K}_\mathrm{P}\cup\mathbf{K},\mathbf{K}_\mathrm{C}\cup\mathbf{K},\nu)$ on $\varphi$ with
    $\mathbf{H}_P,\mathbf{H}_C$.
    if the nodes $N$ and $M$ are identical, and $N$ is marked whilst $M$ is unmarked,
        then mark $M$.
    go to STEP 2.
```

The elements of $\mathbf{K}$ are included either by $\mathbf{K}_\mathrm{P}$ or $\mathbf{K}_\mathrm{C}$, which are both finite sets. It follows that finite kinds of moves can be generated. Therefore, *APKC2* terminates.

In the modification of a tree in *APKC2*, a new node may be added. An idea of *threat* is introduced to explain this situation.

Definition 19 (threat) *Let* M *and* M' *be moves in an argumentation tree* T *on* $AF(\mathbf{K}_\mathrm{P},\mathbf{K}_\mathrm{C},\nu)$*. If* $S(Arg(M))$ *generates more than one new move that attacks* M'*, then it is said that* M *is* a threat to M'*, and that* T contains a threat*.* M *and* M' *are said to be* a threat resource *and* a threat destination*, respectively.*

Intuitively, a threat is a move that may provide information advantageous to the opponent. A move may be a threat to a move in the same branch.

Definition 20 (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.*

Note that nondeterminism is involved in the selection of a branch in *APKC2*, and finally obtained trees are different depending on the selection.

Consider an argumentation tree in Figure 3. In this figure, M_2 and M_4 are a threat resource and a threat destination, respectively, and M_5 is a newly generated node by this threat. If we execute from the left branch, then *APKC2* proceeds by executing M_1, M_2, M_3, M_4, M_5, and terminates with P's win. On the other hand, if we execute from the right branch, then *APKC2* proceeds by executing M_1, M_4 and suspends. The next turn is P. If there exists no branch in the other candidate trees that starts with P and ends with P, the suspended branch never resumes, and *APKC2* terminates with P's lose.

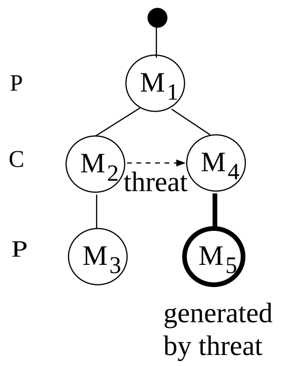


Fig. 3. Argumentation affected on the execution order of branches

We define a dynamic win/lose of an argumentation tree according to *APKC2*.

Definition 21 (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 subtree such that P wins, then CT is said to be* a dynamic solution subtree*.*

Definition 22 (dynamic win of an argumentation tree) *If an argumentation tree has a dynamic solution subtree, then P* dynamically wins *the argumentation tree; otherwise, P* dynamically loses *it.*

3.2 Judgement of Dynamic Win/Lose

APKC2 gives to an execution model for an argumentation procedure. If we only want to judge the result of an argumentation and not simulate the procedure, then there exists a simpler algorithm.

Definition 23 (consistent candidate subtree) *Let CT be a candidate subtree. If there does not exist moves M, M' and a formula ψ that satisfy $Ply(M) = Ply(M') = P$, $\psi \in S(Arg(M))$ and $\neg\psi \in S(Arg(M'))$, then CT is said to be* a consistent candidate subtree.

Let CT be a candidate subtree of an argumentation tree of $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}}, \nu)$. Then we can judge a proposer P's win/lose of CT by the following algorithm. Hereafter, $D \in T$ denotes that a branch D in a tree T.

Judgement of Win/Lose of a Candidate Subtree (*JC*)

```
[STEP 1]
  **if** $CT$ is not consistent, **then** P loses CT.
  **else** set $\mathcal{K} = \bigcup_{D \in CT} S_P(D) \cup \bigcup_{D \in CT} S_C(D)$.
[STEP 2]
  Reconstruct $CT$ on $AF(\mathbf{K}_{\mathrm{P}} \cup \mathcal{K}, \mathbf{K}_{\mathrm{C}} \cup \mathcal{K}, \nu)$, and let the resultant tree be
  $CT'$.
[STEP 3]
  **if** $CT' = CT$,
    **if** all the leaves are P's moves, then P wins CT.
    **else** P loses CT.
  **else** select a new continuous candidate subtree and go to STEP 1.
```

The algorithm *JC* terminates by the same reason as that for termination of *APKC2*.

We show the relationship of dynamic win of an argumentation tree and *JC*.

Lemma 1 *Let T_f be a finally obtained tree when* APKC2 *terminates with P's win. For a subtree T whose root node M is C's move in T_f, let $M_{P_1}, \ldots, M_{P_n}$ be M's child nodes, and let $T_1, \ldots, T_n$ be subtrees whose root nodes are $M_{P_1}, \ldots, M_{P_n}$, respectively. If $T_1, \ldots, T_n$ are all candidate subtrees, then there exists a (static) solution subtree T_i $(1 \leq i \leq n)$.*

Proof)

Assume that T_i is not a solution subtree for some i. Then, T_i includes C's move as a leaf node. Let D be a branch of T_f that contains this node. There should exist another branch as the successive execution of D, since *APKC2* terminates with P's move. On the other hand, when the leaf node of D has been executed, the unmarked nodes nearest to the root node of T_f in every branch of T_i that includes unmarked nodes are C's moves. They are unexecutable. Therefore, a branch in subtrees other than T_i should be selected as D's successive execution. If none of $T_1, \ldots, T_n$ is a solution subtree, it is impossible to terminate *APKC2* with P's move. Hence, one of them should be a solution subtree. □

We can take such T_i as T's candidate subtree, and obtain the following lemma.

Lemma 2 *Let T_f be a finally obtained tree when* APKC2 *terminates with P's win. T_f includes a (static) solution subtree.*

Proof)

For a subtree whose root node M is C's move in T_f, let $M_{P_1}, \ldots, M_{P_n}$ be M's child nodes, and let $T_1, \ldots, T_n$ be subtrees whose root nodes are $M_{P_1}, \ldots, M_{P_n}$, respectively. For each i $(1 \leq i \leq n)$, if T_i is not a candidate subtree, then replace it by its candidate subtree T'_i from lemma 1; otherwise, set T'_i be T_i. There exists a solution subtree T'_i $(1 \leq i \leq n)$, since all of them are candidate subtrees. Repeating this procedure, it is proved that T_f includes a solution subtree. □

Theorem 1 *Let T be an argumentation tree which includes no threat over different candidate subtrees. P dynamically wins T if and only if there exists a candidate subtree in T for which* JC *terminates with P's win.*

Proof) ($\Rightarrow$) Let CT' is a finally obtained tree for a candidate subtree CT in *JC*. From lemma 2, the finally obtained tree T_f in *APKC2* includes a (static) solution subtree. There exists CT that contains this solution subtree, since both threat resource and threat destination are in the same candidate tree from the condition. Moreover, $\bigcup_{D \in T_f} S_P(D)$ is consistent because of the constraints on $\mathbf{H}_P$. Therefore, there exists a candidate tree for which *JC* terminates with P's win.
($\Leftarrow$) Let $CT_0, \ldots, CT_m$ be a sequence of candidate subtrees and $\mathcal{K}_j$ be a set of newly generated formulas by CT_j $(1 \leq j \leq m)$ in *JC*. Suppose that we execute *APKC2* as follows: first executing an arbitrary branch in CT_0, update the commitment store and modify the tree; second, executing an arbitrary branch in the reconstructed tree; and repeat this procedure. In this procedure, newly generated formulas by the execution of a branch is contained either in $\mathcal{K}_1, \ldots, \mathcal{K}_m$. It follows that it should be contained as a node in $\mathcal{K}_m$. If the leaf node of a branch is P's move, then there exists an executable node in another branch. If it is C's move, then P's node should be added on reconstruction. Therefore, there should exist an execution that ends with P. □

4 An Example

Consider the example shown in Section 1. We illustrate various properties of *APKC2* and *JC* using this example.

4.1 Formalisation

The knowledge bases of a proposer P and a defeater C are shown below. The number attached to each formula shows its preference. We assume that the facts and rules are all represented in the knowledge base and the agents have no other knowledge.

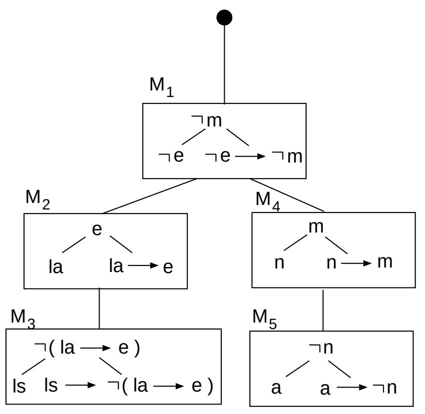
$$\mathbf{K}_{\mathrm{P}} = \left\{ \begin{array}{l} \neg m[1],\ \neg e[2],\ (\neg e \to \neg m)[1],\ \neg(la \to e)[1], \\ ls[1],\ (ls \to \neg(ls \to e))[1],\ \neg n[1],\ a[2], \\ (a \to \neg n)[1] \end{array} \right\}$$

$$\mathbf{K}_{\mathrm{C}} = \left\{ \begin{array}{l} e[1],\ la[1],\ (la \to e)[2],\ m[2],\ n[2], \\ (n \to m)[1],\ \neg a[1],\ (ls \to \neg a)[1] \end{array} \right\}$$

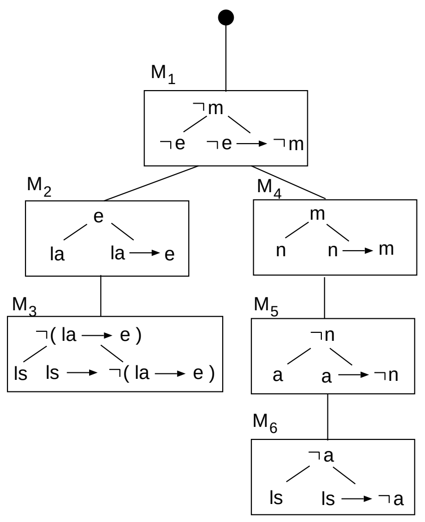
The propositions have the following meanings:
m: P commits murder.
e: there is evidence.
la: P's license was left at the scene of the murder.
ls: P's license was stolen.
n: P was near the scene when the murder was committed.
a: P was at facility A when the murder was committed.

4.2 The Case of Changing from Static Win to Dynamic Lose

Figure 4 shows a relevant part of an initial argumentation tree and a final argumentation tree in *APKC2*. The argumentation starts with the murderer's utterance. The nodes M_1, M_2, M_3, M_4, M_5, and M_6 correspond to the utterances P_1,C_1,P_2,C_2,P_3, and C_3, respectively.



(a) Initial argumentation tree



(b) Final argumentation tree

Fig. 4. The argumentation trees starting from the murderer

This example shows the case in which a proposer statically wins but dynamically loses the argumentation tree.

4.3 Behaviour of Suspend/Resume

Figure 5(a) shows the trees at each step in case the left branch of the tree in Figure 4(a) is selected first. T_0 is the initial argumentation tree, and T_1 is the modified tree based on the knowledge bases obtained after the execution of the left branch. The hatched nodes are marked. T_2 is the tree modified based on the knowledge bases after the execution of the right branch afterward. No unmarked node exists in T_2, which indicates the absence of a counterargument. Then, the procedure terminates. The winner is C, who gives the final argument.

Figure 5(b) shows the trees at each step in case the right branch of the tree in Figure 4(a). T'_1 is the modified tree based on the knowledge bases obtained after the execution of the right branch. The right branch is suspended. T'_2 is the modified tree based on the knowledge bases obtained after the execution of the left branch afterward. In this case, a new node M_6, which corresponds to the utterance C'_3, is added, and it is the only node that is unmarked. To execute this node, the right branch is resumed. T'_3 is the modified tree based on the knowledge bases obtained after this execution. No unmarked node exists in T'_3, which indicates the absence of a counterargument. Then, the procedure terminates. The winner is C, who gives the final argument.

This example shows the procedures with different branch selection orders, and illustrates how suspend/resume occurs.

4.4 The Case of Changing from Static Lose to Dynamic Win

Next, we show an argumentation that starts with the policeman's utterance C_0 in the first example:

C_0: "You committed the murder."

and continues to P_1, C_2, P_3, C_1, P_2, similar to the first example. The argumentation trees are shown in Figure 6. M_0 is a node corresponding to C_0.

The trees can be regarded as C's argumentation trees because the roles of P and C are switched from the first example. C statically loses, since all the leaf nodes in the initial argumentation tree shown in Figure 6(a) are P's move, but dynamically wins, since the final argumentation tree shown in Figure 6(b) is obtained by *APKC2*.

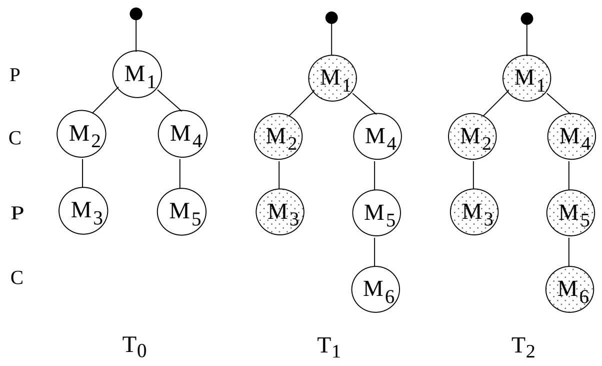
This example shows the case in which a proposer statically loses but dynamically wins the argumentation tree.

4.5 Judgement of dynamic win/lose

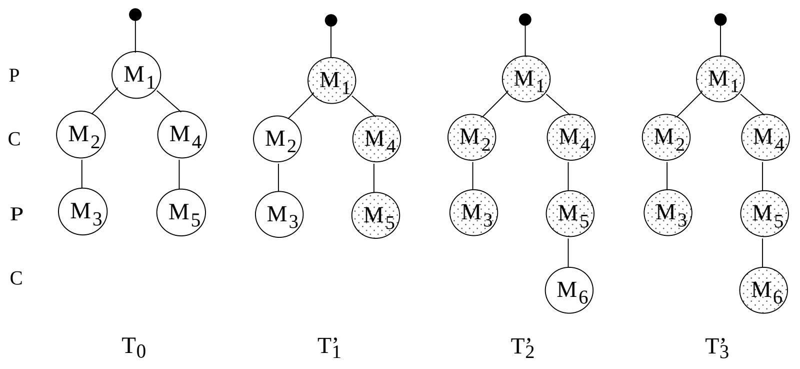
Here, we apply an algorithm *JC* to the first example, starting from the murderer's utterance.

The initial argumentation tree is shown in Figure 4(a). It includes only one candidate subtree[2] and no threats over different candidate subtrees.

[2] This figure shows only the relevant part, and it actually contains more candidate subtrees. Although we ignore them to make a description simple, the result is the same.



(a) APKC2 with left branch selected first



(b) APKC2 with right branch selected first

Fig. 5. Comparison of procedures on the order of selecting branches

Figure 7 shows this procedure.

We take T_0 as a candidate tree, which is consistent.

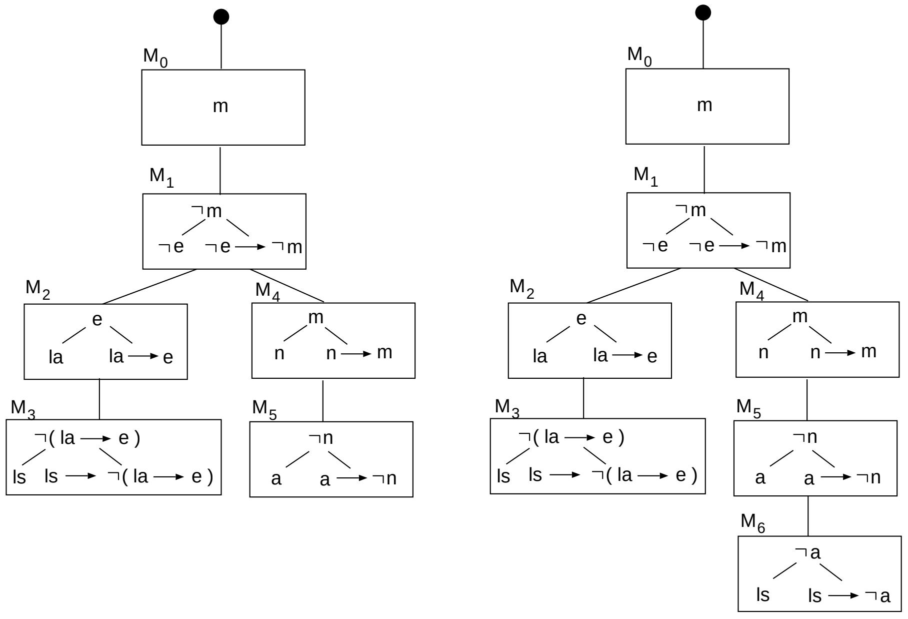
First, we obtain $\mathcal{K}$, a set of all formulas in T_0.

$$\mathcal{K} = \left\{ \begin{array}{l} \neg m,\ \neg e,\ (\neg e \to \neg m),\ \neg(la \to e), \\ ls,\ (ls \to \neg(ls \to e)),\ \neg n,\ a, \\ (a \to \neg n) \\ e,\ la,\ (la \to e),\ m,\ n, \\ (n \to m),\ \neg a,\ (ls \to \neg a) \end{array} \right\}$$

Reconstruct the tree, then a new node M_6 is added to obtain the tree T_1'', which is consistent.

Second, we have the following from T_1:

$$\mathcal{K} = \mathcal{K} \cup \{\ \neg(la \to e),\ ls,\ ls \to \neg(la \to e)\ \ \}$$



(a) Initial argumentation tree (b) Final argumentation tree

Fig. 6. The argumentation trees starting from the policeman

Next, reconstruct the tree to obtain T_2''. Since $T_1'' = T_2''$, and one leaf node is C's move, P loses this candidate subtree.

Since no other candidate subtrees exist, P dynamically loses the argumentation tree from Theorem 1.

5 Discussion

5.1 Improvements on the Earlier Version

Three significant points distinguish the argumentation system proposed in this paper from the earlier version.

First, suspend/resume of a branch is enabled, allowing for the resumption of a settled matter. We mark the executed node instead of deleting it, and make it possible to add a new node to already executed ones. We also provide a simpler judgement algorithm of win/lose for a given candidate subtree. The method of selecting a candidate tree to win can contribute to argumentation strategy.

Second, both agents can continue an argumentation after he/she loses one argumentation line, whilst only the defeater could do so in the earlier version. This makes it possible to handle the case in which a proposer statically loses but dynamically wins.

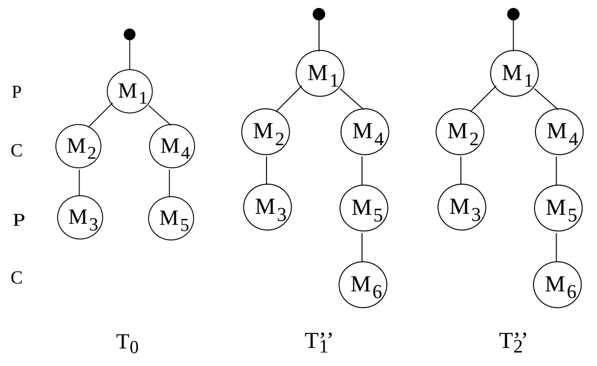


Fig. 7. The argumentation trees for judgement

Third, both P and C can use disclosed knowledge, whereas only C could do so in the earlier version. We prepare the commitment store for this purpose.

In addition, in the earlier version, we had to reconstruct an argumentation tree every time a branch was executed, since some formulas might be deleted from C's knowledge base. However, in the revised version, we do not need to reconstruct a tree, only add nodes to the existing tree, since the usable knowledge is monotonically increasing. This makes the implementation of *APKC2* easier.

Due to these improvements, *APKC2* provides a more natural model for actual argumentations and a simple win/lose judgement algorithm.

5.2 Related Works

García applied argumentation to defeasible logic programming. He considered argumentation to be an explanation and proposed a model in which argumentation is evaluated when a claim is accepted [11]. In his model, evaluation of argumentation is a dialectical proof procedure performed by traversing a constructed dialectical tree. Moguillansky discussed revision of the knowledge base represented in the form of defeasible logic programming [16]. These works both examined reconstruction of the tree with revised knowledge bases, but their goal was to construct undefeated argumentation by selecting suitable defeasible rules, not to consider the effect of the execution of argumentation.

Several works regard argumentation as a dialogue exchanging information between agents [14, 1, 18]. An argument is regarded as a communication protocol between agents. In most models, an agent rejects a proposal if it contradicts his/her knowledge base and accepts it otherwise, and in the end, agreement may be achieved. In these models, an agent's behaviour is determined by the arguments he/she receives, but his/her knowledge base never changes during the argumentation.

Amgoud formalised a negotiation system in an argumentation framework [2]. She considered the knowledge base for each agent separately, as well as its revision by exchanging arguments. The significant difference between her work and ours is that in her approach only a single argumentation line is considered, so only threats to the same branch are taken into account, whereas in our approach all argumentation lines are considered successively, so threats to the other branches are examined. Dunne proposed a "dispute tree" on which successive execution of all argumentation lines are considered [7]. However, the revision of agents' knowledge base, allowing executed moves to add new information to the opponent's knowledge base, is not considered.

Cayrol conducted interesting research on the revision of an argumentation theory [5]. She investigated how acceptable arguments are changed when an argument is added. The aim of her research is a formal analysis on changes to argumentation, and the contents of the additional arguments and reasons for the addition are beyond its scope. In contrast, we focus specifically on the effect of knowledge gained by executing argumentation.

A few models exist in which an independent knowledge base for each agent is considered, but it is more natural to assume the existence of such knowledge bases for treating an actual argumentation. This assumption also contributes the strategy of argumentation. Moreover, we believe that such a strategy is closely related to game programming with incomplete information, such as bridge. In one such game, a player determines his/her next advantageous hand by considering his/her own current hand and the cards displayed so far. It would be interesting to investigate the relationship between strategy in argumentation and strategy in game programming.

6 Conclusion

We have proposed an argumentation system *APKC2*, which is an extension of our earlier argumentation system *APKC*. *APKC* is a system in which multiple argumentation lines are executed in succession, and an agent's knowledge base can change during argumentation. We have extended *APKC* so that the suspend/resume of an argumentation line can be processed, both agents can continue an argumentation after he/she loses one argumentation line and both can use information given in previous arguments. These extensions provide a more natural model of actual argumentation. In addition, we proposed a simpler algorithm for the judgement of the win/lose result of an argumentation tree, which can be applied to argumentation strategy.

In future, we are considering an extension of *APKC2* that can not only directly use new information, but also derive new facts from the new knowledge. We are also considering a strategy to win an argumentation.

References

1. L.Amgoud, S.Parsons, and N.Maudet: Arguments, dialogue, and negotiation, ECAI2000, pp.338-342, 2000.

2. L.Amgoud, Y.Dimopolos and P.Moraitis: A general framework for argumentation-based negotiation, ArgMAS2007, pp.1-17, 2007.
3. L.Amgoud and S.Vesic: Repairing preference-based argumentation frameworks, IJCAI2009, pp.665-670, 2009.
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: Revision of an argumentation system. pp.124-134, KR2008, 2008.
6. C.I.Chesñevar, A.Maguitman and R.Loui: Logical models of argument. ACM Computing Surveys, 32(4), pp.337-383, 2005.
7. P.E.Dunne and T.J.M.Bench-Capon: Two party immediate response disputes: properties and efficiency. Artificial Intelligence, 149(2), pp.221-250, 2003.
8. 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.
9. M.Falappa, G.Kern-Isberner and G.R.Simari: Explanations, belief revision and defeasible reasoning, Artificial Intelligence, 141(1-2), pp.1-28, 2002.
10. A.García, and G.Simari: Defeasible logic programming: an argumentative approach. Theory and practice of logic programming, 4(1), pp.95-138, 2004.
11. A.García, C.Chesñevar, N.Rotstein, and G.Simari: An abstract presentation of dialectical explanations in defeasible argumentation, ArgNMR07, pp.17-32, 2007.
12. S.Joseph and H.Prakken Coherence-driven argumentation to norm consensus, ICAIL2009, pp.58-67, 2009.
13. C.Hamblin: *Fallacies*, Methuen, 1970.
14. S.Kraus, K.Sycara and A.Evenchik: Reaching agreements through argumentation: a logical model and implementation, Artificial Intelligence, 104(1-2), pp.1-69, 1998.
15. S.Modgil: Reasoning about preferences in argumentation frameworks, Artificial Intelligence, 173(9-10), pp.901-1040, 2009.
16. M.O.Moguillansky, et al.: Argument theory change applied to defeasible logic programming, AAAI2008, pp.132-137, 2008.
17. H.Prakken: Combining skeptical epistemic reasoning with credulous practical reasoning. COMMA 2006, pp.311-322, 2006.
18. F.Paglieri and C.Castelfranchi: Revising beliefs through arguments: bridging the gap between argumentation and belief revision in MAS, ArgMAS2004, pp.78-94, 2004.
19. K.Okuno and K.Takahashi: Argumentation with a revision of knowledge base, EUMAS08, CD-ROM, December, 2008.
20. K.Okuno and K.Takahashi: Argumentation system with changes of an agent's knowledge base, IJCAI2009, pp.226-232, 2009.
21. I.Rahwan, and G.Simari (eds.): *Argumentation in Artificial Intelligence*, Springer, 2009.
22. T.Takahashi and H.Sawamura: A logic of multiple-valued argumentation, AAMAS2004, pp.789-805, 2004.

Empirical Argumentation: Integrating Induction and Argumentation in MAS

Santiago Ontañón and Enric Plaza

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain).
santi|enric@iiia.csic.es

Abstract. This paper presents an approach that integrates notions and techniques from two distinct fields of study —namely inductive learning and argumentation in multiagent systems (MAS). We will first discuss inductive learning and the role argumentation may play in multiagent inductive learning, and then how inductive learning can be used to realize an argumentation in MAS based on empirical grounds. We present a MAS framework for empirical argumentation and then we show how this is applied to a particular task where two agents argue in order to reach agreement on a particular topic. Finally, an experimental evaluation of the approach is presented evaluating the quality of the agreements achieved by argumentation.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence — Multiagent systems, Intelligent Agents. I.2.6 [**Artificial Intelligence**]: Learning.

General Terms

Algorithms, Experimentation, Theory.

Keywords

Argumentation, Learning.

1 Introduction

This paper presents an approach that integrates notions and techniques from two distinct fields of study —namely inductive learning and argumentation in multiagent systems (MAS). We will discuss first inductive learning and the role argumentation may play in multiagent inductive learning, and later how inductive learning can be used to realize an argumentation in MAS based on empirical grounds.

Multiagent inductive learning (MAIL) is the study of multiagent systems where individual agents have the ability to perform inductive learning, i.e. an agent is able to learn general descriptions from particular examples. Therefore, induction is a form of empirical-based inference, where what is true (or what is believed by the agent) is derived from the experience of that agent in a particular domain (such experience is usually represented with "cases" or "examples"). Notice that inductive inference is not deductive, and specifically it is not truth-preserving[1], and therefore it captures a form of empirical knowledge that can be called into question by new empirical data and thus needs to be revised.

The challenge of multiagent inductive learning is that several agents will inductively infer empirical knowledge that in principle is not the same, since that knowledge is dependent on each individual in two ways: the concrete empirical data an agent has encountered and the specific inductive method an agent employs.

Therefore, empirical knowledge will be different in principle for each individual agent, and based on a part of the empirical data (the one observed by each individual agent). Communication among agents is necessary in order to reach a shared and agreed-upon empirical-based generalization that is based on, and consistent with, all the empirical data available to a collection of agents. Instead of simply using a communication process that redistributes all data to all agents, we propose an argumentation-based communication process where agents can propose, compare and challenge the empirical knowledge of other agents, with the goal of achieving a more accurate, shared, and agreed-upon body of empirical knowledge.

From the point of view of argumentation in MAS, inductive learning provides a basis for automating, in empirical domains, a collection of activities necessary for implementing artificial agents that support argumentation: how to generate arguments, how to attack and defend arguments, and how to change an agent's beliefs as a result of the arguments exchanged. Logic-based approaches to argumentation like DeLP [2] amend classical deductive logic to support defeasible reasoning. Our approach takes a different path, assuming agents that *learn their knowledge* (by using induction over empirical data) instead of assuming agents have been *programmed* (by giving them a rule-based knowledge base). Therefore, we need to specify empirical methods that are able to perform the required activities of argumentation (generating arguments and attacks, comparing arguments and revising an agent's beliefs): this is the approach we will call *empirical argumentation* for MAS.

This paper presents a MAS framework for empirical argumentation that proposes a way to implement those activities on the basis of the inductive inference techniques developed in the field of Machine Learning. The structure of the paper starts by introducing the notions of inductive learning needed (Section 2). Then Section 3 presents the MAS framework for empirical argumentation called A-MAIL, while Section 4 shows the utility of the framework in the task of concept

[1] Inductive inference is not truth-preserving since new experiences may contradict past generalizations, albeit it is falsity-preserving.

convergence (in which two agents argue with the goal of achieving an agreement on a particular topic); an experimental evaluation of the approach is presented evaluating the quality of the agreements achieved by argumentation. The paper closes with sections on related work and conclusions.

2 Concept Induction

Inductive learning, and in particular concept learning, is the process by which given an *extensional definition* of a concept C (a collection of examples of C and a collection of examples that are not C) then an *intensional* definition (or generalization) of a concept C can be found. Formally, an *induction domain* is characterized as pair $\langle \mathcal{E}, \mathcal{G} \rangle$ where $\mathcal{E}$ is the language describing examples or instances and $\mathcal{G}$ is the language for describing generalizations; usually $\mathcal{E} \subset \mathcal{G}$ is assumed, but this is not necessary. A language is understood as the set of well formed formulas built from a domain vocabulary or ontology $\mathcal{O}$. The relation between languages $\mathcal{E}$ and $\mathcal{G}$ is established by the subsumption relation ($\sqsubseteq$); we say a generalization $g \in \mathcal{G}$ subsumes (or covers) an example $e \in \mathcal{E}$, $g \sqsubset e$, whenever e satisfies the properties described by g [7]. Different approaches to induction work with different languages, from propositional languages (attribute value vectors) to subsets of predicate logic (like Inductive Logic Programming that uses a sublanguage of Horn logic).

Given a collection of examples $E = \{e_1, ..., e_M\}$ described in a language $\mathcal{E}$, an extensional definition of a concept C is a function $C : E \longrightarrow \{+, -\}$, i.e. determines the subset E^+ of (positive) examples of C, and the subset E^- of counterexamples (or negative examples) for C. Induction is a function $I : \mathcal{P}(E) \times C \longrightarrow \mathcal{G}$, which given a collection of examples and a target concept yields an intensional definition $\mathsf{C} \in \mathcal{G}$; generally one single formula in $\mathcal{G}$ is not sufficient to describe an intensional definition so it is usually described as a disjunction of generalizations $\mathsf{C} = h_1 \vee ... \vee h_n$.

Definition 1. *An* intensional definition C *of a concept* C *is a disjunct* $\mathsf{C} = h_1 \vee ... \vee h_n$ *(where* $h_i \in \mathcal{G}$*) such that its generalizations subsume (*$\sqsubseteq$*) all positive examples of* C *and no counterexample of* C*:*

$$\forall e_j \in E^+ : \exists h_i : h_i \sqsubseteq e_j \ \wedge \ \forall e_j \in E^- \wedge \forall h_i : h_i \not\sqsubseteq e_j$$

For simplicity, we will shorten the previous expression as follows: $\mathsf{C} \sqsubseteq E^+ \ \wedge \ \mathsf{C} \not\sqsubseteq E^-$.

As an exemplification of these notions, consider the case where C is the concept *Chair*; in this scenario, the set E may consists of chairs, benches, stools, tables and other furniture, and each specific positive example is a description of one concrete chair and each specific negative example is a description of one furniture item that is not a chair. Finally, and intensional description Chair of concept *Chair* might be "Chair is an object with a seat, four legs and a back."

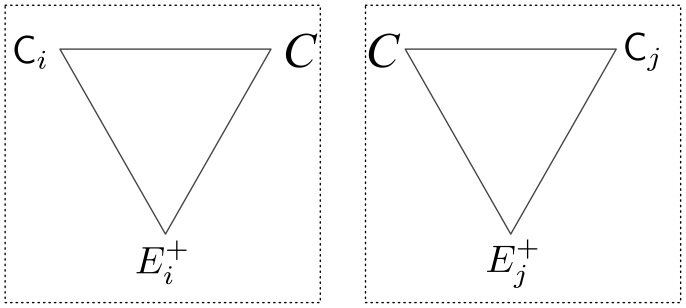


Fig. 1. Schema for two agents where a concept name (C) is shared while intensional descriptions are, in general, not equivalent ($\mathsf{C}_i \ncong \mathsf{C}_i$).

2.1 Inductive agents with empirical beliefs

Since we will focus on argumentation between two agents (say A_1 and A_2), and each agent will have certain beliefs according to what they have learnt, we will now explore how differences between these two agents relate to induction and argumentation. First, we will assume each agent has its own set of examples from which they may learn by induction (say E_1 and E_2) and they are both in principle unrelated although expressed in the same language $\mathcal{E}$. Furthermore, each agent may use, in principle, different induction techniques but they obtain generalizations in the same language $\mathcal{G}$. Thus, for any particular concept C two agents will have intensional descriptions C_1 and C_2 that are, in general, not equal or equivalent. Fig. 1 depicts these relationships between two agents beliefs about what C is based on their empirical data E_1 and E_2.

Finally, since Def. 1 is too restrictive for practical purposes, Machine Learning approaches allow the intensional definitions to subsume less than 100% of positive examples by defining a confidence measure. The goal of induction is, given as a target the function $C : E \longrightarrow \{+,-\}$, to find a new function C, which is a good approximation of C, in the sense of yielding a small error in determining when an example is a positive or negative example of C.

We will use the following confidence measure:

Definition 2. *The* individual confidence *of a hypothesis h for an agent A_i:*

$$B_i(h) = \frac{|\{e \in E_i^+ | h \sqsubseteq e\}| + 1}{|\{e \in E_i | h \sqsubseteq e\}| + 2}$$

$B_i(h)$ is the ratio of examples correctly covered by h over the total number examples covered by h; moreover, we add 1 to the numerator and 2 to the denominator following the Laplace probability estimation procedure (which prevents estimations too close to 0 or 1 when very few examples are covered). Other confidence measures could be used, our framework only requires that the confidence measure reflects how much the set of examples known to an agent endorses the hypothesis. Finally, a threshold τ is established and only hypotheses with confidence $B_i(h) > \tau$ are accepted as valid for the inductive process.

Definition 3. *A hypothesis h is τ-*admissible *for an agent A_i if $B_i(h) \geq \tau$, where $0 \leq \tau \leq 1$.*

3 An Empirical Approach to MAS Argumentation

This section will focus on how to integrate argumentation with inductive agents in scenarios where the goal is to achieve an agreement between two agents on the basis of their empirical knowledge. Here the *empirical* adjective refers to the observations of the real world that each agent has had access to and that is embodied in the set of examples E_1 and E_2 represented using a language $\mathcal{E}$.

Argumentation in Multiagent Inductive Learning (A-MAIL) is a framework where argumentation is used as a communication mechanism for agents that want to perform collaborative inductive tasks such as concept convergence. We do not claim, however, that A-MAIL is a new "argumentation framework" in the sense of Dung [5], it is intended as a framework to integrate argumentation processes and inductive processes in MAS. According to Dung, an argumentation framework $AF = \langle A, R \rangle$ is composed by a set of arguments A and an attack relation R among the arguments. A-MAIL is not a general logic framework and, although certainly we will define what we mean as arguments and attack relations, we take an empirical approach to argumentation. Thus, the main difference from Dung's framework is that, since arguments are generated from examples, our approach necessarily defines a specific relation between arguments and examples, which is not part of the usual interpretations of Dung's framework[2].

3.1 The A-MAIL Approach

Let us define both the kinds of arguments considered by A-MAIL and how arguments attack each other.

There are two kinds of arguments in A-MAIL:

Example Argument: $\alpha = \langle e, \overline{C} \rangle$ is a pair where an example $\alpha.e \in \mathcal{E}$ is related to a concept $\alpha.\overline{C} \in \{C, \neg C\}$, either endorsing C if $\alpha.e$ is a positive example or endorsing $\neg C$ if $\alpha.e$ is a counter-example of C.

Hypothesis Argument: $\alpha = \langle h, \overline{C} \rangle$ is a pair where $\alpha.h$ is a τ-admissible hypothesis and $\alpha.\overline{C} \in \{C, \neg C\}$. An argument $\langle h, C \rangle$ states that $\alpha.h$ is a hypothesis of C, while $\langle h, \neg C \rangle$ states that h is a hypothesis of $\neg C$, i.e. that examples covered by $\alpha.h$ do not belong to C.

[2] Some approaches may consider "counter-examples" as a kind of arguments. This is certainly true, but in our approach there is a constitutive relation between examples and arguments (the "empirical" approach) that is different from merely accepting counter-examples as arguments.

Since hypotheses in arguments are generated by induction, they have an associated degree of confidence for an individual agent:

Definition 4. *The* confidence of a hypothesis argument α *for an agent* A_i *is:*

$$B_i(\alpha) = \begin{cases} \frac{|\{e \in E_i^+ | \alpha.h \sqsubseteq e\}|+1}{|\{e \in E_i | \alpha.h \sqsubseteq e\}|+2} & \text{if } \alpha.\overline{C} = C \\ \\ \frac{|\{e \in E_i^- | \alpha.h \sqsubseteq e\}|+1}{|\{e \in E_i | \alpha.h \sqsubseteq e\}|+2} & \text{if } \alpha.\overline{C} = \neg C \end{cases}$$

Consequently, we can use the threshold τ to impose that only arguments with a strong confidence are admissible in the argumentation process; however, we require only a strong confidence on the part of the individual agent uttering such an argument.

Definition 5. *Am argument α generated by an agent A_i is τ-*admissible *iff it is a hypothesis argument and $B_i(\alpha) > \tau$, or if it is an example argument.*

From now on, only τ-admissible arguments will be considered within the A-MAIL framework.

Now we can define the attack relation between arguments; we will use the notation $\overline{C} \in \{C, \neg C\}$ to specify the boolean valuation of a concept as true or false, while $\widehat{C} = \neg\overline{C}$ is the dual boolean valuation.

Definition 6. *The* attack *relation ($\alpha \twoheadrightarrow \beta$) between two τ-admissible arguments α, β holds when:*

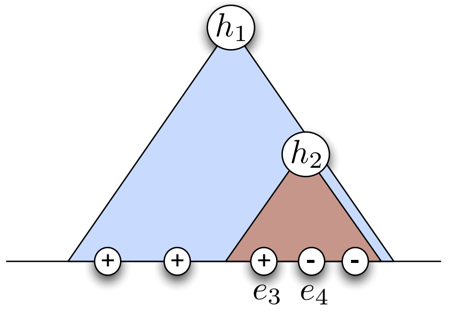
1. $\langle h_1, \widehat{C} \rangle \twoheadrightarrow \langle h_2, \overline{C} \rangle \iff \widehat{C} = \neg\overline{C} \wedge h_2 \sqsubset h_1$, *or*
2. $\langle e, \overline{C} \rangle \twoheadrightarrow \langle h, \widehat{C} \rangle \iff e.\overline{C} = \neg h.\widehat{C} \wedge h \sqsubseteq e$

where $\overline{C} \in \{C, \neg C\}$.

Notice that a hypothesis argument α only attacks another argument β if $\beta.h \sqsubset \alpha.h$, i.e. when α is (strictly) more specific than β. This is required since it implies that all the examples covered by α are also covered by β, and thus if one supports C and the other $\neg C$, they must be in conflict.

Figure 2 shows some examples of arguments and attacks. Positive examples of the concept C are marked with a positive sign, whereas negative examples are marked with a negative sign. Hypothesis arguments are represented as triangles covering examples; when an argument α_1 subsumes another argument α_2, we draw α_2 inside of the triangle representing α_1. Argument α_1 has a hypothesis supporting C, which covers 3 positive examples and 2 negative examples, and thus has confidence 0.57, while argument α_2 has a hypothesis supporting $\neg C$ with confidence 0.60, since it covers 2 negative examples and only one positive example. Now, $\alpha_2 \twoheadrightarrow \alpha_1$ because α_2 supports $\neg C$, α_1 supports C and $\alpha_1.h \sqsubseteq \alpha_2.h$. Moreover, $\varepsilon_3 \twoheadrightarrow \alpha_2$, since e_3 is a positive example of C while α_2 supports $\neg C$ and covers this example ($\alpha_2.h \sqsubseteq \varepsilon_3.e$).

Notice that the viewpoint on the (empirical) admissibility of an argument or of an attack depends on each individual agent, as shown in Fig 3 where



$$\left.\begin{array}{l}\alpha_1 = \langle h_1, C\rangle \\ \alpha_2 = \langle h_2, \neg C\rangle \\ \varepsilon_3 = \langle e_3, C\rangle \\ \varepsilon_4 = \langle e_4, \neg C\rangle\end{array}\right\} \begin{array}{l}\alpha_2 \twoheadrightarrow \alpha_1 \\ \varepsilon_3 \twoheadrightarrow \alpha_2 \\ \varepsilon_4 \twoheadrightarrow \alpha_1\end{array}$$

$$B_i(\alpha_1) = \frac{3+1}{5+2} = 0.57 \qquad B_i(\alpha_2) = \frac{2+1}{3+2} = 0.60$$

Fig. 2. An illustration of the different argument types, their confidences and attacks.

two agents A_i and A_j compare arguments α_1 and α_2 for hypotheses h_1 and h_2, assuming $\tau = 0.75$. From the point of view of agent A_i (the Opponent), proposing argument α_2 as an attack against argument α_1 of agent A_j (the Proponent) is a sound decision, since for A_i, α_1 is not τ-admissible, while α_2 is. However, from the point of view of the Proponent of α_1, α_2 is not τ-admissible. Thus, A_j does not accept α_2 and will proceed by attacking it; on the other side, had it found that confidence $B_j(\alpha_2)$ was higher than $B_j(\alpha_1)$ agent A_j would have revised its beliefs by dismissing α_1 and accepting α_2, as we will explain later.

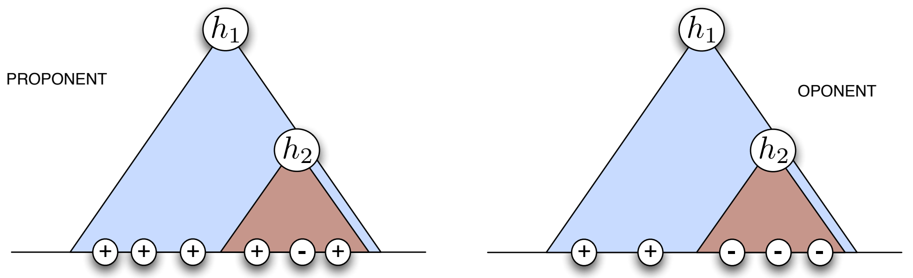
Next we will define when arguments *defeat* other arguments, based on the idea of argumentation lines [2].

Definition 7. *An* Argumentation Line $\alpha_n \twoheadrightarrow \alpha_{n-1} \twoheadrightarrow \ldots \twoheadrightarrow \alpha_1$ *is a sequence of τ-admissible arguments where α_i attacks α_{i-1} and α_1 is called the* root*.*

Notice that odd-numbered arguments are generated by the agent whose hypothesis is under attack (the Proponent of the root argument α_1) and the even-numbered arguments are generated by the Opponent agent attacking α_1. Moreover, since hypothesis arguments can only attack other hypothesis arguments, and example arguments can only attack hypothesis arguments, example arguments can only appear as the left-most argument (e.g. α_n) in an argumentation line.

Definition 8. *An α-*rooted argumentation tree *T is a tree where each path from the root node α to one of the leaves constitutes an argumentation line rooted on α. The example-free argumentation tree T^f corresponding to T is a tree rooted in α that contains the same hypothesis arguments of T and no example argument.*

Therefore, a set of argumentation lines rooted in the same argument α_1 can be represented as an argumentation tree, and vice versa. Notice that example arguments can only appear as leafs in any argumentation tree.



$$B_j(\alpha_1) = \frac{5+1}{6+2} = 0.75 \qquad B_i(\alpha_1) = \frac{2+1}{5+2} = 0.43$$

$$B_j(\alpha_2) = \frac{1+1}{3+2} = 0.40 \qquad B_i(\alpha_2) = \frac{3+1}{3+2} = 0.8$$

Fig. 3. An comparison of two individual viewpoints on arguments, attacks, and acceptability.

$$\left.\begin{array}{r} e_4 \twoheadrightarrow \alpha_3 \twoheadrightarrow \beta_2 \twoheadrightarrow \alpha_1 \\ e_5 \twoheadrightarrow \alpha_3 \twoheadrightarrow \beta_2 \twoheadrightarrow \alpha_1 \\ \alpha_7 \twoheadrightarrow \beta_6 \twoheadrightarrow \alpha_1 \end{array}\right\}$$

$$\begin{tikzcd}
 & & & \alpha_1 & \\
 & & \beta_2 \arrow[ur, two heads] & & \beta_6 \arrow[ul, two heads] \\
 & \alpha_3 \arrow[ur, two heads] & & & \alpha_7 \arrow[u, two heads] \\
e_4 \arrow[ur, two heads] & & e_5 \arrow[ul, two heads] & &
\end{tikzcd}$$

Fig. 4. Multiple argumentation lines rooted in the same argument α_1 can be composed into an argumentation tree.

Figure 4 illustrates this idea, where three different argumentation lines rooted in the same α_1 are shown with their corresponding argumentation tree. The α_i arguments are provided by the Proponent agent (the one proposing the root argument) while β_i arguments are provided by the Opponent trying to attack the Proponent's arguments.

Definition 9. *Let an T α-rooted argumentation tree, where argument α belongs to an agent A_i, and let T^f be the example-free argumentation tree corresponding to T. Then the root argument α is* warranted *(or* undefeated*) iff all the leaves of T^f are arguments belonging to A_i.*

Now we are able to define the state of the argumentation among two agents A_i and A_j at an instant t as the tuple $\langle R_i^t, R_j^t, G^t \rangle$, consisting of:

– $R_i^t = \{\langle h, C\rangle | h \in \{h_1, ..., h_n\}\}$, the set of arguments defending the current intensional definition $\mathsf{C}_i^t = h_1 \vee ... \vee h_n$ of agent A_i;

- R_j^t is the same for A_j.
- G^t contains the collection of arguments generated before t by either agent, and belonging to one argumentation tree rooted in an argument in $R_i^t \cup R_j^t$.

3.2 Argument Generation Through Induction

Agents need two kinds of argument generation capabilities: generating *empirical arguments* based on the individual examples known to an agent, and generating *attack arguments* that attack arguments provided by other agents; notice that a defense argument is simply $\alpha' \twoheadrightarrow \beta \twoheadrightarrow \alpha$, i.e. an attack on the argument attacking a previous argument. Concerning empirical arguments, they are generated by using induction to find an initial intensional definition C from examples; for this reason, an agent A_i can generate an intensional definition of C by using any inductive learning algorithm capable of learning concepts as a disjunction of hypothesis, e.g. learning algorithms such as CN2[3] or FOIL[12].

Attack arguments, however, require a more sophisticated form of induction. When an agent A_i wants to generate an argument β such that $\beta \twoheadrightarrow \alpha$, A_i has to find an inductive hypothesis h for β that satisfies four conditions:

1. $\beta.h$ should support the opposite concept than α: namely $\beta.C = \neg\alpha.C$,
2. $\beta.h$ should have a high confidence $B_i(\beta)$ (at least being τ-admissible),
3. $\beta.h$ should satisfy $\alpha.h \sqsubset \beta.h$, and
4. β should not be attacked by any undefeated argument in G^t.

Currently existing inductive learning techniques cannot be applied out of the box, because they do not satisfy these conditions (mainly the last two conditions).

In previous work, we developed the Argumentation-based Bottom-up Induction (ABUI) algorithm, capable of performing such task [9]; this is the inductive algorithm used in our experiments by the agents. However, any algorithm which can search the space of hypotheses looking for a hypothesis which satisfies the four conditions stated before would work in our framework: e.g. CN2 could be modified in a way that the search of rules is restricted to the subspace of rules satisfying $\alpha.h \sqsubset \beta.h$.

Let L be the inductive algorithm used by an agent A_i; then to attack an argument $\alpha = \langle h, \overline{C} \rangle$ for C has to generate an argument $\beta = \langle h', \widehat{C} \rangle$ such that $\beta \twoheadrightarrow \alpha$. The agent calls L to generate such hypothesis h', then:

- If L returns an individually τ-admissible h', then β is the attacking argument to be used.
- If L fails to find an argument, then A_i looks for examples attacking α in E_i. If any exist, then one such example is randomly chosen to be used as an attacking argument.

Otherwise, A_i is unable to generate any argument attacking α. If an argument or example is not enough to defeat another argument, additional arguments or examples can be sent in subsequent rounds of the A-MAIL protocol (see below).

3.3 Empirical Belief Revision

During argumentation, agents exchange arguments which contain new hypotheses and examples. These exchanges contain empirical knowledge that agents will integrate with their previous empirical beliefs. Consequently, their beliefs will change in such a way that their hypothesis are consistent with the accrued empirical evidence: we call this process empirical belief revision. The belief revision process of an agent A_i at an instant t, with an argumentation state $\langle R_i^t, R_j^t, G^t \rangle$ starts whenever A_i receives an argument from another agent:

1. If it is an example argument ε then $\varepsilon.e$ is added as a new example into E_i, i.e. A_i expands its extensional definition of C.
2. Whether the received argument is an example or an hypothesis, the agent re-evaluates the confidence of any argument in R_i^t or G^t: if any of these arguments becomes no longer τ-admissible for A_i they removed from R_i^{t+1} and G^{t+1}.
3. If any argument α in R_i^t became defeated, and A_i is not able to expand the argumentation tree rooted in α to defend it, then the hypothesis $\alpha.h$ will be removed from $\mathsf{C_i}$. This means that some positive examples in E_i will not be covered by $\mathsf{C_i}$ any longer. The inductive learning algorithm is called again to generate new hypotheses that cover the now uncovered examples.

Thus, we have presented the A-MAIL approach to empirical argumentation for MAS. Notice that, as shown in Fig. 5 all aspects of the argumentation process (generating arguments and attacks, accepting arguments, determining defeat, and revising beliefs) are supported on an empirical basis and, from the point of view of MAS, implemented by autonomous decision making of artificial agents. The activities in Fig. 5 permit the MAS to be self-sufficient in a domain of empirical enquiry, since individual agents are autonomous and every decision is based on the empirical knowledge available to them.

The next section presents an application of this MAS framework to reach agreements in MAS.

4 Concept Convergence

We have developed A-MAIL as part of our research line on deliberative agreement[3], in which 2 or more artificial agents use argumentation to reach different forms of agreement. In this section we will present a particular task of deliberative agreement called concept convergence. The task of *Concept Convergence* is defined as follows: Given two or more individuals which have individually learned non-equivalent meanings of a concept C from their individual experience, find a shared, equivalent, agreed-upon meaning of C.

Definition 10. Concept Convergence *(between 2 agents) is the task defined as follows:*

[3] This is part the project Agreement Technologies: http://www.agreement-technologies.org/

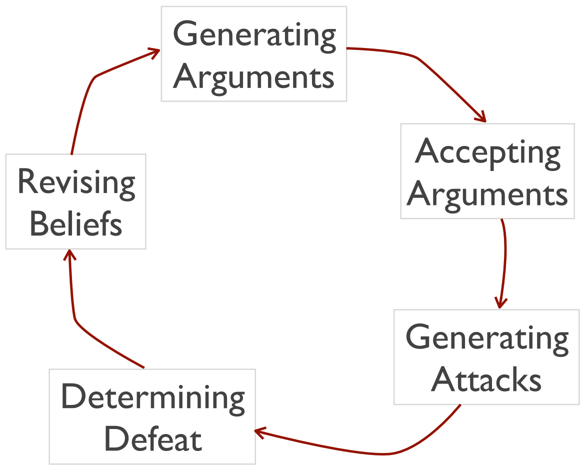


Fig. 5. The closed loop of empirically based activities used in argumentation.

Given *two agents (A_i and A_j) that agree on the sign C denoting a concept ($C_i \cong C_j$) and with individually different intensional ($\mathsf{C}_i \ncong \mathsf{C}_i$) and extensional ($E_i^+ \neq E_j^+$) definitions of that concept,*

Find *a convergent, shared and agreed-upon intensional description ($\mathsf{C}_i' \cong \mathsf{C}_j'$) for C that is consistent for each individual with their extensional descriptions.*

For example, in the experiments reported in this paper, we used the domain of marine sponge identification. The two agents need to agree on the definition of the target concept $C = hadromerida$, among others. While in ontology alignment the focus is on establishing a mapping between the ontologies of the two agents, here we assume that the ontology is shared, i.e. both agents share the concept name *hadromerida*. Each agent has experience in a different area (one in the Atlantic, and the other in the Mediterranean), so they have collected different samples of hadromerida sponges, those samples constitute their extensional definitions (which are different, since each agent has collected sponges on their own). Now they want to agree on an intensional definition C, which describes such sponges. In our experiments, one such intensional definition reached by one of the agents is: $\mathsf{C} =$ "all those sponges which do not have gemmules in their external features, whose megascleres had a tylostyle smooth form and that do not have a uniform length in their spikulate skeleton". In the remainder of this paper we will present how agents can combine argumentation and inductive learning to argue about such definitions.

Concept convergence is assessed individually by an agent A_i by computing the *individual degree of convergence* among two definitions C_i and C_j as:

Definition 11. *The* individual degree of convergence *among two intensional definitions C_i and C_j for an agent A_i is:*

$$K_i(\mathsf{C}_i, \mathsf{C}_j) = \frac{|\{e \in E_i | \mathsf{C}_i \sqsubseteq e \wedge \mathsf{C}_j \sqsubseteq e\}|}{|\{e \in E_i | \mathsf{C}_i \sqsubseteq e \vee \mathsf{C}_j \sqsubseteq e\}|}$$

where K_i is 0 if the two definitions are totally divergent, and 1 when the two definitions are totally convergent. The degree of convergence corresponds to the ratio between the number examples covered by both definitions (intersection) and the number of examples covered by at least one definition (union). The closer the intersection is to the union, the more similar the definitions are.

Definition 12. *The* joint degree of convergence *of two intensional definitions C_i and C_j is:*

$$K(\mathsf{C}_i, \mathsf{C}_j) = min(K_i(\mathsf{C}_i, \mathsf{C}_j), K_j(\mathsf{C}_j, \mathsf{C}_i))$$

Concept convergence is defined as follows:

Definition 13. *Two intensional definitions are* convergent *($\mathsf{C}_i \cong \mathsf{C}_j$) if $K(\mathsf{C}_i, \mathsf{C}_j) \geq 1 - \epsilon$, where $0 \leq \epsilon \leq 1$ is a the degree of divergence allowed.*

The next section describes the protocol to achieve concept convergence.

4.1 Argumentation Protocol

The CC argumentation process follows an iteration protocol composed of a series of rounds, during which two agents will argue about the individual hypotheses that compose their intensional definitions of a concept C. At each round t of the protocol, each agent A_i holds a particular intensional definition C_i^t, and only one agent will hold a *token*. The holder of the token can assert new arguments and then the token will be passed on to the other agent. This cycle will continue until $\mathsf{C}_i \cong \mathsf{C}_j$.

The protocol starts at round $t = 0$ and works as follows:

1. Each agent A_i communicates to the other their current intensional definition by sharing R_i^0. The token is given to one agent at random, and the protocol moves to 2.
2. The agents share their individual convergence degrees ($K_i(\mathsf{C}_i, \mathsf{C}_j)$ and $K_j(\mathsf{C}_j, \mathsf{C}_i)$). If $\mathsf{C}_i \cong \mathsf{C}_j$ the protocol ends with success; if no agent has produced a new attack in the last two rounds then the protocol ends with failure; otherwise it moves to 3.
3. If modified by belief revision, the agent A_i with the token, sends a message communicating its current intensional definition R_i^t. Then, the protocol moves to 4.
4. If any argument $\alpha \in R_i^t$ is defeated, and A_i can generate an argument α' to defend α, the argument will be sent to the other agent. Also, if any of the undefeated arguments $\beta \in R_j^t$ is not individually τ-admissible for A_i, and

A_i can find an argument β' to extend any argumentation line rotted in β, in order to attack it, then β' is sent to the other agent. If at least one of these arguments was sent, a new round $t+1$ starts; the token is given to the other agent, and the protocol moves back to 2. Otherwise, if none of these arguments could be found, the protocol moves to 5.

5. If there is any example $e \in E_i^+$ such that $\mathsf{C}_j^t \not\sqsubseteq e$ (i.e. a positive example not covered by the definition of A_j), A_i will send e to the other agent, stating that the intentional definition of A_j does not cover e. A new round $t+1$ starts, the token is given to the other agent, and the protocol moves to 2.

Moreover, in order to ensure termination, no argument is allowed to be sent twice by the same agent. A-MAIL ensures that the convergence of the resulting concepts is at least τ if (1) the number of examples is finite, (2) the number of hypotheses that can be generated is finite. Convergence higher than τ cannot be ensured, since $100 \times (1-\tau)\%$ of the examples covered by a τ-admissible hypothesis might be negative, causing divergence. Even when both agents use different inductive algorithms, convergence is assured since by assumption they are using the same finite generalization space, and there is no hypothesis τ-admissible by one agent that could not be τ-admissible by the other agent when both use the same acceptability condition over the same collection of examples.

4.2 Experimental Evaluation

In order to empirically evaluate A-MAIL with the purpose of concept convergence we used the marine sponge identification problem. Sponge classification is interesting because the difficulties arise from the morphological plasticity of the species, and from the incomplete knowledge of many of their biological and cytological features. Moreover, benthology specialists are distributed around the world and they have experience in different benthos that spawn species with different characteristics due to the local habitat conditions. The specific problem we target in these experiments is that of agreeing upon a shared description of the features that distinguish one order of sponges from the others.

To have an idea of the complexity of this problem, Figure 6 shows a description of one of the sponges collected from the Mediterranean sea used in our experiments. As Figure 6 shows, a sponge is defined by five groups of attributes: ecological features, external features, anatomy, features of its spikulate skeleton, and features of its tracts skeleton. Specifically, we used a collection of 280 sponges belonging to three different orders of the demospongiae family: axinellida, hadromerida and astrophorida. Such sponges were collected from both the Mediterranean sea and Atlantic ocean. In order to evaluate A-MAIL, we used each of the three orders as target concepts for concept convergence. In an experimental run, we split the 280 sponges randomly among the two agents and, given a target concept, the goal of the agents was to reach a convergent definition of such concept. The experiments model the process that two human experts undertake when they get together to discuss over which features determine whether a sponge belongs to a particular order.

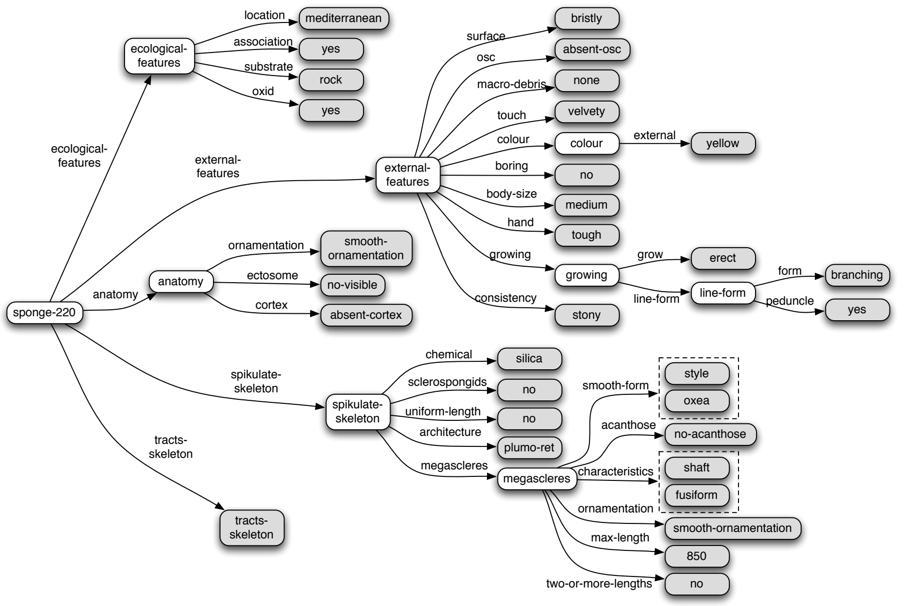


Fig. 6. A description of one of the sponges of the Axinellida order used in our experiments.

	Centralized		*Individual*			*A-MAIL*		
C	P	R	P	R	K	P	R	K
Axinellida	0.98	1.00	0.97	0.95	0.80	0.97	0.95	0.89
Hadromerida	0.85	0.98	0.89	0.91	0.78	0.92	0.96	0.97
Astrophorida	0.98	1.00	0.97	0.97	0.93	0.98	0.99	0.97

Table 1. Precision (P), Recall (R) and degree of convergence (K) for the intensional definitions obtained using A-MAILversus those obtained using .

We compared the results of A-MAIL with respect to agents which do not perform argumentation (*Individual*), and to the result of centralizing all the examples and performing centralized induction (*Centralized*). Thus, the difference between the results of *individual* agents and agents using A-MAIL should provide a measure of the benefits of A-MAIL for concept convergence, where as comparing with *Centralized* gives a measure of the quality of the outcome. All the results are the average of 10 executions, $\epsilon = 0.05$ and $\tau = 0.75$.

Table 1 shows one row for each of the 3 concepts we used in our evaluation: Axinellida, Hadromerida and Astrophorida. For each setting we show three values: precision, measuring how many of the examples covered by the definition are actually positive examples; recall, measuring how many of the total number of positive examples in the data set are covered by the definition; and convergence, as defined in Definition 12. The first thing we see in Table 1 is that A-MAIL is

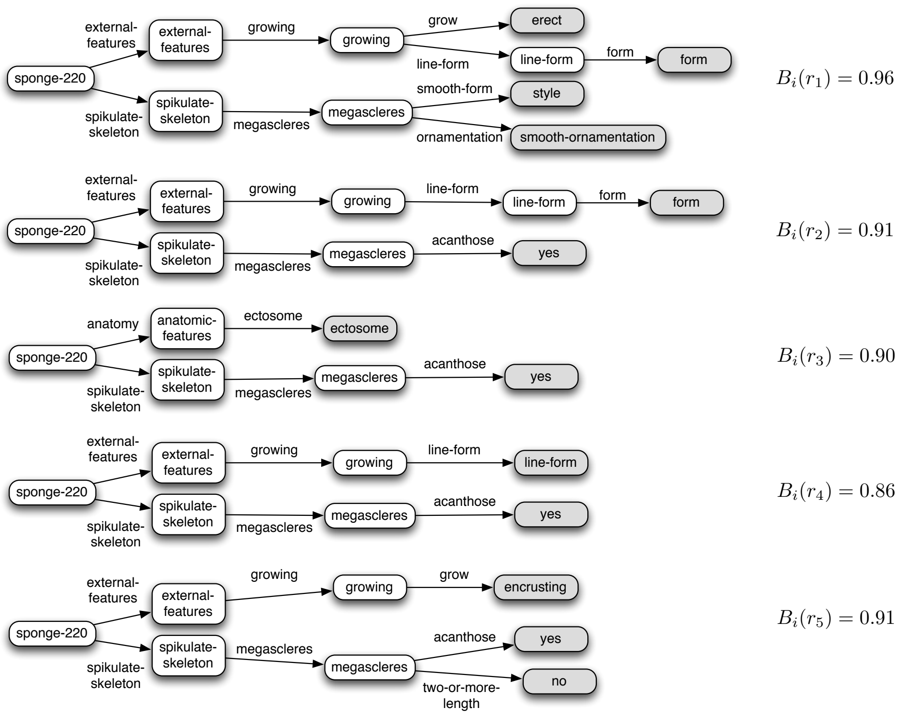


Fig. 7. Set of rules forming the definition of Axinellida and obtained by one of the agents using A-MAIL in our experiments.

able to increase convergence from the initial value appearing in the Individual setting. For all concepts except for Axinellida the convergence was higher than 0.95 (i.e. $1 - \epsilon$). Total convergence was not reached for thhat concepts because in our experiments $\tau = 0.75$, allowing hypotheses to cover some negative examples and preventing overfitting. This means that acceptable hypotheses can cover some negative examples, and thus generate some divergence. Increasing τ could improve convergence but if would make finding hypotheses by induction more difficult, and thus recall might suffer. Moreover, even precision and recall improve thanks to argumentation, reaching values close to the ones achieved by a Centralized setting.

Figure 7 shows the set of rules that one of the agents in our experiments using A-MAIL obtained as the definition of the concept Axinellida. For instance, the first rule states that "all the sponges with an erect and line-form growing, and with megascleres in the spikulate skeleton which had style smooth form and smooth ornamentation belong to the Axinellida order". By looking at those rules, we can clearly see that both the growing external features and the characteristics of the megascleres are the distinctive features of the Axinellida order.

In summary, we can conclude that A-MAIL successfully achieves concept convergence by integrating argumentation and inductive learning, in addition to improve the quality of the intensional definition (precision and recall). This is achieved by exchanging only a small percentage of the examples the agents know (as opposed to the Centralized setting where all the examples are given to a single agent, which might not be feasible in some applications). Additionally, in average, the execution time of A-MAIL is lower than that of a centralized strategy.

5 Related Work

Concerning argumentation in MAS, previous work focuses on several issues like a) logics, protocols and languages that support argumentation, b) argument selection and c) argument interpretation, a recent overview can be found at [13].

The idea that argumentation might be useful for machine learning was discussed in [6], but no concrete proposal has followed, since the authors goal was propose that a defeasible logic approach to argumentation could provide a sound formalization for both expressing and reasoning with uncertain and incomplete information as appears in Machine Learning. Since the possible hypotheses can be induced from data could be considered an argument, and then by defining a proper attack and defeat relation, a sound hypotheses can be found. However, they did not develop the idea, or attempted the actual integration of an argumentation framework with any particular machine learning technique. Amgoud and Serrurier [1] elaborated on the same idea, proposing an argumentation framework for classification. Their focus is on classifying examples based on all the possible classification rules (in the form of arguments) rather than on a single one learned by a machine learning method.

A related idea is that of *argument-based machine learning* [8], where some examples are augmented with a justification or "supporting argument". The idea is that those supporting arguments are then used to constrain the search in the hypotheses space: only those hypotheses which classify examples following the provided justification are considered. Notice that in this approach, arguments are used to augment the information contained in an example. A-MAIL uses arguments in a different way. A-MAIL does not require examples to be augmented with such supporting arguments; in A-MAIL the inductive process itself generates arguments. Notice, however, that both approaches could be merged, and that A-MAIL could also be designed to exploit extra information in the form of examples augmented with justifications. Moreover, A-MAIL is a model for multiagent induction, whereas argument-based machine learning is a framework for centralized induction which exploits additional annotations in the examples in the form of arguments.

The idea of using argumentation with case-based reasoning in multiagent systems has been explored by [11] in the AMAL framework. Compared to A-MAIL, AMAL focuses on lazy learning techniques where the goal is to argue about the classification of particular examples, whereas A-MAIL, although uses cases and vase bases, allows agents to argue about rules generated through inductive learning techniques. Moreover, the AMAL framework explored a related idea to A-MAIL, namely learning from communication [10]. An approach similar to AMAL is PADUA [14], an argumentation framework that allows agents to use examples to argue about the classification of particular problems, but they generate association rules and do not perform concept learning.

6 Conclusions

The two main contributions of this paper are the definition of an argumentation framework for agents with inductive learning capabilities, and the introduction of the concept convergence task. Since our argumentation framework is based on reasoning from examples, we introduced the idea of *argument admissibility*, which measures how much empirical support an argument has, which is used to define an *attack* relation among arguments. A main contribution of the paper has been to show the feasibility of a completely automatic and autonomous approach to argumentation in empirical tasks. All necessary processes are autonomously performed by artificial agents: generating arguments from their experience, generating attacks to defeat or defend, changing their beliefs as a result of the argumentation process — they are all empirically based and autonomously undertook by individual agents.

The A-MAIL framework has been applied in this paper to the concept convergence task. However, it can also be seen as a multi-agent induction technique to share inductive inferences [4]. As part of our future work, we want to extend our framework to deal with more complex inductive tasks, such achieving convergence on a collection of interrelated concepts, as well as scenarios with more than 2 agents. Our long term goal is to study the relation and integration of inductive inference and communication processes among groups of intelligent agents into a coherent unified MAS framework.

Acknowledgments

This research was partially supported by projects Next-CBR TIN2009-13692-C03-01 and Agreement Technologies CONSOLIDER CSD2007-0022.

References

[1] Leila Amgoud and Mathieu Serrurier. Arguing and explaining classifications. In *Argumentation in Multi-Agent Systems*, volume 4946 of *LNCS*, pages 164–177, 2008.

[2] Carlos I. Chesñevar, Guillermo R. Simari, and Lluís Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *LNAI/LNCS Series (Proc. 8th Intl. LPNMR Conf*, pages 158–171. Springer, 2005.

[3] Peter Clark and Tim Niblett. The CN2 induction algorithm. In *Machine Learning*, pages 261–283, 1989.

[4] Winton Davies and Peter Edwards. The communication of inductive inferences. In *ECAI '96: Selected papers from the Workshop on Distributed Artificial Intelligence Meets Machine Learning, Learning in Multi-Agent Environments*, pages 223–241, London, UK, 1997. Springer-Verlag.

[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(2):321–357, 1995.

[6] Sergio Alejandro Gómez and Carlos Iván Chesñevar. Integrating defeasible argumentation and machine learning techniques. *CoRR*, cs.AI/0402057, 2004.

[7] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[8] Martin Mozina, Jure Zabkar, and Ivan Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

[9] Santi Ontañón and Enric Plaza. Multiagent inductive learning: an argumentation-based approach. In *Submitted*, 2010.

[10] Santiago Ontañón and Enric Plaza. Case-based learning from proactive communication. In *IJCAI*, pages 999–1004, 2007.

[11] Santiago Ontañón and Enric Plaza. Learning and joint deliberation through argumentation in multiagent systems. In *Proc. AAMAS'07*, pages 971–978, 2007.

[12] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[13] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer Publishing Company, Incorporated, 2009.

[14] Maya Wardeh, Trevor J. M. Bench-Capon, and Frans Coenen. PADUA: a protocol for argumentation dialogue using association rules. *Artificial Intelligence in Law*, 17(3):183–215, 2009.

Reasoning about Trust using Argumentation: A position paper

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

[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, Graduate Center
City University of New York, 365 Fifth Avenue, New York, NY 10016, USA
[3] Department of Computer Science, University of Liverpool,
Ashton Building, Ashton Street, Liverpool, L69 3BX, United Kingdom
mcburney@liverpool.ac.uk

Abstract. Trust is a mechanism for 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. Managing information about trust involves inference with uncertain information, decision making, and dealing with commitments and the provenance of information, all areas to which systems of argumentation have been applied. Here we discuss the application of argumentation to reasoning about trust, identifying some of the components that an argumentation-based system for reasoning about trust would need to contain and sketching the work that would be required to provide such a system.

1 Introduction

Trust is a mechanism for managing the uncertainty about autonomous entities and the information they store. 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 steadily become an ever more important concept in computer science.

Trust is an especially important issue from the perspective of autonomous agents and multiagent systems. 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.

This paper argues that systems of argumentation have an important role to play in reasoning about trust. We start in Section 2 by briefly reviewing work that defines important aspects of trust and giving an extended example which illustrates some of these aspects. Section 3 then briefly reviews some of the work on reasoning about trust and identifies some of the characteristics of any effective system for dealing with trust information. Building on this discussion, Section 4 then argues that systems of argumentation can handle trust and sketches a specific system of argumentation for doing this. Section 5 then concludes.

2 Trust

As a number of authors have pointed out, trust is a concept that is both complex and rather difficult to pin down precisely and as a result, there are a number of different definitions in the literature. Thus, to pick a few specific examples, Sztompka [26] (cited in [7]) suggests that:

> Trust is a bet about the future contingent actions of others.

while Mcknight and Chervany [20], drawing on a range of existing definitions, define trust as:

> Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

and Gambetta [4] states:

> Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends.

While these definitions differ, there are clearly some common elements. There is a degree of uncertainty associated with trust — whether expressed as a subjective probability, as a bet (which, of course, can be expressed as a subjective probability [11]), or as a “feeling of security”. Trust is tied up with the relationships between individuals. Trust is related to the actions of individuals and how those actions affect others.

It is also pointed out in a number of places that there are different kinds of trust, what Jøsang et al. [12] call “trust scopes”. For example, [12] cites the classification of [9] which identifies the following types of trust:

1. *Provision trust:* the trust that exists between the user of a service or resource, and the provider of that resource.
2. *Access trust:* the trust that exists between the owner of a resource and those that are accessing those resources.
3. *Delegation trust:* the trust that exists between an individual who delegates responsibility for some action or decision and the individual to which that action or decision is delegated.
4. *Identity trust:* trust that an individual is who they claim to be.
5. *Context trust:* trust that an individual has in the existence of sufficient infrastructure to support whatever activities that individual is engaged in.

We illustrate some of these different types of trust with the following example.

> Alice is planning a picnic for a group of friends. She asks around amongst some of her aquaintances for ideas about where to hold the picnic. Bob suggests a park a little way outside of the city where he goes quite regularly (provision trust, relating to information) — he says it is quiet and easy to get to. Carol says she has never been to the park herself, but has heard that the bugs are terrible (provision trust, relating to information).
>
> Alice decides that the picnic will be a potluck[4]. Alice asks David to bring potato salad (delegation trust) and Eric says he will bring bread from the bakery near his house (provision trust, relating to a good). Fran offers to bake a cake (provision trust, relating to a good). Carol says she will make her famous barbeque chicken, cooking it on the public barbeques that Alice believes are provided by the park (context trust).
>
> The picnic is scheduled for midday. George arranges to pick up Alice from her house at 10am in order to drive her to the park (Alice doesn't have a car). Harry, who can borrow a minivan (access trust), offers to collect several people from their homes and stop on the way to buy a case of beer. Iain, who is going to ride with George, says he'll bring a soccer ball so they can all play after lunch. John asks if he can bring a friend of a friend, Keith, whom John has never met, but whom John knows will be visiting the city and is unoccupied that day (identity trust).

As Alice makes the arrangements, she is obviously trusting a lot of people to make sure that the plan comes together in ways that are rather distinct.

Bob and Carol are providing information. To decide whether to go to the park, Alice has to factor in the trustworthiness of that information in deciding this, she has to take into account how reliable Bob and Carol are as information providers, not least because the information that they have gven here is contradictory. She might judge that what she knows about Bob (that he goes to the park often) makes him more trustworthy than Carol in this regard (though in other contexts, such as when deciding what film to see, she might value Carol's opinion more), and the fact that Carol is relying on information from yet another person might strengthen this feeling (or, equally, make Alice value Carol's opinion about the park less).

The trust involved in handling the information from Bob and Carol seems to be some what different to the handling of trust when considering the makeup of the meal. Here Alice has to balance not the reliability of the information that people provide, but the *commitments* they are making, the extent to which Carol, David, Eric, Fran, George, Harry and Iain will do what they say they will do. Carol may be a terribly unreliable source of information about parks, and thus untrustworthy in that regard, but a superb provider of barbequed chicken, and one who has never failed to bring that chicken to a potluck when she says that she will. In contrast, Alice may know that Fran saying she will bake a cake means very little. She is just as likely to bake cookies, or realise late the night before the picnic that she has no flour and will have to bring a green salad instead (thus ruining the meal). David, on the other hand, is quite likely not to make

[4] "Pot luck" means that all the guests are expected to bring something that will contribute to the meal, typically an item of food or a beverage.

potato salad; but if he doesn't, he can be relied upon to subsitute it with some close approximation, a pasta salad for example.

In other words, an individual can be an untrustworthy source of information, but a trustworthy provider of services, or indeed an untrustworthy provider of services but a very reliable information source (it is perfectly possible that Fran only ever provides correct information despite her food-related flakiness) — there are different dimensions of trust for different services that are provided (here, information and food items). We distinguish this by talking of the *context* of trust. Similarly, the failure of an individual to fulfill their commitments is not necessarily binary — how they fail can be important.

There are also other aspects to the failure of a commitment. Actions have time and location components. If George is a few minutes late picking Alice up, it may not affect the picnic. If he is an hour late, that might be catastrophic. If he has the wrong address, then even if he arrives at that (wrong) location at 10am, the success of the picnic is in danger. And if Harry can't find his way to the park, there won't be any soccer after lunch even if he successfully collected everyone and bought the beer just as he said he would. However, as long as he arrives while the picnic is going on, then his passengers have a chance to enjoy themselves, though the later he arrives, the less chance that they will have a good time.

3 Reasoning about Trust

As discussed above, a key aspect of trust is that it stems from the relationship between individuals or groups of individuals. This means that it is a relative notion — Alice and Bob may have different views about Carol's trustworthiness — and thus that *provenance* is important in reasoning about trust [6]. A situation that often arises is one where it is necessary to combine different people's information about trust and when this is done, it is important to know where information about trust is coming from.

In this context, Jøsang et al. [12] distinguish between *functional* trust, the trust in an individual to carry out some task, and *referral* trust, the trust in an individual's recommendation. Thus, in our example, Alice's reasoning about George's offer of a lift, and Carol's offer to bring chicken are *functional* trust — Alice is thinking about George's reliability as a provider of lifts and Carol's reliability as a provider of chicken. However, if Alice were to ask Carol for a recommendation for a good butcher, then Alice would base her assessment of Carol's answer on her (Alice's) assesssment of Carol's ability to make good recommendations, an instance of referral trust, while what Carol expresses about her butcher is another instance of functional trust.

As [12] points out, the fact that Carol trusts her butcher to supply good meat is not necessarily a reason for Alice to do the same, and it certainly isn't a reason for Alice to trust the butcher in any more general context (to do a good job of painting Alice's house, for example). However, under certain circumstances — and in particular when the trust context is the same, as it is when Alice is considering the use of Carol's butcher as a provider of meat [14][5] — it is reasonable to consider trust to be transitive. Thus Alice

[5] Depending on the butcher, of course, even this might be too broad a trust context — perhaps the butcher provides excellent chicken and beef, but can only supply indifferent pork and his game has never been hung for long enough.

can consider combining her direct assessment of Carol's referral trustworhiness in the food domain, with Carol's direct assessment of her butcher's functional trustworthiness to derive an *indirect* functional assessment of the butcher.

Given this transitivity, the notion of a *trust network* then makes sense. If Alice can estimate the referral trustworthiness of her friends, and they can do the same for their friends, then Alice can make judgements about recommendations she receives not just from her friends, but also from the friends of her friends (and their friends and so on). The question is, what is a reasonable way to represent this computationally?

At the moment there is no definitive answer to the question. As the definitions of trust cited above suggest, one way to model trust is to use some form of subjective probability — Alice's degree of trust in Bob's park recommendation is a measure of her belief that she will like the park since Bob says that he likes the park. *Eigentrust* [15] is a mechanism, derived for use in peer-to-peer networks, for establishing a global trust rating that estimates how much any individual should trust another. While such a global rating, based as it is on performance, is reasonable for peer-to-peer systems, it has been argued [6] that in the kind of social networks we are discussing here, it is necessary to capture the fact that, for example, Alice and Bob can have very different estimations of Carol's trustworthiness (and, as we have argued, that they will have different ratings for Carol's trustworthiness in different contexts).

Subjective logic [13] is a formalism for capturing exactly this aspect of trust, and for inferring the degree of trust existing between two nodes in a trust network. Based on the Dempster-Shafer theory of evidence [25] it computes a measure that is a generalisation of probability, distinguishing belief in the reliability of an individual, disbelief in the reliability, and the potential belief that has not yet been determined one way or another (termed the "uncertainty"). Singh and colleagues [10, 27] provide extensions of the approach, the former looking at how best to update the measure of trust one individual has in another depending on their experience of interactions. Thus Alice may have her high regard for Carol's food-related recommendations damaged by a bad experience with a supplier that Carol recommends. Subjective logic is not the only approach to handling this problem. For example, Katz and Golbeck [16] describe an algorithm called TidalTrust for establishing the trust between a *source* node (representing the individual doing the trusting) and a *sink* mode (representing the individual being trusted). Later work by Kuter and Golbeck provides the SUNNY algorithm [18] which is reported to outperform TidalTrust on a benchmark database of trust information.

4 Argumentation and Trust

The Trust field, including sample literature discussd above, gives us methodologies for *computing* trust, while the Argumentation field can give us methodologies for *reasoning* about trust. In short, we believe that argumentation can provide a mechanism for handling many of the aspects that we need to capture about trust, as we discuss at some length in this section.

4.1 Argumentation in general

As we have discussed above, there are two major aspects that need to be handled by any representation of trust — we need to handle measures of trust, and we need to handle the provenance of trust information. Both of these are provided by several existing argumentation systems.

Some approaches to argumentation, for example abstract approaches such as that of Dung [3] and its derivatives, treat arguments as atomic objects. As a result, they say little or nothing about the internal structure of the argument and have no mechanism to represent the source of the information from which the argument is constructed. Such systems can represent the relationship between arguments ("*a* attacks *b*", and "*b* attacks *c*"), but cannot represent *why* this is the case. As a result, such systems cannot capture the fact that *a* attacks *b* because *b* is based on information from source *s*, and there is evidence that source *s* is not trustworthy.

There are, however, a number of existing systems that extend [3] with more detailed information about the argument. One system system is that of Amgoud [1], where an argument is taken to be a pair (H,h), *h* being a formula, the *conclusion* of the argument, and *H* being a set of formulae known as the *grounds* or *support* of the argument. Conclusion and support are related. In particular, [1] requires that *H* be a minimal consistent set of formulae such that $H \vdash h$ in the language in which *h* and *H* are expressed. This means of representing the support is rather restricted. It presents the support as a bag of formulae with no indication as to how they are used in the construction of the argument, and without recording any of the intermediate steps. It is easy enough to see if another argument *rebuts* (H,h), meaning that the conclusion of this second argument is the negation of *h*, and it is also quite simple to establish if the conclusion of the second argument contradicts any of the grounds in *H* (which in some systems of argumentation is known as *undercutting*). However, other forms of relationship are harder to establish. For example, in some cases it is interesting to know if an argument contradicts any of the intermediate steps in the chain of inferences between *H* and *h*.

Since the information about the steps in the argument can be useful, some systems of argumentation, for example [5] and [22], record more detail about the proof of *h* from *H* as part of the grounds. Some, including the system [19] which we will discuss in more detail below, go as far as to record the proof rules used in deriving *h* from *H*, permitting the notion of "attack" to include not only the intermediate conclusions but also the means by which they were derived.

Another problem with Dung's argumentation system from the perspective of reasoning about trust is that it has no explicit means to represent degrees of trust. In [3] the important question is whether, given all the arguments that are known, a specific argument should be considered to hold. While one could construct a system for reasoning about trust in this way — the critical point, after all, is often whether someone's argument is trustworthy or not — the prevelance of numerical measures of trust in the literature leads us to want to represent these.

Systems like that of Amgoud [1] provide one means of handling such measures, allowing formulae to have preference values attached to them. The values propagate to arguments and are taken into consideration when reasoning about the relationship between arguments (roughly speaking, strong arguments shrug off the attacks of weaker arguments). This approach seems a little too restrictive for dealing with trust, but there are systems that are more flexible. One example is the work of Oren et al. [21], which allows formulae and arguments to be weighted with the belief values used by Jøsang's subjective logic [13]. A more abstract approach is that of Fox [17] where values to represent belief in formulae are picked from some suitable *dictionary* of values, and propagated in a suitable way through the proof rules that are used to construct arguments. Arguments are then triples of conclusion, support, and value, and such systems are close to the notion of a *labelled deductive system* [2] (though they pre-date labelled deductive systems by some years).

4.2 A suitable argumentation system

Having given a high level description of how argumentation can help in handling a number of the aspects of reasoning about trust, we give a more detailed example of using a specific system of argumentation. The system we describe is the system *TL* that we introduced in [19], notable because it explicitly represents the rules of inference employed in constructing arguments in the support of the argument (which then makes it possible to dispute the application of those rules).

We start with a set of atomic propositions including $\top$ and $\bot$, the ever true and ever false propositions. The set of well-formed formulae (*wff*s), labeled $\mathcal{L}$, is comprised of the set of atomic propositions closed under the connectives $\{\neg, \rightarrow, \wedge, \vee\}$. $\mathcal{L}$ may then be used to create a database $\varDelta$ whose elements are 4-tuples:

$$(\theta : G : R : \tilde{d})$$

in which each element θ is a formulae, G is the derivation of that formula, R is the sequence of rules of inference used in the derivation, and $\tilde{d}$ is a suitable measure.

In more detail, θ is a *wff* from $\mathcal{L}$, $G = (\theta_0, \theta_1, \ldots, \theta_{n-1})$ is an ordered sequence of *wff*s, with $n \geq 1$, and $R = (\vdash_1, \vdash_2, \ldots, \vdash_n)$ is an ordered sequence of inference rules, such that:

$$\theta_0 \ \vdash_1 \ \theta_1 \ \vdash_2 \ \theta_2 \ \ldots \ \theta_{n-1} \ \vdash_n \ \theta$$

In other words, each element $\theta_k \in G$ is derived from the preceding element θ_{k-1} as a result of the application of the k-th rule of inference, $\vdash_k$, $(k = 1, \ldots, n-1)$. The rules of inference in any such sequence may be non-distinct. Thus G and R together provide an explicit representation of the way that θ was inferred.

The element $\tilde{d} = (d_1, d_2, \ldots, d_n)$ is an ordered sequence of elements from some dictionary $\mathcal{D}$. For reasoning about trust, these elements could be a numerical measure of trust, or some linguistic term that indicates the trust in the relevant inference, for example:

$$\{\textit{very reliable}, \textit{reliable}, \textit{no opinion}, \textit{somewhat unreliable}, \textit{very unreliable}\}$$

We also permit *wff*s $\theta \in \mathcal{L}$ to be elements of $\varDelta$, by including tuples of the form $(\theta : \emptyset : \emptyset : \emptyset)$, where each $\emptyset$ indicates a null term. (Such tuples represent information that has not been derived — basic premises may take this form.) Note that the assignment of labels may be context-dependent, i.e., the d_i assigned to $\vdash_i$ may also depend on θ_{i-1}.

$$\text{Ax}\,\frac{(\theta : G : R : \tilde{d}) \in \Delta}{\Delta \vdash_{TCR} (\theta : G : R : \tilde{d})}$$

$$\wedge\text{-I}\,\frac{\Delta \vdash_{TCR} (\theta : G : R : \tilde{d}) \text{ and } \Delta \vdash_{TCR} (\phi : H : S : \tilde{e})}{\Delta \vdash_{TCR} (\theta \wedge \phi : G \otimes H \otimes (\theta \wedge \phi) : R \otimes S \otimes (\vdash_{\wedge\text{-I}}) : \tilde{d} \otimes \tilde{e} \otimes (d_{\wedge\text{-I}}))}$$

$$\wedge\text{-E1}\,\frac{\Delta \vdash_{TCR} (\theta \wedge \phi : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\theta : G \otimes (\theta) : R \otimes (\vdash_{\wedge\text{-E1}}) : \tilde{d} \otimes (d_{\wedge\text{-E1}}))}$$

$$\wedge\text{-E2}\,\frac{\Delta \vdash_{TCR} (\theta \wedge \phi : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\phi : G \otimes (\phi) : R \otimes (\vdash_{\wedge\text{-E2}}) : \tilde{d} \otimes (d_{\wedge\text{-E2}}))}$$

$$\vee\text{-I1}\,\frac{\Delta \vdash_{TCR} (\theta : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\theta \vee \phi : G \otimes (\theta \vee \phi) : R \otimes (\vdash_{\vee\text{-I1}}) : \tilde{d} \otimes (d_{\vee\text{-I1}}))}$$

$$\vee\text{-I2}\,\frac{\Delta \vdash_{TCR} (\phi : H : S : \tilde{e})}{\Delta \vdash_{TCR} (\theta \vee \phi : H \otimes (\theta \vee \phi) : S \otimes (\vdash_{\vee\text{-I2}}) : \tilde{e} \otimes (e_{\vee\text{-I2}}))}$$

$$\vee\text{-E}\,\frac{\begin{array}{c}\Delta \vdash_{TCR} (\theta \vee \phi : G : R : \tilde{d}) \text{ and}\\ \Delta, (\theta : \emptyset : \emptyset : \emptyset) \vdash_{TCR} (\gamma : H : S : \tilde{e}) \text{ and } \Delta, (\phi : \emptyset : \emptyset : \emptyset) \vdash_{TCR} (\gamma : J : T : \tilde{f}).\end{array}}{\Delta \vdash_{TCR} (\gamma : G \otimes H \otimes J \otimes (\gamma) : R \otimes S \otimes T \otimes (\vdash_{\vee\text{-E}}) : \tilde{d} \otimes \tilde{e} \otimes \tilde{f} \otimes (d_{\vee\text{-E}}))}$$

$$\neg\text{-I}\,\frac{\Delta, (\theta : \emptyset : \emptyset : \emptyset) \vdash_{TCR} (\bot : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\neg\theta : G \otimes (\neg\theta) : R \otimes (\vdash_{\neg\text{-I}}) : \tilde{d} \otimes (d_{\neg\text{-I}}))}$$

$$\neg\text{-E}\,\frac{\Delta \vdash_{TCR} (\theta : G : R : \tilde{d}) \text{ and } \Delta \vdash_{TCR} (\neg\theta : H : S : \tilde{e})}{\Delta \vdash_{TCR} (\bot : G \otimes H \otimes (\bot) : R \otimes S \otimes (\vdash_{\neg\text{-E}}) : \tilde{d} \otimes \tilde{e} \otimes (d_{\neg\text{-E}}))}$$

$$\neg\neg\text{-E}\,\frac{\Delta \vdash_{TCR} (\neg\neg\theta : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\theta : G \otimes (\theta) : R \otimes (\vdash_{\neg\neg\text{-E}}) : \tilde{d} \otimes (d_{\neg\neg\text{-E}}))}$$

$$\rightarrow\text{-I}\,\frac{\Delta, (\theta : \emptyset : \emptyset : \emptyset) \vdash_{TCR} (\phi : G : R : \tilde{d})}{\Delta \vdash_{TCR} (\theta \rightarrow \phi : G \otimes (\theta \rightarrow \phi) : R \otimes (\vdash_{\rightarrow\text{-I}}) : \tilde{d} \otimes (d_{\rightarrow\text{-I}}))}$$

$$\rightarrow\text{-E}\,\frac{\Delta \vdash_{TCR} (\theta : G : R : \tilde{d}) \text{ and } \Delta \vdash_{TCR} (\theta \rightarrow \phi : H : S : \tilde{e})}{\Delta \vdash_{TCR} (\phi : G \otimes H \otimes (\phi) : R \otimes S \otimes (\vdash_{\rightarrow\text{-E}}) : \tilde{d} \otimes \tilde{e} \otimes (d_{\rightarrow\text{-E}}))}$$

Fig. 1. The TL Consequence Relation

This is the case for statistical inference, where the p-value depends on characteristics of the sample from which the inference is made, such as its size.

With this formal system, we can take a database Δ and use the consequence relation $\vdash_{TCR}$ defined in Figure 1 to build arguments for propositions of interest. This consequence relation is defined in terms of rules for building new arguments from old. The rules are written in a style similar to standard Gentzen proof rules, with the antecedents of the rule above the horizontal line and the consequent below. In Figure 1, we use the notation $G \otimes H$ to refer to that ordered sequence created from appending the elements of sequence H after the elements of sequence G, each in their respective order. The rules are as follows:

Ax The rule Ax says that if the tuple $(\theta : G : R : \tilde{d})$ is in the database, then it is possible to build the argument $(\theta : G : R : \tilde{d})$ from the database. The rule thus allows the construction of arguments from database items.

$\wedge$-I The rule $\wedge$-I says that if the arguments $(\theta : G : R : \tilde{d})$ and $(\phi : H : S : \tilde{e})$ may be built from the database, then an argument for $\theta \wedge \phi$ may also be built. The rule thus shows how to introduce arguments about conjunctions; using it requires an inference of the form: $\theta, \phi \vdash (\theta \wedge \phi)$, which we denote

$$\vdash_{\wedge\text{-I}}$$

in Figure 1. This inference is then assigned a value of $d_{\wedge\text{-I}}$.

$\wedge$-E The rule $\wedge$-E1 says that if it is possible to build an argument for $\theta \wedge \phi$ from the database, then it is also possible to build an argument for θ. Thus the rule allows the elimination of one conjunct from an argument, and its use requires an inference of the form: $\theta \wedge \phi \vdash \theta$. $\wedge$-E2 allows the elimination of the other disjunct.

$\vee$-I The rule $\vee$-I1 allows the introduction of a disjunction from the left disjunct and the rule $\vee$-I2 allows the introduction of a disjunction from the right disjunct.

$\vee$-E The rule $\vee$-E allows the elimination of a disjunction and its replacement by tuple when that tuple is a TL-consequence of each disjunct.

$\neg$-I The rule $\neg$-I allows the introduction of negation.

$\neg$-E The rule $\neg$-E allows the derivation of $\perp$, the ever-false proposition, from a contradiction.

$\neg\neg$-E The rule $\neg\neg$-E allows the elimination of a double negation, and thus permits the assertion of the Law of the Excluded Middle (LEM).

$\rightarrow$-I The rule $\rightarrow$-I says that if on adding a tuple $(\theta : \emptyset : \emptyset : \emptyset)$ to a database, where $\theta \in \mathcal{L}$, it is possible to conclude ϕ, then there is an argument for $\theta \rightarrow \phi$. The rule thus allows the introduction of $\rightarrow$ into arguments.

$\rightarrow$-E The rule $\rightarrow$-E says that from an argument for θ and an argument for $\theta \rightarrow \phi$ it is possible to build an argument for ϕ. The rule thus allows the elimination of $\rightarrow$ from arguments and is analogous to MP in standard propositional logic.

This is an intentionally abstract formalism — syntactically complete, but without a specified semantics. The idea is that to capture a specific domain, we have to identify a suitable dictionary from which to construct the $\tilde{d}$ and that this set of values will determine the mechanism by which we can compute an overall value from the sequence of d_i. For example, if one wanted to use Jøsang's subjective logic, then the mechanism for combining the d_i's would be taken from [13]. If one wanted to quantify trust using probability, then the combination rules would be those dictated by probability theory (for example using [28]). If one wanted to use the dictionary mentioned above ("very reliable" and so on) then it would be necessary to determine the right way to combine these values across all the inference rules in Figure 1.

Even without specifying these mechanisms, it should be clear that whatever means we use to quantify trust in combination with *TL*, the formalism can both capture trust values and the precise source of information used. It is also possible to go further. The fact that *TL* includes explicit reference to different forms of inference allows us to capture the fact that inferences may differ depending on the source of the information on which they are based — we might want to make different inferences depending on whether the source was something we have direct experience of or something that comes from a trusted source, or something that comes from an untrusted source.

4.3 Extensions

The previous sections have argued that systems of argumentation can provide the core functionality required to reason about trust. Here we discuss how systems of argumentation, especially the system *TL* sketched above, can provide additional mechanisms that are important in dealing with trust.

First, argumentation systems explicitly allow the representation of different points of view. The system *TL* we have sketched above provides us with the rules for constructing arguments, and it does not limit the number of arguments that one can construct for a specific conclusion. Thus, the database Δ may contain information that represents a number of different assessments of the trustworthiness of, for example, a source of information. This might be done through the inclusion of a number of tuples $(\theta : G : R : \tilde{d})$ with different Gs, representing different views of the sources, and different $\tilde{d}$s representing different assessments of trustworthiness. These pieces of information could then be used to make different inferences, with any potential choice between conclusions being made on the basis of the relevant $\tilde{d}$ values.

That is one, fairly simple, way to represent different viewpoints. Another would be to have different argumentation systems represent the views of different individuals, and to use the mechanisms of argumentation-based dialogue (like those discussed in [24, 8]) to explore the differences in the views of trust and to attempt to resolve them. In such a combination, the individual argumentation systems can be constructed using *TL*, and would then reason about trust based on a single viewpoint. The interaction between different viewpoints is then captured by the dialogue mechanisms of [24, 8], enabling a rational discourse about trust issues.

Another important aspect of reasoning about trust, addressed in [10] for example, is the need for an individual to be able to revise the trust they have in another based on experience. Revision of beliefs is not a subject that has been widely considered within the argumentation community, but [23] suggests some approaches to the subject, and these can be implemented on top of *TL*. This would allow us to represent the case in which one individual revises its view of a source as a result of considering information provided by another individual.

5 Conclusion

This paper has presented the case for using argumentation as a mechanism for reasoning about trust. Starting from some of the many views of trust expressed in the literature, we extracted the major features that need to be represented, discussed formalisms for handling trust, and then suggested how argumentation could be used for reasoning about trust. We sketched in some detail how a specific system of argumentation, *TL*, could be used in this way and identified some additional argumentation-based mechanisms that could be of use in dealing with trust.

Acknowledgement

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.

References

1. L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.
2. C.Chesñevar and G. Simari. Modelling inference in argumentation through labelled deduction: Formalization and logical properties. *Logica Universalis*, 1(1):93–124, 2007.
3. 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.
4. D. Gambetta. Can we trust them? In D. Gambetta, editor, *Trust: Making and beraking cooperative relations*, pages 213–238. Blackwell, Oxford, UK, 1990.
5. A. J. Garcia and G. R. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(2):95–138, 2004.
6. 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.
7. J. Golbeck and C. Halaschek-Wiener. Trust-based revision for expressive web syndication. *The Logic Journal of the IGPL*, (to appear).
8. T. F. Gordon. The pleadings game: An exercise in computational dialectics. *Artificial Intelligence and Law*, 2(4):239–292, 1994.
9. T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 4(4):2–16, 2000.
10. 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 Multiagent Systems*, 2008.
11. E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK, 2003.

12. A. Jøsang, E. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.
13. A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Society Conference*, Hobart, January 2006.
14. A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
15. 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.
16. Y. Katz and J. Golbeck. Social network-based trust in prioritzed default logic. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.
17. P. Krause, S. Ambler, M. Elvang-Gørannson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11 (1):113–131, 1995.
18. 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.
19. 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.
20. D. H. McKnight and N. L. Chervany. The meanings of trust. Working Paper 96-04, Carlson School of Management, University of Minnesota, 1996.
21. N. Oren, T. Norman, and A. Preece. Subjective logic and arguing with evidence. *Artificial Intelligence*, 171(10–15):838–854, 2007.
22. 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.
23. S. Parsons and E. Sklar. How agents alter their beliefs after an argumentation-based dialogue. In S. Parsons, N. Maudet, P. Moraitis, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems, Second International Workshop*, volume 4049 of *Lecture Notes in Computer Science*, pages 297–312. Springer, 2005.
24. H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.
25. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.
26. P. Sztompka. *Trust: A Sociological Theory*. Cambridge University Press, Cambridge, UK, 1999.
27. 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*, 2006.
28. Y. Xiang and N. Jia. Modeling causal reinforcement and undermining for CPT elicitation. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1708–1718, 2007.

Computing Argumentation in Polynomial Number of BDD Operations: A Preliminary Report

Yuqing Tang[1], Timothy J. Norman[2], and Simon Parsons[1,3]

[1] Dept. of Computer Science, Graduate Center, City University of New York
365 Fifth Avenue, New York, NY 10016, USA
ytang@gc.cuny.edu
[2] Dept of Computing Science, The University of Aberdeen
Aberdeen, AB24 3UE, UK
t.j.norman@abdn.ac.uk
[3] Dept of Computer & Information Science, Brooklyn College, City University of New York,
2900 Bedford Avenue, Brooklyn, NY 11210 USA
parsons@sci.brooklyn.cuny.edu

Abstract. Many advances in argumentation theory have been made, but the exponential complexity of argumentation-based reasoning has made it impractical to apply argumentation theory. In this paper, we propose a binary decision diagram (BDD) approach to argumentation-based reasoning. In the approach, sets of arguments and defeats are encoded into BDDs so that an argumentation process can work on a set of arguments and defeats simultaneously in one BDD operation. As a result, the argumentation can be computed in polynomial number of BDD operations on the number of input sentences.

1 Introduction

Argumentation provides an elegant approach to nonmonotonic reasoning [15] and decision making [17, 26], and now sees wide use as a mechanism for supporting dialogue in multiagent systems [32, 33]. As an approach that has its roots in logic — in many systems of argument, the arguments are constructed using some form of logical inference — the efficiency of reasoning using argumentation is a topic of considerable interest [13, 16, 25] with a number of negative results that stress the fact that generating arguments and establishing properties of arguments can be very costly in computational terms.

In this paper we take a rather different look at the computation of arguments. We have been investigating the creation of multiagent plans [36–38], especially the construction of plans that take into account the communication between agents [34, 35]. In doing so, we have been using a representation, that of quantified boolean formulae (QBFs) and binary decision diagrams (BDDs), which has been widely adopted in symbolic planning in non-deterministic domains. It turns out that this representation provides a way to compute arguments, and given the computational efficiency of planning based on QBFs and BDDs, it seems that it can provide an efficient way to compute arguments.We investigate exactly how efficient this approach is in this paper and conclude that we can carry out many of the basic operations needed to compute arguments in a polynomial number of operations.

Note that we are not claiming to be performing general logical inference in polynomical time. As we explain in detail later in the paper, the "polynomial number of opertions" are operations on the BDD representation, and while this representation in many cases can be constructed compactly from a set of logical formulae, there are some cases in which the size of this representation is exponential in the number of formulae.

2 Background

This section gives the technical background needed by the remainder of the paper, a description of *quantified boolean formulae*, and *binary decision diagrams*.

2.1 Qantified boolean formulae

A propositional language $\mathcal{L}$ based on a set of proposition symbols $\mathcal{P}$ with quantification can be defined by allowing standard connectives $\wedge, \vee, \rightarrow, \neg$ and quantifiers $\exists, \forall$ over the proposition variables. The resulting language is a logic of quantified boolean formulae (QBF) [5]. A *symbol renaming operation*, which we use below, can be defined on $\mathcal{L}$, denoted by $\mathcal{L}[\mathcal{P}/\mathcal{P}']$, which means that a new language is obtained by substituting the symbols of $\mathcal{P}$ with the symbols of $\mathcal{P}'$ where $\mathcal{P}'$ contains the same set of propositions as that of $\mathcal{P}$ but using different symbol names (notice that $|\mathcal{P}'| = |\mathcal{P}|$). Similarly for a formula $\xi \in \mathcal{L}$, if $\boldsymbol{x}$ is a vector of propositional variables for $\mathcal{P}$, then a variable renaming operation can be defined by $\xi[\boldsymbol{x}/\boldsymbol{x}']$ which means that all the appearances of variables $\boldsymbol{x} = x_1 x_2 \ldots x_n$ are substituted by $\boldsymbol{x}' = x'_1 x'_2 \ldots x'_n$ which is a vector of the corresponding variables or constants in $\mathcal{P}'$. In QBF, propositional variables can be universally and existentially quantified: if $\phi[\boldsymbol{x}]$ is a QBF formula with propositional variable vector $\boldsymbol{x}$ and x_i is one of its variables, the existential quantification of x_i in ϕ is defined as $\exists x_i \phi[\boldsymbol{x}] = \phi[\boldsymbol{x}][x_i/FALSE] \vee \phi[\boldsymbol{x}][x_i/TRUE]$ and the universal quantification of x_i in ϕ is defined as $\forall x_i \phi[\mathbf{x}] = \phi[\boldsymbol{x}][x_i/FALSE] \wedge \phi[\boldsymbol{x}][x_i/TRUE]$. Here $FALSE$ and $TRUE$ are two propositional constants representing "true" and "false" in the logic. Quantifications over a set $X = \{x_1, x_2, \ldots, x_n\}$ of variables is defined as sequantial quantifications over each variables x_i in the set:

$$Q_X \xi = Q_{x_n} Q_{x_{n-1}} \ldots Q_{x_1} \xi$$

where Q is either $\exists$ or $\forall$. The introduction of quantification doesn't increase the expressive power of propositional logic but allows us to write concise expressions whose quantification-free versions have exponential sizes [11].

With above language, we can encode sets and relations to manipulate sets of arguments and defeats. Let x be an element of a set $X = 2^{\mathcal{P}}$, x can then be explicitly encoded by a conjunction composed of all proposition symbols in $\mathcal{P}$ in either positive or negative form

$$\xi(x) = \bigwedge_{p_i \in x} p_i \wedge \bigwedge_{p_j \notin x \text{ and } p_j \in \mathcal{P}} \neg p_j$$

where $p_i \in x$ means that the corresponding bit p_i is set to be $TRUE$ in the encoding of x, and $p_j \notin x$ means that the corresponding bit p_j is set to be $FALSE$ in the encoding

Set operator	QBF operator
$X_1 \cap X_2$	$\xi(X_1) \wedge \xi(X_2)$
$X_1 \cup X_2$	$\xi(X_1) \vee \xi(X_2)$
$X_1 \setminus X_2$	$\xi(X_1) \wedge \neg\xi(X_2)$
$x \in X$	$\xi(x) \rightarrow \xi(X)$
$X_1 \subseteq X_2$	$\xi(X_1) \rightarrow \xi(X_2)$

Table 1. The mapping between set operators and QBF operators

of x. We denote that a formula γ can be satisfied in an element x by $x \models \gamma$. Then a set of elements can be characterized by a formula $\gamma \in \mathcal{L}$, with the set denoted by $X(\gamma)$, where $X(\gamma) = \{x | x \models \gamma\}$.[4] Two special sets, the empty set $\emptyset$ and the universal set $\mathcal{U}$, are represented by $FALSE$ and $TRUE$ respectively.

With these notions we can have a mapping between the set operations on states and the boolean operations on formulae as shown in Table 1 when X_1 and X_2 are interpreted as two sets of states.

2.2 Binary decision diagrams

In the above, we have showed the natural connections between the set paradigm and its implicit representation using QBF formulae. Now we will briefly survey that the QBF formulae and the operations over them can be represented and efficiently computed using a data structure called Binary Decision Diagrams (BDD) [5]. In this way, the time and space complexity for exploring the space of arguments and defeats for acceptable arguments can be significantly reduced due to the compact representation provided by BDDs in comparison to explicit search techniques.

A BDD is a rooted directed acyclic graph. The terminal nodes are either $TRUE$ or $FALSE$. Each non-terminal node is associated with a boolean variable x_i, and two BDDs, called left and right, corresponding to the values of the sub-formula when x_i is assign $FALSE$ and $TRUE$ respectively. The value of a QBF formula can be determined by traversing the graph from the root to the leaves following the boolean assignment given to the variables of the QBF formula. The advantage of using BDDs to represent QBF formulae is that most basic operations on QBFs can be performed in linear or quadratic time in terms of the number of nodes used in a BDD representation of the formulae if a special form of BDD, called Reduced Ordered Binary Decision Diagram (ROBDD) [5], is used. A ROBDD is a compact BDD which uses a fixed ordering over the variables from the root to the leaves in the BDD, merges duplicate subgraphs into one, and directs all their incoming edges into the merged subgraph. Following the notation traditionally used in symbolic model checking and AI planning, we will refer to an ROBDD simply as a BDD.

Let ξ, ξ_1, ξ_2 be QBF formulae, let the number of nodes used in its BDD representation denoted by $|| \cdot ||$. With this BDD representation, the complexity of a QBF binary operator $\langle op \rangle$ (e.g. $\wedge, \vee, \rightarrow$) on two formulae ξ_1 and ξ_2, namely $\xi_1 \langle op \rangle \xi_2$, is

[4] Note that $X(p_1 \wedge p_2 \wedge \ldots \wedge p_k) \neq \{s\}$ where $x = \{p_1, p_2, \ldots, p_k\}$.

QBF/Set operator	BDD operator	Complexity
$\neg\xi$	$\neg G(\xi)$	$O(\|\|\xi\|\|)$
$\exists x_i(\xi)$	$G(\xi_{x_i=0}) \vee G(\xi_{x_i=1})$	$O(\|\|\xi\|\|^2)$
$\forall x_i(\xi)$	$G(\xi_{x_i=0}) \wedge G(\xi_{x_i=1})$	$O(\|\|\xi\|\|^2)$
$\xi_1 \wedge \xi_2$	$G(\xi_1) \wedge G(\xi_2)$	$O(\|\|\xi_1\|\| \cdot \|\|\xi_2\|\|)$
$\xi_1 \vee \xi_2$	$G(\xi_1) \vee G(\xi_2)$	$O(\|\|\xi_1\|\| \cdot \|\|\xi_2\|\|)$
$\xi_1 \rightarrow \xi_2$	$G(\xi_1) \rightarrow G(\xi_2)$	$O(\|\|\xi_1\|\| \cdot \|\|\xi_2\|\|)$
$\|X\|$	$Sat\text{-}count(G(\xi(X)))$	$O(\|\|\xi(X)\|\|)$

Table 2. The mapping between QBF operators and BDD operators. ξ, ξ_1, ξ_2 are formulae in QBF; $G(\xi), G(\xi_1), G(\xi_2)$ are BDD representations for these formulae.

$O(||\xi_1|| \times ||\xi_2||)$, that of negation $\neg\xi$ is $O(||\xi||)$ (or $O(1)$ if complement edges are introduced to the BDDs), and that of quantification $Q_{x_i}\ (f[\boldsymbol{x}])$, where Q is either $\exists$ or $\forall$, is $O(||f||^2)$ [5, 11] as summarized in Table 2.

The key achievement of using BDDs (and the front end language of QBFs) to represent sets and relations is that the complexity of the operations will depend on the complexity of the BDD representation instead of the size of the sets and relations, and the complexity of the BDD representation of the sets and relations doesn't depend on the size of those sets and relations. Instead, the operations on BDDs are polynomial in the size of the BDD, and so operations on sets and relations will be polynomial in the size of their BDD representation rather than exponential in their size.

3 Set-theoretic argumentation

Having introduced the ideas from QBFs and BDDs, in this section we give an overview of the argumentation system we will capture using them. The framework we use is mostly drawn from the work of Amgoud and her colleagues [1, 2] with some slight modifications. This framework will abstract away the inference procedure by which the arguments are created and only keep track of the premises the arguments are based on. In the next section, we will introduce the inference procedure back into the representation of arguments.

Definition 1. *An argument based on $\Sigma \subseteq \mathcal{L}$ is pair (H, h) where $H \subseteq \Sigma$ and $H \neq \emptyset$ such that*

1. *H is consistent with respect to $\mathcal{L}$,*
2. *$H \vdash h$,*
3. *H is minimal (for set inclusion).*

H is called the support and h is called the conclusion of the argument. $\mathcal{A}(\Sigma)$ denotes the set of all arguments which can be constructed from Σ.

This definition of argument can be understood as a set of constraints on how information can be clustered as arguments. Condition (1) is to ensure that an argument is coherent.

The coherence of an agent's information is defined in terms of the consistency of the language $\mathcal{L}$ in which the information is written. Condition (2) can be understood as insisting that the conclusion of an argument should be supported by a set of information in the sense of inference in the language $\mathcal{L}$. Condition (3) can be understood as saying that no redundant information should appear in an argument.

Definition 2. *(H', h') is a subargument of the argument (H, h) iff $H' \subseteq H$.*

Definition 3. *Let (H_1, h_1), (H_2, h_2) be two arguments of $\mathcal{A}(\Sigma)$.*

1. *(H_1, h_1) rebuts (H_2, h_2) iff $h_1 \equiv \neg h_2$.*
2. *(H_1, h_1) undercuts (H_2, h_2) iff $\exists h \in H_2$ such that $h_1 \equiv \neg h$.*
3. *(H_1, h_1) contradicts (H_2, h_2) iff (H_1, h_1) rebuts a subargument of (H_2, h_2).*

The binary relations rebut, undercut, *and* contradict *gather all pairs of arguments satisfying conditions (1), (2) and (3) respectively.*

Definitions of rebut, undercut, and contradict will be given below and we will collectively refer to the relations as defeat if no distinction is necessary. Following Dung's work [15], we have the following component definitions:

Definition 4. *An* argumentation framework *is a pair, $Args = \langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A}$ is a set of arguments, and $\mathcal{R}$ is the binary relation* defeat *over the arguments.*

Definition 5. *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework, and $S \subseteq \mathcal{A}$. An argument A is defended by S iff $\forall B \in \mathcal{A}$ if $(B, A) \in \mathcal{R}$ then $\exists C \in S$ such that $(C, B) \in \mathcal{R}$.*

Definition 6. *$S \subseteq \mathcal{A}$. $\mathcal{F}_{\mathcal{R}}(S) = \{A \in \mathcal{A} | A$ is defended by S with respect to $\mathcal{R}\}$.*

Now, for a function $F : D \to D$ where D is the domain and the range of the function, a fixed point of F is an $x \in D$ such that $x = F(x)$. When the D is associated with an ordering P — for example, P can be set inclusion over the power set D of arguments — x is a *least fixpoint* of F if x is a least element of D with respect to P and x is a fixed point.

Definition 7. *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework. The set of acceptable arguments, denoted by $Acc^F_{\mathcal{R}}$, is the least fixpoint of the function $\mathcal{F}_{\mathcal{R}}$ with respect to set inclusion.*

The least fixpoint semantics can be viewed as a mathematical translation of the principle such that an argument survives if it can defend itself and be defended by a set of arguments which can also survive all the attacks made upon them.

4 Representing arguments in QBFs and BDDs

We now turn our attention to using QBFs and BDDs to represent the components of an argumentation system, and then to perform the computations we need to carry out on that representation.

We can label each item $f_i \in \Sigma$ with a proposition l_i. Namely, we will extend the language $\mathcal{L}$ to contain both the information base Σ and the labels for these sentences. Formally, the proposition symbols can be extended to be $\mathcal{P} = \mathcal{P}_D \cup \mathcal{P}_L$ where $\mathcal{P}_D$ is the set of proposition symbols for the domain information, and $\mathcal{P}_L$ is the set of system proposition symbols labeling the sentences in Σ. Given a finite information base $\Sigma \subseteq \mathcal{L}$, $|\mathcal{P}_L(\Sigma)| = |\Sigma|$, namely each sentence $f_i \in \Sigma$ has a corresponding label l_i.

For any formula ξ in $\mathcal{L}$ based on $\mathcal{P} = \mathcal{P}_D \cup \mathcal{P}_L$, $\xi_D = \exists_{\mathcal{P}_L} \xi$ is the formula with only domain symbols left, and $\xi_L = \exists_{\mathcal{P}_D} \xi$ is the formula with only the label symbols left.

4.1 Labeling

For representational convenience, we define

$$SEL(l_i) = l_i \wedge \bigwedge_{j \neq i} \neg l_j.$$

A sentence f_i of Σ corresponds to a pair $\langle SEL(l_i), f_i \rangle$ which can be represented by $SEL(l_i) \wedge f_i$. Given a set of input information $\Sigma = \{f_i\}$ for $f_i \in \mathcal{L}_D$, a labeling table $\Lambda(\Sigma)$ can be expressed as follows

$$\Lambda(\Sigma) = \{\langle SEL(l_i), f_i \rangle\}$$

where $f_i \in \Sigma$ and $l_i \in \mathcal{P}_L$, and the corresponding QBF representation

$$\xi(\Lambda(\Sigma)) = \bigvee_{f_i \in \Sigma} [SEL(l_i) \wedge f_i]$$

The above $\Lambda(\Sigma)$ expression requires $2 \times |\Sigma|$ QBF/BDD operations. [5] Given a subset $\sigma \subseteq \Sigma$,

$$SEL(\sigma) = \bigwedge_{f_i \in \sigma} (l_i) \wedge \bigwedge_{f_j \notin \sigma} \neg l_j$$

4.2 Consistent subsets

Since the support of an argument is a consistent set of propositions, a natural place to start thinking about argument computation is with the computation of consistent subsets. The set of all consistent subsets of Σ is

$$CONS(\Sigma) = \bigvee_{\sigma \subseteq \Sigma} [SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i] \qquad (1)$$

[5] The first condition of using QBF/BDD is to guarantee a way to express the information/specification that we need with only polynomial, linear, or even logarithmic number of QBF/BDD operations; the second condition is to guarantee that the size of the initial, intermediate, and final BDDs corresponding to the information/specification is small enough to fit into memory. For the second condition, if the size of the BDD explodes we may partition the expression into conjunctions or disjunctions, and modify the algorithms manipulating these BDDs correspondingly to try to avoid the explosion. If this still fails, then it means that the problem cannot be efficiently handled by BDDs. In this case, it usually also means that some aspect of the information required to solve the problem is simply too complex.

Computing the above expression directly requires an exponential number of QBF/BDD operations, so we want to find another way to compute it.

Proposition 1. *$CONS(\Sigma)$ can be constructed using $2 \times |\Sigma| - 1$ operations as follows*

$$CONS(\Sigma) = \bigwedge_{f_i \in \Sigma} [l_i \rightarrow f_i]. \tag{2}$$

Proof. The form of formula 2 follows from

$$\begin{aligned} CONS(\Sigma) &= \bigwedge_{f_i \in \Sigma} [l_i \rightarrow f_i] \\ &= \bigwedge_{f_i \in \Sigma} [l_i \rightarrow (l_i \wedge f_i)] \\ &= \bigwedge_{f_i \in \Sigma} [\neg l_i \vee (l_i \wedge f_i)] \\ &= \bigvee_{\sigma \subseteq \Sigma} [\bigwedge_{f_j \notin \sigma} \neg l_j \wedge \bigwedge_{f_i \in \sigma} (l_i \wedge f_i)] \\ &= \bigvee_{\sigma \subseteq \Sigma} [SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i] \end{aligned}$$

$(l_i \rightarrow f_i) \leftrightarrow (l_i \rightarrow (l_i \wedge f_i))$ follows from:

$$\begin{aligned} A \rightarrow B &\leftrightarrow \neg A \vee B \\ &\leftrightarrow (\neg A \vee A) \wedge (\neg A \vee B) \\ &\leftrightarrow \neg A \vee (A \wedge B) \\ &\leftrightarrow A \rightarrow (A \wedge B) \end{aligned}$$

□

With the above expression, we can exclude empty consistent subsets by

$$CONS^+(\Sigma) = CONS(\Sigma) \wedge (\bigvee_{f_i \in \Sigma} l_k)$$

Because we are only interested in non-empty consistent subsets, from here on we will mean $CONS^+(\Sigma)$ when we use $CONS(\Sigma)$. The set of subsets of selected sentences is

$$\begin{aligned} CONS(\Sigma)_L = \; & \exists_{\mathcal{P}_D} \left(\bigvee_{\sigma \in \Sigma} (SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i) \right) \\ & = \bigvee_{\sigma \in \Sigma} (SEL(\sigma) \wedge \left[\exists_{\mathcal{P}_D} \left(\bigwedge_{f_i \in \sigma} f_i \right) \right] \\ & = \quad \bigvee_{\sigma \subseteq \Sigma} SEL(\sigma) \end{aligned}$$

As we see, the complexity of $CONS_L(\Sigma)$ is $O(2 \times |\Sigma| - 1 + |\mathcal{P}_D|)$. Let

$$CONJ(\sigma) = SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i.$$

Proposition 2. *Given a sentence set selector $\sigma \subseteq \Sigma$ represented by $SEL(\sigma)$, if the conjunction of the selected sentences in σ is consistent then it can be expressed as follows*

$$\begin{aligned} CONJ(\sigma) = &\quad SEL(\sigma) \wedge CONS(\Sigma) \\ &= \bigvee_{\sigma \subseteq \Sigma} [SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i] \end{aligned}$$

Proof. $CONS(\Sigma)$ is a disjunction of conjunctions of all consistent subsets of Σ. Among these conjunctions, $SEL(\sigma)$ only can make the one corresponding to the σ selection true, which is $SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i$, and others false. Namely $SEL(\sigma) \wedge CONS(\Sigma) = SEL(\sigma) \wedge \bigwedge_{f_i \in \sigma} f_i$ $\square$

Similarly, the set of conjunctions of a set of selected sentences can be expressed by:

$$CONJ(\{\sigma_i\}) = \bigvee_{\sigma_i} [SEL(\sigma_i)] \wedge CONS(\Sigma)$$

With this expression, we will be able to filter conbinations of consistent and inconsistent sets of sentences into consistent sets.

4.3 QBF/BDD representation of arguments

We can extend the language $\mathcal{P}$ further to contain $\mathcal{P} = \mathcal{P}_L \cup \mathcal{P}_D \cup \mathcal{P}_{L,C} \cup \mathcal{P}_{D,C}$ where $\mathcal{P}_{D,C}$ is a set of renaming symbols of $\mathcal{P}_D$ to represent the conclusions of arguments; $\mathcal{P}_{L,C}$ is an optional set of symbols to label an interesting sub-space of conclusions (the ones we want to compute arguments for). For example, if the sentences in Σ and their negations are of interest, then $P_{L,C} = 2log|\Sigma|$ (we don't need to label a set of sentences, instead we just need to label individual setences and their negations so that we need $2log|\Sigma|$ symbols). Similarly, we will denote $\mathcal{P}_D$ by $\mathcal{P}_{D,P}$ for premises when a distinction is needed.

An argument (H, h) in $\mathcal{L}_D$ can then be represented by formula $\xi(H, h)$ in $\mathcal{L}$

$$\xi(H, h) = SEL(H) \wedge \bigwedge_{f_i \in H} f_i \wedge h[\mathcal{P}_{D,C}]$$

where $h[\mathcal{P}_{D,C}]$ means the expression h is in terms of the symbols of $\mathcal{P}_{D,C}$.

The set of all arguments that can be constructed from Σ will be equivalent to

$$\mathcal{A}(\Sigma) = CONS(\Sigma)$$

for the moment by abstracting away the conclusions. Later we will reintroduce the conclusions to the representation during the query for conclusions and the defeat process.

4.4 Arguments for conclusions

We can construct the set of arguments for a set of conclusions all at once as follows. Let us assume that, besides the input information base Σ, we also have a set of conclusions C that we wish to support.

$$C = \{h_k\}$$

with $K = log|C|$ and a set of labeling symbols

$$\mathcal{P}_{L,C} = \boldsymbol{l}_C = \{l_{1,C}, \ldots, l_{K,C}\}$$

and let c_k be defined as $\boldsymbol{l}_C = k$, namely c_k is the encoding of integer k using the boolean symbols of $\mathcal{P}_{L,C}$.

The set of arguments for C based on Σ can be represented as

$$Args(\Sigma, C)_L = \forall_{\boldsymbol{x} \in \mathcal{P}_D \cup \mathcal{P}_{D,C}} \bigvee_{h_k \in C} \bigvee_{\sigma \subseteq \Sigma} [(\bigwedge_{f_i \in \Sigma} f_i \rightarrow h_k) \wedge SEL(\sigma) \wedge c_k)]$$

and results in

$$Args(\Sigma, C) = Args(\Sigma, C)_L \wedge CONS(\Sigma) \wedge \bigvee_{h_k \in C} (c_k \wedge h_k)$$

Proposition 3. $Args(\Sigma, C)_L$ *can be expressed as*

$$\forall_{\boldsymbol{x} \in \mathcal{P}_D \cup \mathcal{P}_{D,C}} CONS(\Sigma)_L \wedge \left[\bigvee_{h_k \in C} (c_k)\right] \wedge \left[\bigvee_{f_i \in \Sigma} (l_i \wedge \neg f_i) \vee \bigvee_{h_k \in C} (c_k \wedge h_k)\right]$$

using $O(2 \times |C|) + O(|\Sigma|) + O(2 \times |\Sigma| + |\mathcal{P}_D|) + O(|\mathcal{P}_D \cup \mathcal{P}_{D,C}|)$ *QBF/BDD operations.*

Proof. Start with the first two items above,

$$CONS(\Sigma)_L \wedge \left[\bigvee_{h_k \in C} (c_k)\right] \bigvee_{\sigma \subseteq \Sigma} \bigvee_{h_k \in H} [SEL(\sigma) \wedge c_k]$$

Conjoining with the remaining two items $\left[\bigvee_{f_i \in \Sigma} (l_i \wedge \neg f_i) \vee \bigvee_{h_k \in C} (c_k \wedge h_k)\right]$, gives:

$$\bigvee_{\sigma \subseteq \Sigma} \bigvee_{h_k \in C} \left[SEL(\sigma) \wedge c_k \wedge \left(\bigvee_{f_i \in \Sigma} (l_i \wedge \neg f_i) \vee \bigvee_{h_k \in C} (c_k \wedge h_k)\right)\right]$$

$$= \bigvee_{\sigma \subseteq \Sigma} \bigvee_{h_k \in \Sigma} \left[\left(SEL(\sigma) \wedge (\bigvee_{f_i \in \sigma} \neg f_i)\right) \vee (c_k \wedge h_k)\right]$$

$$= \bigvee_{\sigma \subseteq \Sigma} \bigvee_{h_k \in \Sigma} \left[SEL(\sigma) \wedge c_k \wedge \left(\bigwedge_{f_i \in \sigma} (f_i) \rightarrow (h_k)\right)\right]$$

The first line is derived using $\bigvee_i A_i \wedge \bigvee_j B_j = \bigvee_i \left[A_i \wedge \left(\bigvee_j B_j\right)\right]$. The second line is derived using

$$SEL(\sigma) \wedge (\bigvee_{f_i \in \Sigma} (l_i \wedge \neg f_i)) = SEL(\sigma) \wedge (\bigvee_{f_i \in \sigma} \neg f_i)$$

since:

$$SEL(\sigma) \wedge (l_i \wedge \neg f_i) = FALSE$$

for any $f_i \notin \sigma$. The second line also employs $c_k \wedge \bigvee_{h_k \in C}(c_k \wedge h_k) = c_k \wedge h_k$ □

Algorithm 4.1 Computing BDD for set-inclusion $\subseteq$

```
1: Associate with each element in $f_i \in \Sigma$ two BDD variables $l_i$ and $l'_i$.
2: Take the variable order $l_1, l'_1, l_2, l'_2, ..., l_n, l'_n$ $(n = |\Sigma|)$
3: for each $l_i$ do
4:     link $l_i = 1$ to $l'_i$
5:     link $l_i = 0$ to $l_{i+1}$
6: end for
7: for each $l'_i \neq l'_n$ do
8:     link $l'_i = 1$ to $l_{i+1}$
9:     link $l'_i = 0$ to terminal 0
10: end for
11: link $l'_n = 0$, to terminal 0
12: link $l'_n = 1$, to terminal 1
```

4.5 Minimization of consistent sets with respect to conclusions

In the above, $Args(\Sigma, C)$ may contain non-minimal arguments. To overcome this, we need to minimize the arguments in $Args(\Sigma, C)$ with respect to their conclusions. Given a set of arguments $Q \subseteq \mathcal{A}$ and a partial relation $B \subseteq \mathcal{A} \times \mathcal{A}$ (e.g. the set-inclusion $\subseteq$ relation on the supports of arguments) on $\mathcal{A}$, the set of minimal arguments in Q with respect to B is

$$Min(Q, B) = \{A \in Q | \text{for all } C \in Q, (C, A) \in B \text{ implies } (A, C) \in B\}$$

By encoding Q with a QBF formula $Q[\mathcal{P}]$ based on a set $\mathcal{P}$ of propositional symbols, and encoding the partial relation B with another QBF $B[\mathcal{P}, \mathcal{P}']$ with the first component of B based on symbols in $\mathcal{P}$ and the second component of B based on the symbols in $\mathcal{P}'$, we can compute $Min(Q, B)$ as fllows

$$Min(Q, B) = Q \wedge \forall_{\mathcal{Z}}[(Q[\mathcal{P}/\mathcal{Z}] \rightarrow (B[\mathcal{P}/\mathcal{Z}, \mathcal{P}'/\mathcal{P}] \rightarrow B[\mathcal{P}'/\mathcal{Z}])]$$

where $\mathcal{Z}$ is a temporary set of symbols renamed from $\mathcal{P}$ to hold the intermediate results during the computation.

The set-inclusion relation between two sets of supports $H_1[\mathcal{P}]$ and $H_2[\mathcal{P}']$ can be implemented as:

$$\xi(\subseteq) = \bigwedge_{f_i \in \Sigma} [l_i \rightarrow l'_i].$$

This requires $2 \times |\Sigma|$ QBF/BDD operations to construct. A linear BDD size implementation of $\subseteq$ on the supports of $\mathcal{A}$ is given in Algorithm 4.1 [6].

The set of minimal supports which attack a sentence $h_k \in C$ can be computed as

$$Args_{min}(\Sigma, h_k) = Min\left((Args(\Sigma, C) \wedge c_k)_L, \xi(\subseteq)\right).$$

The set of minimal supports with respect to each each setence in C can be computed as

$$Args_{min}(\Sigma, h_k) = \bigvee_{h_k \in C} Args_{min}(\Sigma, h_k).$$

For description convenience, below we will use $Args(\Sigma, C)$ for $Args_{min}(\Sigma, C)$.

[6] To the best of our knowledge only an exponential implementation exists in the literature [3].

4.6 A QBF representation of defeat

A defeat $\mathsf{defeat}((H,h),(H',h'))$ can be represented by

$$\xi(H,h,H',h') = CONJ(H) \wedge SEL(h)[\mathcal{P}_{L,C}] \wedge h[\mathcal{P}_{D,C}] \\ \wedge CONJ(H')[\mathcal{P}'_D] \wedge SEL(h')[\mathcal{P}'_{L,C}] \wedge h'[\mathcal{P}'_{D,C}]$$

by extending the language $\mathcal{L}[\mathcal{P}]$ to be $\mathcal{L}[\mathcal{P}] \cup \mathcal{L}[\mathcal{P}']$. A defeat relation $D = \{(A_i, A'_i)\}$ can be represented by a single QBF/BDD formula:

$$\xi(D) = \bigvee_{(A_i,A'_i)\in D} [\xi(A_i) \wedge \xi(A'_i)].$$

Now we need an expression with a polynomial number of operations to generate the set of all possible defeats from Σ. To do this, we need to inspect the specific types of defeats. We start with undercut:

Definition 8. *An argument (H_1, h_1)* undercuts *another argument (H_2, h_2) iff there exists an $f \in H_2$ such that $h_1 \equiv \neg f$.*

and this gives us:

Proposition 4. *Let $C = \Sigma \cup \{\neg f_i | f_i \in \Sigma\}$, the set of all possible undercuts can be constructed as*

$$\mathsf{undercut}(\Sigma) = Args(\Sigma, C) \wedge Args(\Sigma', C') \wedge (\bigvee_{f'_i \in \Sigma'} (c_{\neg f_i} \wedge l_i))$$

where $c_{\neg f_i}$ denotes the encoding of the label that corresponds to $\neg f_i$ in C.

Proof. $Args(\Sigma, C)$ and $Args(\Sigma', C')$ constructs the arguments for C based on Σ using two sets of symbols, and the corresponding selection of input sentences and conclusion sentences. $(\bigvee_{f'_i \in \Sigma'} (c_{\neg f_i} \wedge l_i))$ builds up the undercut relation between these two sets of arguments. □

Note that the setting of the conclusion points $C = \Sigma \cup \{\neg f_i | f_i \in \Sigma\}$ can be changed according to any application-dependent argumentation process, for example $CONS(\Sigma)$ and their negations or other application oriented conclusions and their negations.

Next we consider rebut:

Definition 9. (H_1, h_1) rebuts (H_2, h_2) *iff* $h_1 \equiv \neg h_2$.

We can construct the rebut relation in the same way as undercut by assuming a set of interesting conclusion points. However, we can also construct the rebut relation in the following way and leaving the conclusion points open to make the system more flexible.

Definition 10. *Given a set H of sentences, let*

$$S(H) = \{s | s \models H\} \\ S(h) = \{s | s \models h\}$$

where s is an assignment to $\mathcal{P}$. $H \vdash h$ iff $S(H) \subseteq S(h)$.

The definition of rebut is then:

Definition 11. *H_1 rebuts H_2, if there is some h such that $H_1 \vdash h$ and $H_2 \vdash \neg h$.*

and we have:

Proposition 5. *Given two consistent sets of sentences H_1 and H_2, H_1 rebuts H_2 iff $S(H_1) \cap S(H_2) = \emptyset$, namely $[CONJ(H_1) \wedge CONJ(H_2)] \leftrightarrow FALSE$.*

Proof. If H_1 rebuts H_2, then there is a h such that $H_1 \vdash h$ and $H_2 \vdash \neg h$. Since $S(h) \cap S(\neg h) = \emptyset$, and $S(H_1) \subseteq S(h)$ and $S(H_2) \subseteq S(\neg h)$, therefore $S(H_1) \cap S(H_2) = \emptyset$.

If $S(H_1) \cap S(H_2) = \emptyset$, the rebutting point h can be constructed as follows. Let $padding = \neg(H_1 \vee H_2)$, and $h = H_1 \vee padding$. In this way, $S(padding) = \mathcal{U} \setminus (S(H_1) \cup S(H_2))$, $S(h) = S(H_1) \cup S(padding)$, $S(\neg h) = \mathcal{U} \setminus (S(H_1) \cup S(padding)) = S(H_2)$. Therefore $S(H_1) \subseteq s(h)$ and $S(H_2) \subseteq S(\neg h)$, namely h is the rebutting point we are looking for such that $H_1 \vdash h$ and $H_2 \vdash \neg h$. $\square$

Actually h can be anything such that $S(H_1) \subseteq S(h) \subseteq (S(H_1) \cup S(padding))$, so we have the following corollary.

Corollary 1. *Given two sets of sentences H_1 and H_2 which rebut each other, the rebut point h can be obtained by setting $S(H_1) \subseteq S(h) \subseteq S(H_1) \cup S(padding)$ where $padding = \neg(H_1 \vee H_2)$. The choice of $h = H_1 \vee \neg(H_1 \vee H_2)$ which makes H_1 and H_2 be the minimal sets of sentences such that $H_1 \vdash h$ and $H_2 \vdash \neg h$* $\square$

As a result, the set of all rebuts can be expressed as

$$\mathsf{rebut}(\Sigma) = \bigvee_{\sigma \subseteq \Sigma, \sigma' \subseteq \Sigma} [CONJ(\sigma) \wedge CONJ'(\sigma') \wedge \neg\,(CONJ(\sigma)_D \wedge CONJ(\sigma')_D)]$$

and we have:

Proposition 6. $\mathsf{rebut}(\Sigma)$ *can be expressed as*

$$\mathsf{rebut}(\Sigma) = CONS(\Sigma) \wedge CONS'(\Sigma) \wedge \left[\bigvee_{f_i \in \Sigma}(l_i \wedge \neg f_i) \vee \bigvee_{f_j \in \Sigma}(l'_j \wedge \neg f_j)\right]$$

using $2 \times O(CONS(\Sigma)) + 6 \times |\Sigma| + 3$ QBF/BDD operations.

Proof.

$$\begin{aligned}
&\mathsf{rebut}(\Sigma)\\
&= \bigvee_{\sigma \subseteq \Sigma, \sigma' \subseteq \Sigma} [CONJ(\sigma) \wedge CONJ'(\sigma') \wedge \neg\,(CONJ(\sigma)_D \wedge CONJ(\sigma')_D)]\\
&= \bigvee_{\sigma \subseteq \Sigma, \sigma' \subseteq \Sigma} [CONJ(\sigma) \wedge CONJ'(\sigma') \wedge (\neg CONJ(\sigma)_D \vee \neg CONJ(\sigma')_D)]\\
&= \bigvee_{\sigma \subseteq \Sigma, \sigma' \subseteq \Sigma} \left[CONJ(\sigma) \wedge CONJ'(\sigma') \wedge \left(\bigvee_{f_i \in \sigma} \neg f_i \vee \bigvee_{f_j \in \sigma'} \neg f_j\right)\right]
\end{aligned}$$

$$= \bigvee_{\sigma \subseteq \Sigma, \sigma' \subseteq \Sigma} \left[CONJ(\sigma) \wedge CONJ'(\sigma') \wedge \left(\bigvee_{f_i \in \sigma} (l_i \wedge \neg f_i) \vee \bigvee_{f_j \in \sigma'} (l'_j \wedge \neg f_j) \right) \right]$$

$$= CONS(\Sigma) \wedge CONS'(\Sigma) \wedge \left[\bigvee_{f_i \in \Sigma} (l_i \wedge \neg f_i) \vee \bigvee_{f_j \in \Sigma} (l'_j \wedge \neg f_j) \right]$$

□

Finally we consider the computation of the contradict relation:

Definition 12. (H_1, h_1) *contradicts* (H_2, h_2) *if and only if* (H_1, h_1) *rebuts a subargument of* (H_2, h_2).

The contradict relation can be computed by

$$\mathsf{contradict}(\Sigma) = \exists_{\mathcal{Z}} \left(\mathsf{rebut}(\Sigma)[\mathcal{P}'/\mathcal{Z}] \wedge \xi(\subseteq)[\mathcal{P}/\mathcal{Z}] \right)$$

4.7 Computing fixed points of argumentation

The relations undercut, rebut and contradict give us the relationship between individual arguments, but, as is usual, we are more interested in computing things like which arguments are *acceptable*, where such properties are defined as fixed-points.

Definition 13. *An argument H defends another argument H' if there exists another argument H'' such that H'' defeats H' but H defeats H''.*

The defend relation can be constructed from the defeat relation on the set of arguments as follows:

$$\mathsf{defend}(\Sigma, \mathsf{defeat}) = \exists_{\mathcal{Z}} \left(\mathsf{defeat}(\Sigma)[\mathcal{P}'/\mathcal{Z}] \wedge \mathsf{defeat}(\Sigma)[\mathcal{P}/\mathcal{Z}] \right)$$

where $\mathsf{defeat}(\Sigma)$ is either $\mathsf{undercut}(\Sigma)$, $\mathsf{rebut}(\Sigma)$, $\mathsf{contradict}(\Sigma)$, or any disjunction of the relations (e.g. $\mathsf{undercut}(\Sigma) \vee \mathsf{rebut}(\Sigma)$). The composition of two relations R_1 and R_2 on the set $\mathcal{A}$ of arguments can be computed by

$$ComposeR(R_1, R_2) = \exists_{\mathcal{Z}} R_1[\mathcal{P}'/\mathcal{Z}] \wedge R_2[\mathcal{P}/\mathcal{Z}].$$

With these constructs defined, the fixed point of argumentation can be computed using Algorithm 4.2. In Algorithm 4.2, the closure of a binary relation R on $\mathcal{A}$, is computed using a method called iterative squaring [7] which is guaranteed to terminate within $O(log|\mathcal{A}|)$ steps. In line 3 : $OldR \leftarrow I_{\mathcal{P}_L} \cup \mathsf{defend}_{\mathcal{P}_L}$, the defend relation is first projected to sentence labeling symbols so that during the computation of the defending closure only the label of arguments are considered without referring to their internal structure; the union with the identity relation $I_{\mathcal{P}_L} = \bigwedge_{f_i \in \Sigma} (l_i \leftrightarrow l'_i)$ is to keep the defended arguments in the closure.

Proposition 7. *Algorithm 4.2 computes the fixed point of the defend relation, namely the set of acceptable arguments constructed from Σ.*

Algorithm 4.2 Computing Fixed Point of Argumentation

```
 1: function $ComputeFixedpoint(\Sigma, \mathsf{defeat})$ {
    (1) $\Sigma$: The set of input information
    (2) defeat is binary relation on $\mathcal{A}$ }
 2: $\mathsf{defend} \leftarrow \mathsf{defend}(\Sigma, \mathsf{defeat})$
 3: $OldR \leftarrow I_{\mathcal{P}_L} \cup \mathsf{defend}_{\mathcal{P}_L}$
 4: $R \leftarrow FAIL$
 5: while $(OldR \neq R)$ do
 6:     $tmpR \leftarrow R$
 7:     $R \leftarrow ComposeR(OldR, OldR)$
 8:     $OldR \leftarrow tmpR$
 9: end while
10: $Undefeated \leftarrow CONS(\Sigma) \wedge \neg\, (\exists_{\boldsymbol{x} \in \mathcal{P}} \mathsf{defeat})\, [\mathcal{P}'/\mathcal{P}]$
11: $Acc \leftarrow \exists_{\boldsymbol{x} \in \mathcal{P}}(Undefeated \wedge R)[\mathcal{P}'/\mathcal{P}] \vee Undefeated$
12: return $Acc$ end function
```

Proof. Let $step(R)$ be the maximum length of paths between a pair $(A, A') \in R$ in the induced graph of the defend relation defend. Let the starting R in line 3 denoted by $R_0 = \mathsf{defend} \cup I$. In R_0, for every $(A, A') \in R$, either $(A, A') \in \mathsf{defend}$, namely A defends A' using one step, or A is identical to A' namely A defends A' using 0 step, therefore $step(R_0) = 1$. Let the consequent content of R in each $while$ iteration denoted by R_i where i is the number of the iteration. Each time, when $R_{i+1} \leftarrow ComposeR(R_i, R_i)$ is applied in line 7, R_{i+1} will gather all the argument pairs of the form (A, A') such that A defends A' using defending steps less or equal than $step(R_{i+1}) = 2 \times step(R_i)$ steps. Assume that i is the number such that $R_{i+1} = R_i$, if the iteration continues we will have

$$R_{i+2} = ComposeR(R_{i+1}, R_{i+1}) = ComposeR(R_i, R_i) = R_{i+1} = R_i$$

namely for all $j \geq i$, $R_j = R_i$. Theforefore, after the $while$ loop terminates R will gather all the argument pairs (A, A') via any number of defending steps. Since the number of arguments is finite, all the defending paths are of finite length, therefore the algorithm is guaranteed to terminate. □

Proposition 8. *The complexity of algorithm 4.2 is $O(|\Sigma| \times K^2 \times |\mathcal{P}|)$ where K is the maximum size of the BDDs which appear during the fixed point computing process.*

Proof. As the analyzed in the proof of proposition 7, the $step(R_i) = step^2(R_{i+1})$. The maximum possible step of R_is is the number of arguments which is $2^{|\Sigma|}$. Therefore the algorithm is guaranteed to terminate after $m = log_2 2^{|\Sigma|} = |\Sigma|$ iterations, therefore the number of iteraction is bounded above by $O(|\Sigma|)$. In each iteration, $CompoeseR$ can be computed using $O(1 + |\mathcal{P}|)$ number of BDD operations, each operation is of complexity $O(K^2)$ where K is the maximum size of BDDs used. Therefore the whole algorithm is bounded above by $O(|\Sigma|) \times O(K^2 \times |\mathcal{P}|)$. □

5 Discussion

Proposition 8 shows that we can compute the fixed-point in a polynomial number of BDD operations. As we mentioned above, this is a long way from saying that we can do general logical inference in polynomial time, rather what we are saying is that while the complexity of algorithm 4.2 depends on the maximum size of the BDD (K), this doesn't depend on the size of Σ but rather on the complexity of the information contained in Σ. In the worse case, K can still be exponential in $|\mathcal{P}|$, but in many practical applications K tends to be small.

Because of this feature of systems built using the QBF/BDD representation, there has been a lot of work on reducing the size of BDDs. Many successful approaches have been developed in literature, especially those developed for symbolic model checking in software and hardware verfication [24], and in non-determinstic AI planning [10]. Examples of techniques for reducing the size of BDDs are early quantification [19], quantification scheduling [9], transition partitioning [6], iterative squaring [7, 8], frontier simplification [12], input splitting [28, 29], and state set A^* branching [22, 21, 23] (a BDD version of the A^* search heuristic [31]).

Another factor affecting the BDD size greatly is variable ordering. The problem of finding an optimal variable ordering is NP-complete [4]. Algorithms based on dynamic programming [14], heuristics [20], dynamic variable reordering [30] and machine learning approaches [18] have been proposed for finding a good variable ordering in reasonable time[7].

We are currently working on an implementation of the reasoning mechanism proposed above with the aim of experimentally clarifying the nature of K for different argumentation problems.

6 Conclusions and Future Work

In this paper, we have proposed a symbolic model checking approach to compute argumentation. The computation only uses a polynomial number of BDD operations in terms of the number of sentences in the input and the number of symbols used in the input. A key idea in the approach is to construct the set of consistent arguments all together using a polynomial number of BDD operations. In the same way, the defeat relation among these arguments can also be computed all at once using a polynomial number of BDD operations. And with the iterative squaring technique, we are able to compute the fixed point of a set of arguments in polynomial number of BDD operations.

We are currently working on implementing the BDD-based argumentation system proposed in this paper, with the aim of conducting experiments to classify the nature of the BDDs constructed for argumentation. This will allow us to determine how effective this approach will be in general. This in turn may lead us to look for new heuristics for controlling the size of the BDDs we need to construct to compute arguments. Another direction that we are working on is to extend the current method to compute more

[7] [18] is also a good source for other references on BDD variable (re-)ordering.

sophisticated and controllable approaches argumentation, such as those based on argumentation schemes [27]. On the way, we will need to develop BDD techniques to efficiently specify application-dependent patterns of arguments (such as those captured by argument schemes), specify application-dependent patterns of defeats (defeat schemes), and extend the basic entailment-based reasoning modelled here to specify the necessary patterns of rule-based procedural reasoning. In combination with our continuing efforts to use BDD techniqes in multiagent planning and dialogues [34, 36, 38, 39], all these efforts are aimed at our ultimate goal of a practical argumentation-based dialogue model for multiagent planning.

Acknowledgment

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29(2):125–169, 2002.
2. Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.
3. Rudolf Berghammer and Alexander Fronk. Exact computation of minimum feedback vertex sets with relational algebra. *Fundam. Inf.*, 70(4):301–316, 2005.
4. Beate Bollig and Ingo Wegener. Improving the variable ordering of OBDDs is NP-Complete. *IEEE Transations on Computers*, 45(9):993–1002, 1996.
5. Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.
6. J. R. Burch, E. M. Clarke, and D. E. Long. Symbolic model checking with partitioned transition relations. In *Proceedings of International Conference on Very Large Scale Integration*, pages 49–58. North-Holland, 1991.
7. J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic Model Checking: 10^{20} States and Beyond. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 1–33, Washington, D.C., 1990. IEEE Computer Society Press.
8. Gianpiero Cabodi, Paolo Camurati, Luciano Lavagno, and Stefano Quer. Disjunctive partitioning and partial iterative squaring: an effective approach for symbolic traversal of large circuits. In *DAC '97: Proceedings of the 34th Annual Conference on Design Automation*, pages 728–733, New York, NY, USA, 1997. ACM.

9. Pankaj Chauhan, Edmund M. Clarke, Somesh Jha, Jim Kukula, Tom Shiple, Helmut Veith, and Dong Wang. Non-linear quantification scheduling in image computation. In *ICCAD '01: Proceedings of the 2001 IEEE/ACM International Conference on Computer-aided Design*, pages 293–298, Piscataway, NJ, USA, 2001. IEEE Press.
10. A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.
11. O. Coudert and J. C. Madre. The implicit set paradigm: a new approach to finite state system verification. *Formal Methods in System Design*, 6(2):133–145, 1995.
12. Olivier Coudert, Christian Berthet, and Jean Christophe Madre. Verification of synchronous sequential machines based on symbolic execution. In *Automatic Verification Methods for Finite State Systems*, pages 365–373, 1989.
13. Yannis Dimopoulos, Bernhard Nebel, and Francesca Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141:57–78, 2002.
14. Rolf Drechsler, Nicole Drechsler, and Wolfgang Günther. Fast exact minimization of BDDs. In *DAC '98: Proceedings of the 35th Annual Conference on Design Automation*, pages 200–205, New York, NY, USA, 1998. ACM.
15. 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):321–358, 1995.
16. Paul E. Dunne and T. J. M. Bench-capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149:2003, 2001.
17. Thomas Gordon and Nikos Karacapilidis. The zeno argumentation framework. In *in Proceedings of the Sixth International Conference on AI and Law*, pages 10–18. ACM Press, 1997.
18. Orna Grumberg, Shlomi Livne, and Shaul Markovitch. Learning to order BDD variables in verification. *Journal of Artificial Intelligence Research (JAIR)*, 18:83–116, 2003.
19. Ramin Hojati, Sriram C. Krishnan, and Robert K. Brayton. Early quantification and partitioned transition relations. In *ICCD '96: Proceedings of the 1996 International Conference on Computer Design, VLSI in Computers and Processors*, pages 12–19, Washington, DC, USA, 1996. IEEE Computer Society.
20. Jawahar Jain, William Adams, and Masahiro Fujita. Sampling schemes for computing OBDD variable orderings. In *ICCAD '98: Proceedings of the 1998 IEEE/ACM International Conference on Computer-aided Design*, pages 631–638, New York, NY, USA, 1998. ACM.
21. Rune M. Jensen, Randal E. Bryant, and Manuela M. Veloso. An efficient BDD-based A* algorithm. In *Proceedings of AIPS-02 Workshop on Planning via Model Checking*, 2002.
22. Rune M. Jensen, Randal E. Bryant, and Manuela M. Veloso. SetA*: An efficient BDD-based heuristic search algorithm. In *Proceedings of 18th National Conference on Artificial Intelligence (AAAIŠ02)*, pages 668–673, 2002.
23. Rune M. Jensen, Manuela M. Veloso, and Randal E. Bryant. State-set branching: Leveraging BDDs for heuristic search. *Artificial Intelligence*, 172(2-3):103–139, 2008.
24. Ranjit Jhala and Rupak Majumdar. Software model checking. *ACM Comput. Surv.*, 41(4):1–54, 2009.
25. Antonios C. Kakas and Francesca Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9:515–562, 1999.
26. Nikos Karacapilidis and Dimitris Papadias. Computer supported argumentation and collaborative decision making: The hermes system. *Information Systems*, 26:259–277, 2001.
27. Joel Katzav and Chris Reed. On argumentation schemes and the natural classification of arguments. *Argumentation*, 18(2), 2004.

28. Christoph Meinel and Thorsten Theobald. *Algorithms and Data Structures in VLSI Design*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.
29. In-Ho Moon, James H. Kukula, Kavita Ravi, and Fabio Somenzi. To split or to conjoin: the question in image computation. In *DAC '00: Proceedings of the 37th conference on Design automation*, pages 23–28, New York, NY, USA, 2000. ACM.
30. Shipra Panda, Fabio Somenzi, and Bernard F. Plessier. Symmetry detection and dynamic variable ordering of decision diagrams. In *ICCAD '94: Proceedings of the 1994 IEEE/ACM International Conference on Computer-aided Design*, pages 628–631, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
31. Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.
32. Henry Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.
33. Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, December 2003.
34. Yuqing Tang, Timothy J. Norman, and Simon Parsons. A model for integrating dialogue and the execution of joint plans. In *Proceedings of the Eigth International Joint Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, May 10-15 2009.
35. Yuqing Tang, Timothy J. Norman, and Simon Parsons. Towards the implementation of a system for planning team activities. In *Proceedings of the Second Annual Conference of the ITA*, University of Maryland University College, Maryland, 2009.
36. Yuqing Tang and Simon Parsons. Argumentation-based dialogues for deliberation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 552–559, New York, NY, USA, 2005. ACM Press.
37. Yuqing Tang and Simon Parsons. Using argumentation-based dialogues for distributed plan management. In *Proceedings of the AAAI Spring Symposium on Distributed Plan and Schedule Management*, Stanford, 2006.
38. Yuqing Tang and Simon Parsons. A dialogue mechanism for public argumentation using conversation policies. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 445–452, Estoril, Portugal, May 12-16 2008.
39. Yuqing Tang and Simon Parsons. An MDP model for planning team actions with communication. Technical report, International Technology Alliance in Network and Information Science, 2009.

Preference-based Argumentation Capturing Prioritized Logic Programming

Toshiko Wakaki

Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama, 337–8570 Japan
twakaki@sic.shibaura-it.ac.jp

Abstract. Firstly we present a novel approach of an abstract preference-based argumentation framework (an abstract PAF, for short), which generalizes a Dung's abstract argumentation framework (AF) to deal with additional preferences over the set of arguments. In our formalism, the semantics of such a PAF is given as $\mathcal{P}$-extensions that are selected from extensions of acceptability semantics by taking account for such preferences. Secondly, using a prioritized logic program (PLP) capable of representing priority information along with integrity constraints, the proposed method defines the non-abstract preference-based argumentation framework (a non-abstract PAF) translated from a PLP, whose semantics is also given by $\mathcal{P}$-extensions instantiating those of an abstract one. Finally we show the interesting result that, $\mathcal{P}$-extensions of such a non-abstract PAF under stable semantics capture *preferred* answer sets of a PLP, which ensures advantages as well as correctness of our approach.

Categories and Subject Descriptors
1.2.11[Distributed Artificial Intelligence]:[Multiagent systems]
General Terms Theory, Languages
Keywords preferences, argumentation, prioritized logic programs

1 Introduction

In the research field of argumentation, Dung's frameworks of abstract argumentation [9] have gained wide acceptance and are the basis for the implementation of concrete formalisms. In his paper [9], Dung showed that argumentation can be viewed as a special form of logic programming with negation as failure and gives a series of theorems that relate semantics of logic programs and semantics of argumentation frameworks. As practical applications, there have been a number of proposals for negotiation between multiagents that make use of argumentation.

Recently, several approaches to generalize Dung's theory have been proposed in order to handle additional information such as preferences as well as constraints which a negotiating agent generally has as its knowledge, because preferences are useful to solve conflicts between arguments and constraints are needed to eliminate extensions not satisfying the required conditions. With respect to handling preferences, quite recently, Amgoud and Vesic [2] pointed out that, there is the critical problem such that extensions are not conflict-free w.r.t. the attack relation for existing preference-based argumentation frameworks such as Amgoud and Cayrol's approach [1]. The property of conflict-freeness for extensions is important since it ensures sound results. Hence they [2] proposed a new abstract preference-based argumentation framework whose semantics ensures requirements of *conflict-freeness* along with *generalization* to recover Dung's acceptability semantics in case that preferences are not available.

On the other hand, with respect to formalisms for integrating logic programming and argumentation, Dung [8] showed that answer set semantics [11] of an extended logic program (an ELP, for short) is captured by stable semantics of Dung's argumentation framework, whereas Prakken and Sartor [13] introduced an argument-based formalism for extended logic programming with defeasible properties, which instantiated Dung's grounded semantics if it is restricted to static priorities. Especially with respect to logic programming based on answer set semantics [10, 11], a significant amount of studies have been done such as Brewka and Eiter's preferred answer sets for extended logic programs [4], Sakama and Inoue's prioritized logic programming [17], Delgrande and Schaub's ordered logic programs [7] and so on. However, few works have been achieved with respect to the semantical relation between such logic programming capable of handling preferences and preference-based argumentation which generalizes Dung's argumentation framework [9] as far as we know.

Under such circumstances, firstly, we present a new approach of an abstract preference-based argumentation framework (an abstract PAF, for short), which generalizes Dung's abstract argumentation framework to deal with additional preferences with meeting requirements of conflict-freeness and generalization. In our formalism, the semantics of such a PAF is given as $\mathcal{P}$-extensions that are selected from extensions of Dung's acceptability semantics by taking into account such preferences.

Secondly, since Sakama and Inoue's formalism of a prioritized logic program (PLP) is capable of representing priority information along with integrity constraints in a nonmonotonic logic program, we use such PLP as the underlying logic to construct an non-abstract PAF instantiating an abstract PAF. That is, the proposed method defines a non-abstract preference-based argumentation framework (a non-abstract PAF, for short) translated from a PLP, whose semantics is also given by $\mathcal{P}$-extensions instantiating those of an abstract one. As a result, we can show the interesting result that, $\mathcal{P}$-extensions of such a non-abstract PAF under stable semantics capture *preferred* answer sets of a PLP, which generalizes Dung's theorem about relation between answer sets of an ELP P and stable extensions of the argumentation framework associated with P. Thus this property ensures advantages as well as correctness of our approach.

Finally under an inconsistent knowledge base, the PLP system [19, 18] can reason nothing from it, whereas the non-abstract PAF translated from a PLP enables to reason intended results from it based on preferred semantics, i.e. preferred $\mathcal{P}$-extensions. Therefore we can regard such a non-abstract PAF as the enhanced PLP so that it can also reason paraconsistently from inconsistent knowledges.

This paper is organized as follows. Section 2 gives preliminaries. Section 3 presents a new abstract PAF. Section 4 presents the non-abstract PAF translated from a PLP and the semantics. Section 5 discusses the related work and Section 6 concludes the paper.

2 Preliminaries

We briefly review the basic notions used throughout this paper.

2.1 Extended Logic Programs and Answer Set Semantics

The logic programs we consider in this paper are extended logic programs (ELPs), which have two kinds of negation, i.e. classical negation ($\neg$) along with negation as failure (*not*) defined as follows.

Definition 1 *An extended logic program (ELP)[11, 10] is a set of rules of the form:*

$$L \leftarrow L_1, \ldots, L_m, notL_{m+1}, \ldots, notL_n, \qquad (1)$$

or of the form:

$$\leftarrow L_1, \ldots, L_m, notL_{m+1}, \ldots, notL_n, \qquad (2)$$

where L and L_i's are literals, i.e. either atoms or atoms preceded by the classical negation sign $\neg$ and $n \geq m \geq 0$. The symbol "not" denotes negation as failure. We call a literal preceded by "not" a NAF-literal. For a rule r of the form (1), we call L the head of the rule, head(r), and $\{L_1, \ldots, L_m, notL_{m+1}, \ldots, notL_n\}$ the body of the rule, body(r). Especially, $body(r)^+$ and $body(r)^-$ denote $\{L_1, \ldots, L_m\}$ and $\{L_{m+1}, \ldots, L_n\}$ respectively. We often write $L \leftarrow body(r)^+, not\ body(r)^-$ instead of (1) by using sets, $body(r)^+$ and $body(r)^-$. Each rule of the form (2) is called an integrity constraint. For a rule with an empty body, we may write L instead of $L \leftarrow$. As usual, a rule with variables stands for the set of its ground instances.

The semantics of an ELP is given by the *answer sets* [11, 10] as follows.

Definition 2 *Let Lit_P be the set of all ground literals in the language of P. First, let P be a not-free ELP (i.e., for each rule $m = n$). Then, $S \subseteq Lit_P$ is an answer set of P if S is a minimal set satisfying the conditions:*

1. *For each ground instance of a rule $L \leftarrow L_1, \ldots, L_m$ in P, if $\{L_1, \ldots, L_m\} \subseteq S$, then $L \in S$. In particular, for each integrity constraint $\leftarrow L_1, \ldots, L_m$ in P, $\{L_1, \ldots, L_m\} \not\subseteq S$ holds;*
2. *If S contains a pair of complementary literals, then $S = Lit_P$.*

Second, let P be any ELP and $S \subseteq Lit_P$. The reduct of P by S is a not-free ELP P^S whose form is $L \leftarrow L_1, \ldots, L_m$, or $\leftarrow L_1, \ldots, L_m$, iff there is a ground rule of the form (1), (2) in P s.t. $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then, S is an answer set of P if S is an answer set of P^S.

An answer set is *consistent* if it is not Lit_P. A program P is *consistent* if it has a consistent answer set; otherwise, P is *inconsistent*. We write $P \models L$ if a literal L is included in every answer set of P.

2.2 Prioritized Logic Programs and Preferred Answer Sets

A prioritized logic program (PLP) [17] is defined as follows.

Definition 3 (**Priorities**) *Given an ELP P and the set of ground literals Lit_P, a reflexive and transitive relation $\preceq$ is defined on Lit_P. For any element e_1 and e_2 from Lit_P, $e_1 \preceq e_2$ is called a priority, and we say e_2 has a higher priority than e_1. We write $e_1 \prec e_2$ if $e_1 \preceq e_2$ and $e_2 \not\preceq e_1$, and say e_2 has a strictly higher priority than e_1.*

Definition 4 (**Prioritized Logic Programs, PLPs**) *A prioritized logic program (PLP, for short) is defined as a pair (P, Φ), where P is an ELP* [1] *and Φ is a set of priorities on Lit_P.*

The declarative semantics of a PLP (P, Φ) is given by *preferred answer sets* which are selected from answer sets of P based on the preference relation $\sqsubseteq_{as}$ derived from priorities in Φ. In what follows, the closure Φ^* is defined as the set of priorities which are reflexively or transitively derived using priorities in Φ.

Definition 5 (**Preferences between answer sets**) *Given a PLP (P, Φ), the preference relation $\sqsubseteq_{as}$ over answer sets of P is defined as follows:*
For any answer sets S_1, S_2 and S_3 of P,

1. *$S_1 \sqsubseteq_{as} S_1$,*
2. *$S_1 \sqsubseteq_{as} S_2$ if for some literal $e_2 \in S_2 \setminus S_1$,*
   (i) there is a literal $e_1 \in S_1 \setminus S_2$ such that $e_1 \preceq e_2 \in \Phi^*$, and
   (ii) there is no literal $e_3 \in S_1 \setminus S_2$ such that $e_2 \prec e_3 \in \Phi^*$,
3. *if $S_1 \sqsubseteq_{as} S_2$ and $S_2 \sqsubseteq_{as} S_3$, then $S_1 \sqsubseteq_{as} S_3$.*

We say that S_2 is preferable to S_1 with respect to Φ if $S_1 \sqsubseteq_{as} S_2$ holds. We write $S_1 \sqsubset_{as} S_2$ if $S_1 \sqsubseteq_{as} S_2$ and $S_2 \not\sqsubseteq_{as} S_1$. Hereafter, each $S_1 \sqsubseteq_{as} S_2$ is called a preference between answer sets.

Definition 6 (**Preferred answer sets**) *Let (P, Φ) be a PLP. Then, an answer set S of P is called a preferred answer set (or p-answer set, for short) of (P, Φ) if $S \sqsubseteq_{as} S'$ implies $S' \sqsubseteq_{as} S$ (with respect to Φ) for any answer set S' of P.*

2.3 Abstract/non-Abstract Argumentation Frameworks and Acceptability Semantics

Dung presented an abstract argumentation framework and acceptability semantics [9] defined as follows.

Definition 7 (**Abstract Argumentation Frameworks**) *An abstract argumentation framework is a pair AF=$(\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is a set of arguments and $\mathcal{R}$ is a binary relation over $\mathcal{A}$, i.e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. $(a, b) \in \mathcal{R}$, or equivalently $a\ \mathcal{R}\ b$, means that a attacks b. A set S of arguments attacks an argument a if a is attacked by an argument of S.*

[1] In this paper, for a PLP (P, Φ), P is restrictedly given as an ELP though such P is originally allowed to be a GEDP, i.e. a member of the superclass of an ELP [17].

Definition 8 *(***Acceptable sets / Conflict-free sets***) Let AF=($\mathcal{A}$,$\mathcal{R}$) be an argumentation framework. A set $S \subseteq \mathcal{A}$ is conflict-free iff there are no arguments a and b in S such that a attacks b. An argument $a \in \mathcal{A}$ is acceptable w.r.t. a set $S \subseteq \mathcal{A}$ iff for any $b \in \mathcal{A}$ such that $(b, a) \in \mathcal{R}$, there exists $c \in S$ such that $(c, b) \in \mathcal{R}$.*

Definition 9 *(***Acceptability Semantics***) Let AF=($\mathcal{A}$,$\mathcal{R}$) be an argumentation framework and $E \subseteq \mathcal{A}$ be a conflict-free set of arguments. Let $F : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ be a function with $F(E) = \{a \mid a \text{ is acceptable w.r.t. } E\}$.*
Acceptability Semantics such as `complete` *(resp.* `stable`*,* `preferred`*,* `grounded`*) semantics is given by the respective extensions defined as follows. E is admissible iff $E \subseteq F(E)$. E is a complete extension iff $E = F(E)$. E is a grounded extension iff E is a minimal (w.r.t. set-inclusion) complete extension. E is a preferred extension iff E is a maximal (w.r.t. set-inclusion) complete extension. E is a stable extension iff E is a preferred extension that attacks every argument in $A \setminus E$.*

Definition 10 *(***Credulous Justification vs Skeptical Justification***)*
Let AF=($\mathcal{A}$, $\mathcal{R}$) be an argumentation framework and Sname be one of `complete`*,* `stable`*,* `preferred`*, and* `grounded`*. Then for an argument $a \in \mathcal{A}$,*

- *a is credulously justified (w.r.t. ($\mathcal{A}$, $\mathcal{R}$)) under Sname semantics iff a is contained in at least one Sname extension of ($\mathcal{A}$, $\mathcal{R}$);*
- *a is skeptically justified (w.r.t. ($\mathcal{A}$, $\mathcal{R}$)) under Sname semantics iff a is contained in every Sname extension of ($\mathcal{A}$, $\mathcal{R}$).*

Non-abstract argumentation formalisms for ELPs [13, 15] are defined as follows.

Definition 11 (Arguments) *[15] Let P be an extended logic program whose rules have the form (1). An argument associated with P is a finite sequence $Ag = [r_1, \ldots, r_n]$ of ground instances of rules $r_i \in P$ such that for every $1 \leq i \leq n$, for every literal L_j in the body of r_i there is a $k > i$ such that $head(r_k) = L_j$. The head of a rule in Ag, i.e. $head(r_i)$ is called a conclusion of Ag, whereas a NAF-literal not L in the body of a rule of Ag is called an assumption of Ag. We write $assm(Ag)$ for the set of assumptions and $conc(Ag)$ for the set of conclusions of an argument Ag. Especially we call the head of the first rule r_1 the claim of an argument Ag as written $claim(Ag)$.*

A subargument of Ag is a subsequence of Ag which is an argument. An argument Ag with a conclusion L is a minimal argument for L if there is no subargument of Ag with conclusion L. An argument Ag is minimal if it is minimal for its claim, i.e. $claim(Ag)$. Given an extended logic program P, the set of minimal arguments associated with P is denoted by $Args_P$.

As usual, the notions of attack such as "rebut", "undercut", "attack", "defeat" abbreviated to `r`, `u`, `a`, `d` are defined as a binary relation over $Args_P$ as follows.

Definition 12 (Rebut, Undercut, Attack, Defeat) *For two arguments, Ag_1 and Ag_2, the notions of attack such as rebut, undercut, attack, defeat (*`r`*,* `u`*,* `a`*,* `d` *for short) are defined as follows:*

– Ag_1 *rebuts* Ag_2*, i.e.* $(Ag_1, Ag_2) \in \mathtt{r}$ *if there exists a literal* L *such that* $L \in conc(Ag_1)$ *and* $\neg L \in conc(Ag_2)$*;*
– Ag_1 *undercuts* Ag_2*, i.e.* $(Ag_1, Ag_2) \in \mathtt{u}$ *if there exists a literal* L *such that* $L \in conc(Ag_1)$ *and not* $L \in assm(Ag_2)$*;*
– Ag_1 *attacks* Ag_2*, i.e.* $(Ag_1, Ag_2) \in \mathtt{a}$ *if* Ag_1 *rebuts or undercuts* Ag_2*;*
– Ag_1 *defeats* Ag_2*, i.e.* $(Ag_1, Ag_2) \in \mathtt{d}$ *if* Ag_1 *undercuts* Ag_2*, or* Ag_1 *rebuts* Ag_2 *and* Ag_2 *does not undercut* Ag_1*.*

Definition 13 (Abstract vs non-Abstract Argumentation Frameworks) *Let* P *be an ELP,* $Args_P$ *be the set of minimal arguments associated with* P *and* $attacks_P$ *be the binary relation over* $Args_P$ *defined according to some notion of attack (e.g.* $\mathtt{r}$, $\mathtt{u}$, $\mathtt{a}$, $\mathtt{d}$*). Then we call* $AF_P \overset{def}{=} (Args_P, attacks_P)$ *the "non-abstract argumentation framework" associated with* P*.*

Although Dung's acceptability semantics is defined as the set of extensions under the specified argumentation semantics w.r.t. an abstract argumentation framework $AF = (\mathcal{A}, \mathcal{R})$, it is also given as the set of extensions w.r.t. the non-abstract $AF_P = (Args_P, attacks_P)$ instantiating AF using an ELP P.

2.4 Answer Set Programming as Argumentation

Dung [8] showed that stable extensions of the argumentation framework AF_P associated with an ELP P without integrity constraints capture answer set semantics of P as follows.

Theorem 1. *Let* P *be an ELP having no integrity constraints, and* AF_P=($Args_P$, $attack_P$) *be the concrete argumentation framework associated with* P*, where* $attacks_P$ *is the binary relation over* $Args_P$ *defined according to undercut (i.e.* $\mathtt{u}$*) as the notion of attack. Then* S *is an answer set of* P *iff there is a stable extension* E *of* AF_P *such that*

$$S = \{\ L\ |L\ is\ a\ literal\ s.t.\ L = claim(Ag)\ for\ an\ argument\ Ag \in E\}.^2$$

3 A New Approach of an Abstract Preference-based Argumentation Framework

We present a new approach of an abstract preference-based argumentation framework (an abstract PAF for short), where Dung's acceptability semantics is extended in a natural way so as to take account for additional preferences.

An abstract PAF takes as input three elements: a set $\mathcal{A}$ of arguments, an attack relation $\mathcal{R}$ on $\mathcal{A}$, and a preorder $\leq$ on $\mathcal{A}$, where a pair AF=($\mathcal{A}$, $\mathcal{R}$) coincides a Dung's argumentation framework. It returns *extensions* that are subsets of $\mathcal{A}$ satisfying two basic requirements as addressed by Amgoud [2] as follows.
Conflict-freeness: If E is an extension (i.e. $\mathcal{P}$-extension in our approach) of

[2] In [8], it is expressed that $S = \{\ L\ |L\ is\ supported\ by\ an\ argument\ from\ E\}$.

PAF=($\mathcal{A}$, $\mathcal{R}$, $\leq$), then E is conflict free w.r.t. $\mathcal{R}$.

Generalization: Dung's acceptability semantics of AF=($\mathcal{A}$, $\mathcal{R}$) is captured as the special case of the semantics of PAF=($\mathcal{A}$, $\mathcal{R}$, $\leq$).

Our formalism of PAF which satisfies these basic requirements is based on the idea that an arguing agent wants to filter out extensions of the traditional acceptability semantics according to his/her preference, as is defined as follows.

Definition 14 *(***Priorities between arguments***)* A reflexive and transitive relation $\leq$ is defined over A. For any element a_1 and a_2 from $\mathcal{A}$, $a_1 \leq a_2$, or equivalently $(a_1, a_2) \in \leq$, is called a *priority*, and we say a_2 *has a higher priority than* a_1. We write $a_1 < a_2$ if $a_1 \leq a_2$ and $a_2 \not\leq a_1$, and say a_2 *has a strictly higher priority than* a_1.

Definition 15 *(***Preference-based Argumentation Frameworks***)*
A preference-based argumentation framework (PAF) *is a tuple* PAF*=(*$\mathcal{A}$*,*$\mathcal{R}$*,*$\leq$*), where* $\mathcal{A}$ *is a set of arguments,* $\mathcal{R}$ *is an attack relation on* $\mathcal{A}$*, and* $\leq$ *is a preorder on* $\mathcal{A}$*.*

The semantics of an abstract PAF=($\mathcal{A}$,$\mathcal{R}$, $\leq$) is given as *preferable extensions* (or $\mathcal{P}$-extensions) defined as follows. In what follows, given a set $\leq$ of priorities between arguments, we define the closure $\leq^*$ as the set of priorities which are reflexively and transitively derived from priorities in $\leq$.

Definition 16 *(***Preferences between extensions***)* *Let Sname be one of* `complete`*,* `stable`*,* `preferred`*, and* `grounded`*, i.e. names of Dung's acceptability semantics. Given* PAF*=(*$\mathcal{A}$*,* $\mathcal{R}$*,* $\leq$*), let* $\mathcal{E}$ *be the set of extensions for* $AF = (\mathcal{A}, \mathcal{R})$ *under Sname semantics. Then the preference relation* $\sqsubseteq_{ex}$ *is defined over* $\mathcal{E}$ *as follows. For any Sname extensions* E_1*,* E_2 *and* E_3 *from* $\mathcal{E}$*,*

1. $E_1 \sqsubseteq_{ex} E_1$*,*
2. $E_1 \sqsubseteq_{ex} E_2$ *if for some argument* $a_2 \in E_2 \setminus E_1$*,*
   (i) there is an argument $a_1 \in E_1 \setminus E_2$ *such that* $a_1 \leq a_2$ *w.r.t.* $\leq^*$*, and*
   (ii) there is no argument $a_3 \in E_1 \setminus E_2$ *such that* $a_2 < a_3$ *w.r.t.* $\leq^*$*,*
3. if $E_1 \sqsubseteq_{ex} E_2$ *and* $E_2 \sqsubseteq_{ex} E_3$*, then* $E_1 \sqsubseteq_{ex} E_3$*.*

Note that $\sqsubseteq_{ex}$ *is reflexive and transitive according to the items no.1 and no.3. We say that* E_2 *is preferable to* E_1 *with respect to* $\leq$ *if* $E_1 \sqsubseteq_{ex} E_2$ *holds. We write* $E_1 \sqsubset_{ex} E_2$ *if* $E_1 \sqsubseteq_{ex} E_2$ *and* $E_2 \not\sqsubseteq_{ex} E_1$*. Hereafter, each* $E_1 \sqsubseteq_{ex} E_2$ *is called a preference between extensions.*

Example 1. Consider PAF=$(\mathcal{A}, \mathcal{R}, \leq)$ where $\mathcal{A} = \{a, b, c, d\}$, $\mathcal{R} = \{(a, b), (b, a), (c, d), (d, c), (c, a), (b, d)\}$ and $\leq = \{(a, b), (a, c), (b, d), (c, d)\}$. Then both of $\{a, d\}$ and $\{b, c\}$ are preferred extensions as well as stable extensions of $AF = (\mathcal{A}, \mathcal{R})$, and $\{b, c\} \sqsubseteq_{ex} \{a, d\}$. Note that $\{a, d\} \not\sqsubseteq_{ex} \{b, c\}$ by the presence of $b \leq d$ and $c \leq d$ in $\leq$.

Definition 17 *(*$\mathcal{P}$**-extensions***)*
Let Sname be one of `complete`*,* `stable`*,* `preferred`*, and* `grounded`*, and* $\mathcal{E}$ *be the set of Sname extensions (e.g. a set of preferred extensions) for* $AF = (\mathcal{A}, \mathcal{R})$*. Given* $PAF = (\mathcal{A}, \mathcal{R}, \leq)$*, a Sname extension* $E \in \mathcal{E}$ *(e.g. a preferred extension) is called a Sname* $\mathcal{P}$*-extension (e.g. a preferred* $\mathcal{P}$*-extension) of* PAF *if* $E \sqsubseteq_{ex} E'$ *implies* $E' \sqsubseteq_{ex} E$ *(with respect to* $\leq$*) for any Sname extension* $E' \in \mathcal{E}$*.*

Example 2. (**Ex. 1, Cont.**) $\{a, d\}$ is a preferred $\mathcal{P}$-extension as well as a stable $\mathcal{P}$-extension of PAF w.r.t. $\leq$, but $\{b, c\}$ is neither of them.

Proposition 1 *(***Generalization***) Let Sname be one of* `complete`*,* `stable`*,* `preferred`*, and* `grounded`*.* E *is an Sname extension of* $AF=(\mathcal{A}, \mathcal{R})$ *iff* E *is a Sname* $\mathcal{P}$*-extension of* $PAF = (\mathcal{A}, \mathcal{R}, \leq)$ *when* $\leq$ *is empty.*

4 Preference-based and Constrained Argumentation capturing Prioritized LP

In Section 3, an abstract PAF is presented. In this section, a non-abstract preference-based argumentation framework (a non-abstract PAF, for short) compiled from a PLP expressing domain knowledges is proposed as follows.

In the following, let P (resp. IC) be a set of rules of the form (1) (resp. (2)). Then in answer set programming (ASP), the semantics of an ELP $P \cup IC$ is given by answer sets which are selected from answer sets of P by taking account of the set IC of integrity constraints. On the other hand, the semantics of a prioritized logic program, i.e. a PLP $(P \cup IC, \Phi)$, is given by *preferred answer sets* (*p-answer sets*, for short) which are selected from *answer sets* of $P \cup IC$ by taking account of the set Φ of priorities between literals, where integrity constraints from IC and priorities from Φ are regarded as hard and soft constraints respectively.

Similar idea is also applied to our formalisms of argumentation handling preferences and constraints. That is, our basic idea is that, given an ELP $P \cup IC$ as the underlying logic, the semantics of the constrained argumentation framework, i.e. $CAF = (Args_P, attacks_P, IC)$ is given by $\mathcal{C}$*-extensions* which are selected from *extensions* of the non-abstract argumentation framework $AF_P = (Args_P, attacks_P)$ under a particular Dung's argumentation semantics by taking account of constraints expressed by integrity constraints from IC, whereas given a PLP $(P \cup IC, \Phi)$ as the underlying logic, the semantics of the non-abstract preference-based argumentation framework, i.e. $PAF = (Args_P, attacks_P, IC, \leq)$ translated from a PLP is given by $\mathcal{P}$*-extensions* which are selected from $\mathcal{C}$*-extensions* of the CAF by taking account of the set $\leq$ of priorities between arguments as are constructed via priorities between literals from Φ.

4.1 Constrained AFs built on ELPs with integrity constraints

First of all, we show a constrained argumentation framework whose underlying logic is an ELP with integrity constraints as follows.

Definition 18 *(***From ELPs with constraints to constrained AFs***)*
A constrained argumentation framework $CAF(P, IC)$ associated with an ELP $P \cup IC$ is defined as follows:

$$CAF(P, IC) \stackrel{def}{=} (Args_P, attacks_P, IC),$$

where P and IC are sets of rules of the form (1) and (2) respectively.

After defining the claims of a set of arguments, we show the definition of satisfiability of an extension w.r.t. constraints as follows.

Definition 19 *(***The claims of a set of arguments***) Let E be a set of arguments. Then $claims(E)$ which we call the claims of E is defined as follows:*

$$claims(E) \stackrel{def}{=} \{L \mid L \; is \; a \; literal \; s.t. \; L = claim(Ag) \; for \; an \; argument \; Ag \in E\}.$$

Definition 20 *(***Satisfiability***) Let $CAF(P, IC) \stackrel{def}{=} (Args_P, attacks_P, IC)$ be a constrained argumentation framework associated with $P \cup IC$. Note that for a rule from IC having the following form (2) whose name is r_{ic}:*

$$r_{ic}: \qquad \leftarrow L_1, \ldots, L_m, notL_{m+1}, \ldots, notL_n, \tag{2}$$

$body(r_{ic})^+ = \{L_1, \ldots, L_m\}$ and $body(r_{ic})^-=\{L_1, \ldots, L_m\}$.
Then for $E \subseteq Args_P$, whether E satisfies IC is defined as follows.

- *E violates IC iff $E \cup IC$ is inconsistent*
  iff $\exists r_{ic} \in IC$ s.t. $body(r_{ic})^+ \subseteq claims(E)$ and $body(r_{ic})^- \cap claims(E) = \emptyset$
- *E satisfies IC iff E does not violate IC iff $E \cup IC$ is consistent*
  iff $\forall r_{ic} \in IC$ if $body(r_{ic})^- \cap claims(E) = \emptyset$, then $body(r_{ic})^+ \not\subseteq claims(E)$.

The semantics of a constrained argumentation framework is given by $\mathcal{C}$-extensions as follows.

Definition 21 *(***$\mathcal{C}$-extensions***) Let $CAF(P, IC) = (Args_P, \; attacks_P, IC)$ be a constrained argumentation framework associated with an ELP $P \cup IC$, and $AF_P = (Args_P, \; attacks_P)$ be the argumentation framework associated P. Then the semantics of $CAF(P, IC)$ is defined as follows. For $E \subseteq Args_P$,*

- *E is C-admissible iff E is admissible for AF_P and satisfies IC.*
- *E is a complete C-extension of $CAF(P, IC)$ iff E is a complete extension of AF_P and satisfies IC.*
- *E is a preferred C-extension of $CAF(P, IC)$ iff E is a preferred extension of AF_P and satisfies IC.*
- *E is a grounded C-extension of $CAF(P, IC)$ iff E is a grounded extension of AF_P and satisfies IC.*
- *E is a stable C-extension of $CAF(P, IC)$ iff E is a stable extension of AF_P and satisfies IC.*

The following Theorem extends Theorem 1 to handle integrity constraints.

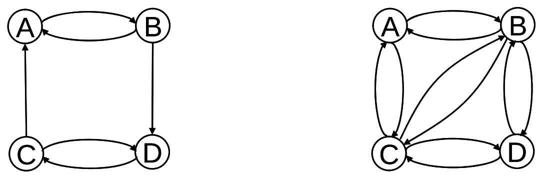


Fig. 1. Argumentation Frameworks (AF_Ps) of Example 3 and Example 4

Theorem 2. *Let* $CAF(P, IC) \overset{def}{=} (Args_P,\ attacks_P, IC)$ *be a constrained argumentation framework associated with an ELP* $P \cup IC$*, where* P *and* IC *are the sets of rules of the form (1) and (2) respectively, and* $attacks_P$ *is the binary relation over* $Args_P$ *which is defined according to undercut (i.e.* `u`*) as the notion of attack. Then* S *is an answer set of* $P \cup IC$ *iff there is a stable* $\mathcal{C}$*-extension* E *of* $CAF(P, IC)$ *such that* $S = claims(E)$*.*

Proof: See appendix.

Example 3. Let us consider the following ELP $P \cup IC$:

$$\begin{array}{lll} P: & p \leftarrow not\ q, & q \leftarrow not\ p, \\ & q \leftarrow not\ r, & r \leftarrow not\ q. \\ IC: & \leftarrow p, r. & \end{array}$$

P has two answer sets, S_1 and S_2 such that $S_1 = \{p, r\}$ and $S_2 = \{q\}$, whereas $P \cup IC$ has only one answer sets, S_2. On the other hand, the set $Args_P$ of minimal arguments associated with P is $\{A, B, C, D\}$ such that

$$\begin{array}{ll} A = [p \leftarrow not\ q], & B = [q \leftarrow not\ p] \\ C = [q \leftarrow not\ r], & D = [r \leftarrow not\ q], \end{array}$$

whereas the attack relation, $attacks_P$ is obtained according to *undercut* as follows,

$$attacks_P = \{(A, B), (B, A), (C, D), (D, C), (C, A), (B, D)\},$$

Thus, w.r.t. $AF_P = (Args_P,\ attacks_P)$ whose graph is shown as the left one of Fig. 1, there are two preferred as well as stable extensions, E_1 and E_2 as follows:

$$E_1 = \{A, D\}, \qquad E_2 = \{B, C\}$$

with $claims(E_1) = \{p, r\}$ and $claims(E_2) = \{q\}$.

Instead, E_2 is the preferred $\mathcal{C}$-extension as well as the stable $\mathcal{C}$-extension of $CAF(P, IC) = (Args_P, attacks_P, IC)$, but E_1 is neither of them since $claims(E_1) \cup IC$ is inconsistent, but $claims(E_2) \cup IC$ is consistent. Note that $claims(E_2)$ coincides with the answer set $S_2 = \{q\}$ of $P \cup IC$ as addressed by Theorem 2.

4.2 Preference-based AFs translated from PLPs

Here, we show a non-abstract preference-based argumentation framework translated from a PLP $(P \cup IC, \Phi)$.

Definition 22 *(***From PLPs to Preference-based AFs***) For a PLP $(P \cup IC, \Phi)$, the non-abstract preference-based argumentation framework $PAF(P, IC, \Phi)$ associated with the PLP is defined as follows:*

$$PAF(P, IC, \Phi) = (Args_P, attacks_P,\ IC,\ \leq)$$

where $\leq$ is a priority relation between arguments defined as a preorder [3] *on $Args_P$ such that,*

$Ag_1 \leq Ag_2$ *iff* $e_1 \preceq e_2 \in \Phi^*$ *for* $claim(Ag_1) = e_1$ *and* $claim(Ag_2) = e_2$.
For any argument Ag_1 and Ag_2 from $Args_P$, $Ag_1 \leq Ag_2$ or $(Ag_1, Ag_2) \in \leq$ is called "a priority between arguments", and we say Ag_2 has a higher priority than Ag_1. We write $Ag_1 < Ag_2$ if $Ag_1 \leq Ag_2$ and $Ag_2 \not\leq Ag_1$, and say "Ag_2 has a strictly higher priority than Ag_1". When IC is empty, we may write

$$PAF(P, \Phi) = (Args_P, attacks_P, \leq)$$

instead of $PAF(P, \emptyset, \Phi) = (Args_P, attacks_P,\ \emptyset,\ \leq)$.

Note that $\leq$ is equivalent to $\leq^*$, i.e. the reflexive and transitive closure of $\leq$. In our approach, given a PLP $(P \cup IC, \Phi)$, *preferences between $\mathcal{C}$-extensions* are defined w.r.t. $PAF(P, IC, \Phi)$ as follows.

Definition 23 *(***Preferences between $\mathcal{C}$-extensions***) Let Sname be be one of* `complete`, `stable`, `preferred`, *and* `grounded`. *For a PLP $(P \cup IC, \Phi)$, let $PAF(P, IC, \Phi) = (Args_P, attacks_P,\ IC,\ \leq)$ be the non-abstract preference-based argumentation framework, $CAF(P, IC) = (Args_P,\ attacks_P, IC)$ be the constrained argumentation framework, and $\mathcal{E}$ be the set of Sname $\mathcal{C}$-extensions for $CAF(P, IC)$. Then a preference relation $\sqsubseteq_{ex}$ over $\mathcal{E}$ is defined as follows. For any $\mathcal{C}$-extensions, E_1, E_2 and E_3 from $\mathcal{E}$,*

1. *$E_1 \sqsubseteq_{ex} E_1$,*
2. *$E_1 \sqsubseteq_{ex} E_2$ if for some argument $Ag_2 \in E_2 \setminus E_1$,*
   (i) there is an argument $Ag_1 \in E_1 \setminus E_2$ s.t. $Ag_1 \leq Ag_2$ w.r.t. $\leq$, and
   (ii) there is no argument $Ag_3 \in E_1 \setminus E_2$ s.t. $Ag_2 < Ag_3$ w.r.t. $\leq$,
3. *if $E_1 \sqsubseteq_{ex} E_2$ and $E_2 \sqsubseteq_{ex} E_3$, then $E_1 \sqsubseteq_{ex} E_3$.*

Note that $\sqsubseteq_{ex}$ is reflexive and transitive according to the items no.1 and no.3. We say that E_2 is preferable to E_1 with respect to $\leq$ if $E_1 \sqsubseteq_{ex} E_2$ holds. We write $E_1 \sqsubset_{ex} E_2$ if $E_1 \sqsubseteq_{ex} E_2$ and $E_2 \not\sqsubseteq_{ex} E_1$. Hereafter, each $E_1 \sqsubseteq_{ex} E_2$ is called "a preference between $\mathcal{C}$-extensions".

The semantics of $PAF(P, IC, \Phi)$ is given by $\mathcal{P}$-extensions as follows.

Definition 24 *(***$\mathcal{P}$-extensions***) Let Sname be be one of* `complete`, `stable`, `preferred`, *and* `grounded`. *For a PLP $(P \cup IC, \Phi)$, let $\mathcal{E}$ be the set of the Sname $\mathcal{C}$-extensions for $CAF(P, IC)$. Then a $\mathcal{C}$-extension $E \in \mathcal{E}$ is is called a Sname $\mathcal{P}$-extension of $PAF(P, IC, \Phi)$ if $E \sqsubseteq_{ex} E'$ implies $E' \sqsubseteq_{ex} E$ (with respect to $\leq$) for any $E' \in \mathcal{E}$. In other words, $E \in \mathcal{E}$ is a Sname $\mathcal{P}$-extension of $PAF(P, IC, \Phi)$ iff $E \not\sqsubset_{ex} E'$ with respect to $\leq$ for any $E' \in \mathcal{E}$.*

[3] A binary relation is a *preorder* iff it is *reflexive* and *transitive*

The following theorem shows that stable $\mathcal{P}$-extensions of $PAF(P, IC, \Phi)$ capture preferred answer sets of a PLP $(P \cup IC, \Phi)$, which extends Theorem 2.

Theorem 3. *For a PLP $(P \cup IC, \Phi)$, let $PAF(P, IC, \Phi) = (Args_P, attacks_P, IC, \Phi)$ be the non-abstract preference-based argumentation framework, where $attacks_P$ is the binary relation over $Args_P$ defined according to undercut (i.e.* `u`*). Then S is a preferred answer set (i.e. p-answer set) of a PLP $(P \cup IC, \Phi)$ iff there is a stable $\mathcal{P}$-extension E of $PAF(P, IC, \Phi)$ such that $S = claims(E)$.*

Proof: See appendix.

The sceptical (resp. credulous) query-answering problem is uniformly handled for our preference-based argumentation framework as follows.

Definition 25 *(***Credulous / Skeptical query-answering***)*
For a PLP $(P \cup IC, \Phi)$, let $PAF(P, IC, \Phi) = (Args_P, attacks_P, IC, \Phi)$ be the preference-based argumentation framework and Sname be one of `complete`*,* `stable`*,* `preferred`*, and* `grounded`*. Then for an argument $Ag \in Args_P$,*

- *Ag is credulously justified w.r.t. $PAF(P, IC, \Phi)$ under Sname semantics iff Ag is contained in at least one Sname $\mathcal{P}$-extension of $PAF(P, IC, \Phi)$;*
- *Ag is skeptically justified w.r.t. $PAF(P, IC, \Phi)$ under Sname semantics iff Ag is contained in every Sname $\mathcal{P}$-extension of $PAF(P, IC, \Phi)$.*

The following proposition denotes that, Dung's acceptability semantics is the special case of our preference-based argumentation semantics.

Proposition 2 *Let Sname be be one of* `complete`*,* `stable`*,* `preferred`*, and* `grounded`*. For a PLP $(P \cup IC, \Phi)$ whose IC and Φ are empty, E is a Sname extension of an argumentation framework AF_P associated with P iff E is a Sname $\mathcal{P}$-extension of $PAF(P, IC, \Phi)$.*

In the following examples, each $attacks_P$ is constructed based on *undercut* as the notion of attack in order to illustrate Theorem 3.

Example 4. Let us consider a PLP (P, Φ) of Example 4.2 in [17] as follows:

$$\begin{aligned}
P:\quad & p \leftarrow not\ q,\ not\ r,\\
& q \leftarrow not\ p,\ not\ r,\\
& r \leftarrow not\ p,\ not\ q,\\
& s \leftarrow p.\\
\Phi:\quad & p \preceq q,\ r \preceq s.
\end{aligned}$$

P has three answer sets $S_1 = \{p, s\}$, $S_2 = \{q\}$, $S_3 = \{r\}$, whereas the PLP (P, Φ) has the unique p-answer set, $S_2 = \{q\}$ since $S_3 \sqsubseteq_{as} S_1$, $S_1 \sqsubseteq_{as} S_2$ and $S_3 \sqsubseteq_{as} S_2$ due to $p \preceq q,\ r \preceq s$ from Φ.

On the other hand, according to Definition 22, the preference-based argumentation framework associated with (P, Φ) is $PAF(P, \Phi) = (Args_P, attacks_P, \leq)$, where $Args_P$ is $\{A, B, C, D\}$ such that

$$A = [p \leftarrow not\ q, not\ r], \qquad B = [q \leftarrow not\ p, not\ r]$$
$$C = [r \leftarrow not\ p, not\ q], \qquad D = [s \leftarrow p;\ p \leftarrow not\ q, not\ r]$$

with $claim(A) = \{p\}$, $claim(B) = \{q\}$, $claim(C) = \{r\}$ and $claim(D) = \{s\}$, $attacks_P$ is the following attack relation derived according to *undercut*,
$\{(A,B),(B,A),(C,A),(A,C),(B,C),(C,B),(B,D),(D,B),(C,D),(D,C)\}$
and $\leq$ is the binary relation over $Args_P$ such that $\leq = \{(A,B),(C,D)\} \cup \Psi$ where $\Psi = \{(x,x) | x \in Args_P\}$ since $p \preceq q \in \Phi$ for $claim(A) = \{p\}$ and $claim(B) = \{q\}$, and $r \preceq s \in \Phi$ for $claim(C) = \{r\}$ and $claim(D) = \{s\}$. Now $AF_P = (Args_P, attacks_P)$ whose graph is shown as the right one of Fig.1 has three preferred extensions as follows:

$$E_1 = \{A, D\}, \qquad E_2 = \{B\}, \qquad E_3 = \{C\}$$

where $claims(E_1) = \{p, s\}$, $claims(E_2) = \{q\}$ and $claims(E_3) = \{r\}$. Note that they are also stable extensions. Therefore E_2 is the unique preferred (resp. stable) $\mathcal{P}$-extension of $PAF(P, \Phi)$ since $E_3 \sqsubseteq_{ex} E_1$, $E_1 \sqsubseteq_{ex} E_2$ and $E_3 \sqsubseteq_{ex} E_2$ due to $(A,B) \in\leq$, $(C,D) \in\leq$ and transitive law of $\sqsubseteq_{ex}$. Noted that the unique p-answer set, S_2 of the PLP coincides with $claims(E_2)$ for the stable $\mathcal{P}$-extension, E_2 of $PAF(P, \Phi)$.

Example 5. Consider a PLP $(P \cup IC, \Phi)$, where P and Φ are given in Example 4 and IC has the integrity constraint as follows:

IC: $\quad \leftarrow q.$

Then P has two answer sets $S_1 = \{p, s\}$ and $S_3 = \{r\}$, whereas the PLP $(P \cup IC, \Phi)$ has the unique p-answer set, S_1 since $S_3 \sqsubseteq_{as} S_1$.

On the other hand, the preference-based argumentation framework associated with a PLP $(P \cup IC, \Phi)$ is $PAF(P,\ IC,\ \Phi) = (Args_P, attacks_P, IC, \leq)$, where $AF_P = (Args_P, attacks_P)$ and $\leq$ are the same ones shown in Example 4.

Though there are three preferred as well as stable extensions, E_1, E_2 and E_3 for this AF_P, both E_1 and E_3 are preferred as well as stable $\mathcal{C}$-extensions of this $CAF(P, IC)$ but E_2 is not because both $claims(E_1) \cup IC$ and $claims(E_3) \cup IC$ are consistent, but $claims(E_2) \cup IC$ is inconsistent. As a result, according to Definition 23, $E_1 = \{A, D\}$ is not only the unique preferred $\mathcal{P}$-extension but also the unique stable $\mathcal{P}$-extension of $PAF(P,\ IC,\ \Phi)$ but E_3 is not. Note that, the unique p-answer set, $S_1 = \{p, s\}$ of this PLP $(P \cup IC, \Phi)$ coincides with $claims(E_1)$ for $E_1 = \{A, D\}$ of $PAF(P, IC, \Phi)$.

Example 6. (Gordon's Perfected Shipping Problem)
Let us consider the famous legal reasoning example due to Gordon [12]. The problem is described as follows: *"A person wants to find out if her security interest in a certain ship is perfected. According to the Uniform Commercial Code (UCC) which is a state law, a security interest in goods may be perfected by taking possession of the collateral. However, the federal Ship Mortgage Act (SMA) states that a security interest in a ship may only be perfected by filing a financing statement. She currently has possession of the ship, but a statement*

has not been filed. Both UCC and SMA are applicable: the question is which takes precedence here." The situation is presented by ELP P_1 as follows.

$$\begin{array}{lll} P_1: & perfected \leftarrow posses, ucc, & (UCC) \\ & \neg perfected \leftarrow ship, \neg file, sma, & (SMA) \\ & posses \leftarrow, \quad ship \leftarrow, \quad \neg file \leftarrow, & \\ & ucc \leftarrow not\ \neg perfected, \quad sma \leftarrow not\ perfected. & \end{array}$$

Since the two laws are in conflict with one another, they lead to two answer sets S_1 and S_2 of P_1 as follows.

$$S_1 = \{perfected, posses, ship, \neg file, ucc\}.$$
$$S_2 = \{\neg perfected, posses, ship, \neg file, sma\}.$$

Now, there are two well-known legal principles for resolving such conflict between laws as follows.
"The principle of Lex Posterior gives precedence newer laws, and the principle of Lex Superior gives precedence to laws supported by the higher authority. In our case, UCC is newer than the SMA, and the SMA has higher authority since it is a federal law." Such knowledge may be described as the following sets:

$$\Phi_1 = \{sma \preceq ucc\},\ \Phi_2 = \{ucc \preceq sma\},\ \Phi_3 = \{sma \preceq ucc,\ ucc \preceq sma\},$$

where Φ_1 takes account of only the principle of Lex Posterior, Φ_2 only Lex Superior, and Φ_3 both. Then S_1 (resp. S_2) is the unique p-answer set of (P_1, Φ_1) (resp. (P_1, Φ_2)), but both of S_1 and S_2 become *tie* p-answer sets of (P_1, Φ_3) since $S_1 \sqsubseteq_{as} S_2$ and $S_2 \sqsubseteq_{as} S_1$ due to a conflict between these principles.

On the other hand, the preference-based argumentation framework associated with (P_1, Φ_i) is $PAF(P_1, \Phi_i)=(Args_P, attacks_P,\ \leq_i)$ (for $1 \leq i \leq 3$), where $Args_{P_1}$ is $\{A, B, C, D, F, G, H\}$ such that,

$$A = [perfected \leftarrow posses, ucc;\ posses;\ ucc \leftarrow not\ \neg perfected],$$
$$B = [\neg perfected \leftarrow ship, \neg file, sma;\ ship;\ \neg file; sma \leftarrow not\ perfected],$$
$$C = [ucc \leftarrow not\ \neg perfected],$$
$$D = [sma \leftarrow not\ perfected],$$
$$F = [posses \leftarrow], \qquad G = [ship \leftarrow], \qquad H = [\neg file \leftarrow]$$

with $claim(A) = \{perfected\}$, $claim(B) = \{\neg perfected\}$, $claim(C) = \{ucc\}$, $claim(D) = \{sma\}$, $claim(F) = \{posses\}$, $claim(G) = \{ship\}$, $claim(H) = \{\neg file\}$, $attacks_{P_1}$ is $\{(A, B), (B, A), (A, D), (B, C)\}$ derived according to *undercut*, and each $\leq_i$ is the binary relation over $Args_{P_1}$ such that $\leq_1 = \{(D, C)\} \cup \Psi$, $\leq_2 = \{(C, D)\} \cup \Psi$, $\leq_3 = \{(C, D), (D, C)\} \cup \Psi$ where $\Psi = \{(x, x) | x \in Args_{P_1}\}$ due to the respective Φ_i. In this case, $AF_{P_1}=(Args_{P_1}, attacks_{P_1})$ has two preferred as well as stable extensions, $E_1 = \{A,\ C,\ F,\ G,\ H\}$ and $E_2 = \{B,\ D,\ F, G,\ H\}$ with $claims(E_1) = \{perfected, ucc, posses, ship, \neg file\}$ and
$claims(E_2) = \{\neg perfected, sma, posses, ship, \neg file\}$,
whereas E_1 (resp. E_2) is the unique preferred as well as stable $\mathcal{P}$-extension of

$PAF(P_1, \Phi_1)$ (resp. $PAF(P_1, \Phi_2)$), but both E_1 and E_2 are the preferred as well as stable $\mathcal{P}$-extensions of $PAF(P_1, \Phi_3)$ since $E_1 \sqsubseteq_{ex} E_2$ and $E_2 \sqsubseteq_{ex} E_1$ due to $\leq_3$.

The following example shows that even for a PLP (P, Φ) whose P is inconsistent, intended results of argumentation are derived based on the PAF.

Example 7. (**Ex. 6 Cont.**) Consider the PLP (P_2, Φ_i) $(1 \leq i \leq 3)$ such that $P_2 = P_1 \cup \{ab \leftarrow not\ ab\}$, Due to the added rule to P_1, P_2 is inconsistent since it has no answer sets. Hence the PLP (P_2, Φ_i) with any Φ_i has no p-answer sets. This reveals the limitation of answer set programming which is only applicable to consistent knowledge bases. Instead, for the PLP (P_2, Φ_i), we have

$$PAF(P_2, \Phi_i) = (Args_{P_2}, attacks_{P_2},\ \leq_i) \quad (for\ 1 \leq i \leq 3),$$

where $Args_{P_2} = Args_{P_1} \cup \{I\}$ such that $I = [ab \leftarrow not\ ab]$ and $attacks_{P_2} = attacks_{P_1} \cup \{(I, I)\}$ as derived according to *undercut*. In this case, $AF_{P_2} = (Args_{P_2}, attacks_{P_2})$ has no stable extensions but has the same two preferred extensions, E_1 and E_2 that AF_{P_1} has. Similarly, each $PAF(P_2, \Phi_i)$ $(1 \leq i \leq 3)$ has no stable $\mathcal{P}$-extensions but has the same preferred $\mathcal{P}$-extensions that $PAF(P_1, \Phi_i)$ has.

5 Related Work

Amgoud and Vesic [2] proposed only a new abstract PAF, whereas we present not only a new approach of an abstract PAF but also propose a non-abstract PAF constructed from a prioritized logic program. In our approach, we can show Theorem 3 for such a non-abstract PAF as is the generalization of Theorem 1 presented by Dung [8]. This property ensures advantages as well as correctness of our approach.

Coste-Marquis *et al.*[6] proposed an abstract CAF where constraints are expressed by a propositional formula defined over the set of abstract arguments, whereas in our approach, a non-abstract CAF is defined where constraints are given as nonmonotonic rules embedded in an extended logic program expressing a agent's domain knowledge. From the computational point of view, Besnard and Doutre's approach [3] for encoding acceptable semantics can be applied to their CAF, whereas a non-abstract CAF presented in this paper can be easily encoded in ASP setting by extending our previous work [20] to compute argumentation semantics in ASP based on Caminada's reinstatement labellings [5].

Šefránek [16] presented the semantics, i.e. preferred answer sets of a prioritized logic program $(P, \prec, \mathcal{N})$ based on argumentation, where P is an ELP, $\prec$ is a strict partial order on rules of P and $\mathcal{N}$ is a function assigning names to rules of P. He proposed an argumentation framework translated from such a prioritized logic program, and defined preferred answer sets in his framework. However, not only argumentation framework proposed in [16] is inapplicable to a inconsistent P but also it is not the generalization of Dung's argumentation framework fro handling additional preferences.

6 Conclusion

To handle preferences along with constraints, we present new abstract preference-based argumentation frameworks as well as the non-abstract ones translated from prioritized logic programs. In our approach, we can show Theorem 3 such that, stable $\mathcal{P}$-extensions of a preference-based argumentation framework $PAF(P, IC, \Phi)$ associated with the PLP $(P \cup IC, \Phi)$ capture p-answer sets of the PLP. Hence advantages and correctness of our approach are ensured.

On the other hand, when agent's knowledge expressed by an ELP P is inconsistent, we cannot reason anything from a PLP (P, Φ) as well as from our $PAF(P, \Phi)$ under stable semantics, since there are no p-answer sets of the PLP as well as no stable $\mathcal{P}$-extensions of $PAF(P, \Phi)$. However, with such inconsistent P, we can infer the intended results from a non-abstract $PAF(P, \Phi)$ under preferred semantics because there exists a preferred $\mathcal{P}$-extension for $PAF(P, \Phi)$. Thus in some sense, a non-abstract PAF presented in the paper can be regarded as the extended PLP.

Applying the techniques used in our previous works [18–20], the encoding to compute $\mathcal{P}$-extension of the non-abstract PAF can be easily established in ASP setting. Thus such a system which encodes the PAF presented in the paper will behave as the enhanced PLP system such that not only it can compute p-answer sets of a PLP via the stable $\mathcal{P}$-extensions but also it can infer intended results via the preferred $\mathcal{P}$-extensions even if P is inconsistent.

Our future works are not only to investigate computational complexity of the proposed methods but also to implement the PAF system in ASP setting so that it may be used in multiagent systems of negotiation based on argumentation.

Acknowledgments: This research is partially supported by Grant-in-Aid for Scientific Research from JSPS, No. 20500141.

References

1. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. In Annals of Mathematics and Artificial Intelligence, Vol. 34, Issue 1-3, pp. 197-215 (2002)
2. Amgoud, L., Vesic, S.: Repairing preference-based argumentation frameworks. In: Proceedings of IJCAI 2009, pp. 665-670 (2009)
3. Besnard, P., Doutre, S.: Checking the acceptability of a set of arguments. In: Proceedings of NMR-2004, pp.59–64 (2004)
4. Brewka, G., Eiter, T.: Preferred answer sets for extended logic programs. *Artificial Intelligence* **109**, pp. 297-356 (1999)
5. Caminada, M.: On the issue of reinstatement in argumentation. In: Proceedings of JELIA-2006, LNAI(LNCS), vol. 4160, pp. 111–123. Springer (2006)
6. Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: Proceedings of KR'06, pp. 112-122 (2006)
7. Delgrande, J. P., Schaub, T., Tompits, H.: A framework for compiling preferences in logic programs. Theory and Practice of Logic Programming 3(2), pp. 129-187 (2003)
8. Dung, P.M.: An argumentation semantics for logic programming with explicit negation. In: Proceedings of ICLP 1993, MIT press, pp.616-630 (1993)

9. Dung, P.M.: 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)
10. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the fifth International Conference and Symposium on Logic Programming (ICLP/SLP-1988), pp. 1070–1080. MIT Press (1988)
11. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9, pp. 365–385 (1991)
12. Gordon, Thomas F., The pleadings game: An Artificial Intelligence Model of Procedural Justice. Dissertation, TU Darmstadt (1993)
13. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics 7(1):25-75 (1997)
14. Prakken, H., Vreeswijk, G.A.W.: Logics for defeasible argumentation. In: D.M. Gabbay and F. Guenthner, eds., Handbook of Philosophical Logic, Second Ed., vol. 4, pp. 218–319. Kluwer, Dordecht (2001)
15. Schweimeier, R., Schroeder, M.: A Parameterized hierarchy of argumentation semantics for extended logic programming and its application to the well-founded semantics. Theory and Practice of Logic Programming 5(1,2), pp. 207–242. Cambridge University Press (2005)
16. *Šefránek* J.: Preferred answer sets supported by arguments. In: Proceedings of NMR-2008, pp.232-240 (2008)
17. C. Sakama, C., Inoue, K.: Prioritized logic programming and its application to commonsense reasoning. *Artificial Intelligence 123*, pp. 185-222 (2000)
18. Wakaki, T., Inoue, K., Sakama, C., Nitta, K.: Computing preferred answer sets in answer set programming. In: Proceedings of LPAR 2003, LNAI(LNCS), vol. 2850, pp. 259–273, Springer (2003)
19. Wakaki, T., Inoue, K., Sakama, C., Nitta, K.: The PLP system. Proc. of JELIA 2004, LNAI(LNCS), vol. 3229, Springer, pp. 706-709 (2004)
20. Wakaki, T., Nitta, K.: Computing argumentation semantics in answer set programming. New Frontiers in Artificial Intelligence, LNAI(LNCS), vol. 5547, pp. 254-269. Springer (2009)

Appendix: Proofs of Theorems

Proof of Theorem 2

Proof: ($\Longleftarrow$) Suppose E is a stable $\mathcal{C}$-extension of $CAF(P, IC)$= $(Args_P, attacks_P, IC)$. According to Definition 21, E is a stable extension of $AF_P = (Args_P, attacks_P)$ and satisfies IC. Therefore, there is the answer set S of P such that $S = claims(E)$ due to Theorem 1. Thus according to Definition 2, S is also an answer set of the *not*-free P^S, i.e. the reduct of P.
Now, since such E satisfies IC, which means that, for $S = claims(E)$,

$$\forall r_{ic} \in IC \text{ if } body(r_{ic})^- \cap S = \emptyset, \text{ then } body(r_{ic})^+ \not\subseteq S,$$

the answer set S of P^S satisfies $body(r_{ic})^+ = \{L_1, \ldots, L_m\} \not\subseteq S$ if $body(r_{ic})^- \cap S = \{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$ for any integrity constraint $r_{ic} \in IC$ as follows:

$$r_{ic}: \qquad \leftarrow L_1, \ldots, L_m, notL_{m+1}, \ldots, notL_n.$$

Therefore it is concluded that S is an answer set of $(P \cup IC)^S$. Hence $S = claims(E)$ is an answer set of $P \cup IC$.
($\Longrightarrow$) The converse is also proved similarly. □

After preparing the following lemma, we show the proof of Theorem 3.

Lemma 1. *For a PLP* $(P \cup IC, \Phi)$*, let* $PAF(P, IC, \Phi)$ *be* $(Args_P, attacks_P, IC, \leq)$*,* $CAF(P, IC)$ *be* $(Args_P, attacks_P, IC)$*,* E_1*,* E_2 *be stable* $\mathcal{C}$*-extensions of* $CAF(P, IC)$*, and* S_1*,* S_2 *be answer sets of* $P \cup IC$*. Then it holds that,*

$$E_1 \sqsubseteq_{ex} E_2 \quad \textit{iff} \quad S_1 \sqsubseteq_{as} S_2 \quad \textit{for } S_1 = claims(E_1) \textit{ and } S_2 = claims(E_2).$$

Proof:
Suppose E is a stable $\mathcal{C}$-extension of $CAF(P, IC)$. Then according to Theorem 2, $claims(E)$ coincides with an answer set S of $P \cup IC$. Moreover, for an argument $Ag \in Args_P$ and its claim $e \in Lit_P$, i.e. $e = claim(Ag)$, it holds that,

$$Ag \in E \quad \text{iff} \quad e \in S, \qquad \text{and} \qquad Ag \notin E \quad \text{iff} \quad e \notin S. \tag{3}$$

Now with respect to stable $\mathcal{C}$-extensions E_1, E_2 of $CAF(P, IC)$ whose claims are $S_1 = claims(E_1)$, $S_2 = claims(E_2)$ respectively, it holds that, due to (3), for a literal $e_2 \in Lit_P$ such that $e_2 = claim(Ag_2)$,

$$Ag_2 \in E_2 \setminus E_1 \text{ iff } Ag_2 \in E_2 \text{ and } Ag_2 \notin E_1 \text{ iff } e_2 \in S_2 \text{ and } e_2 \notin S_1$$
$$\text{iff } e_2 \in S_2 \setminus S_1. \tag{4}$$

Similarly for a literal $e_1 \in Lit_P$ such that $e_1 = claim(Ag_1)$, it hold that,

$$Ag_1 \in E_1 \setminus E_2 \text{ iff } e_1 \in S_1 \setminus S_2. \tag{5}$$

On the other hand, according to Definition 22,

$$Ag_1 \leq Ag_2 \text{ iff } e_1 \preceq e_2 \in \Phi^* \text{ for } claim(Ag_1) = e_1 \text{ and } claim(Ag_2) = e_2. \tag{6}$$

Thus due to (4), (5), (6), it holds that,

$$\exists Ag_2 \in E_2 \setminus E_1 \text{ and } \exists Ag_1 \in E_1 \setminus E_2 \text{ such that } Ag_1 \leq Ag_2$$
$$\text{iff } \exists e_2 \in S_2 \setminus S_1 \text{ and } \exists e_1 \in S_1 \setminus S_2 \text{ such that } e_1 \preceq e_2 \in \Phi^*. \tag{7}$$

Therefore by extending (7), it is obviously derived that,

$$\exists Ag_2 \in E_2 \setminus E_1 [\ \exists Ag_1 \in E_1 \setminus E_2 \text{ such that } Ag_1 \leq Ag_2$$
$$\wedge \neg \exists Ag_3 \in E_1 \setminus E_2 \text{ s.t. } Ag_2 < Ag_3 \text{ w.r.t. } \leq],$$
$$\text{iff } \exists e_2 \in S_2 \setminus S_1 [\exists e_1 \in S_1 \setminus S_2 \text{ such that } e_1 \preceq e_2 \in \Phi^*$$
$$\wedge \neg \exists e_3 \in S_1 \setminus S_2 \text{ s.t. } e_2 \prec e_3 \in \Phi^*] \tag{8}$$

where $S_i = claims(E_i)$ and $e_j = claim(Ag_j)$ $(1 \leq i \leq 2,\ 1 \leq j \leq 3)$.
(8) means that $E_1 \sqsubseteq_{ex} E_2$ iff $S_1 \sqsubseteq_{as} S_2$ for $S_1 = claims(E_1)$ and $S_2 = claims(E_2)$ w.r.t. the item no.2 of Definition 23 and that of Definition 5. Since both $\sqsubseteq_{ex}$ and $\sqsubseteq_{as}$ are reflexive and transitive, it also holds that, $E_1 \sqsubseteq_{ex} E_2$ iff $S_1 \sqsubseteq_{as} S_2$ w.r.t. items no.1 and no.3 of these definitions. □

Proof of Theorem 3

Proof: For a PLP$(P \cup IC, \Phi)$, let AS be the set of all answer sets of $P \cup IC$ and $\mathcal{E}$ be the set of all stable $\mathcal{C}$-extensions of $CAF(P, IC)$= $(Args_P, attacks_P, IC)$. Then, it follows that,

$E \in \mathcal{E}$ is a stable $\mathcal{P}$-extensions of $PAF(P, IC, \leq)$ built on a PLP$(P \cup IC, \Phi)$
iff $E \sqsubseteq_{ex} E'$ implies $E' \sqsubseteq_{ex} E$ (with respect to $\leq$) for any $E' \in \mathcal{E}$
iff w.r.t. $S = claims(E) \in AS$,
$S \sqsubseteq_{as} S'$ implies $S' \sqsubseteq_{as} S$ (with respect to Φ) for any $S' = claims(E') \in AS$
due to Theorem 2 and Lemma 1,
iff $S = claims(E) \in AS$ is a preferred answer set of (P, Φ). □

Arguing About Preferences And Decisions[⋆]

T.L. van der Weide, F. Dignum, J.-J. Ch. Meyer,
H. Prakken, and G.A.W. Vreeswijk

Universiteit Utrecht
{tweide,dignum,jj,henry,gv}@cs.uu.nl

Abstract. Complex decisions involve many aspects that need to be considered, which complicates determining what decision has the most preferred outcome. Artificial agents may be required to justify and discuss their decisions to others. Designers must communicate their wishes to artificial agents. Research in argumentation theory has examined how agents can argue about what decision is best using goals and values. Decisions can be justified with the goals they achieve, and goals can be justified by the values they promote. Agents may agree on having a value, but disagree about what constitutes that value. In existing work, however, it is not possible to discuss what constitutes a specific value, whether a goal promotes a value, why an agent has a value and why an agent has specific priorities over goals. This paper introduces several argument schemes, formalised in an argumentation system, to overcome these problems. The techniques presented in this paper are inspired by multi attribute decision theory.

Categories and Subject Descriptors:
I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods
General Terms: Design
Keywords: Argumentation, Decision-Making, Practical Reasoning, Preferences

1 Introduction

In complex situations, decisions involve many aspects that need to be considered. These aspects are typically different in nature and therefore difficult to compare. This complicates determining what outcome is the most preferable. Throughout the paper, we will use buying a house as an example, which involves many different aspects. For example, an agent may care about the costs of a house, but also about how fun, comfortable, close to shops, and how beautiful a house is. Artificial agents are expected to act in the designer's or user's best interest. This requires designers or users to communicate their wishes to the agent and the agent to explain and discuss why a certain decision was made. A significant amount of research has been concerned with using argumentation

[⋆] 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.

theory for decision-making and practical reasoning to determine which decisions are defensible from a given motivation, see for example [7, 2, 1].

A possible argumentation framework for decision-making for this purpose is the one proposed in [1]. Several decision principles are formalised to select the best decision using arguments in favour and against the available decisions. Agents are assumed to have a set of prioritised goals, which are used to construct argument in favour and against decisions. For example, agent α has the goal to live in a house that is downtown and the less important goal to live in a house bigger than $60m^2$. In complex situations it is useful to argue about *what* goals should be pursued. Why have the goal to live downtown and why not in a village? Why is living downtown more important than living in a bigger house? However, justifying and attacking goals is not possible using the framework of [1].

In order to solve this problem, we could use the framework described in [2], where goals can be justified and attacked using the values they promote and demote. People use their values as standards or criteria to guide selection and evaluation of actions [10, 12]. The values of agents reflect their preferences. For example, agent α has the value of fun and of comfort. The goal to live downtown promotes the value of fun and the goal to live in a bigger house promotes the value of comfort. In [2] an argument scheme is proposed for practical reasoning in which goals and actions can be justified and attacked by the values they promote and demote. What constitutes a specific value like fun, comfort, justice, or health often is disputable and therefore it is also disputable whether goals and actions promote values. Namely, another agent may find that living downtown demotes the value of fun because of the noise and lack of parking space. However, in [2] it is not possible to explain or discuss what constitutes a value and consequently it is also not possible to justify or attack that a goal or action promotes or demotes a value.

This paper presents an argumentation approach to discuss what constitutes a specific value and its effects on agent's goals and preferences over outcomes. To argue about decisions, an argumentation system is described in Section 2. Since the subject of argumentation is making decisions, some basic notions of decision theory are also described in Section 2. Next, we propose a model to specify the meaning of values and their relation to preferences in Section 3. This model is based on previous work [15] and inspired by techniques from decision theory to find an appropriate multi-attribute utility function [8, 9]. A value is seen as an aspect over which an agent has preferences and can be decomposed into the aspects it contains. Given the meaning of a value, several argument schemes are proposed in Section 4 to justify that goals promote or demote values. The introduced formalism is demonstrated with an example of buying houses in Section 5. The paper is concluded with some discussion and conclusions.

2 Background

In Section 2.1, an argumentation system is described that will be used to argue about what decision is best. Outcomes describe the effects of decisions and attributes describe properties of outcomes. Attributes of outcomes can be used to describe what constitutes a value and to justify goals. To argue about what decision is best, the notions of outcomes and attributes from decision theory are introduced in our argumentation system in Section 2.2.

2.1 Argumentation

Argument schemes are stereotypical patterns of defeasible reasoning [14]. An argument scheme consists of a set of premises, a conclusion, and is associated to a set of critical questions that can be used to critically evaluate the inference. In later sections, argument schemes are proposed to reason about what decision is best.

We introduce an argumentation system to reason defeasibly and in which argument schemes can be expressed. For the largest part, this argumentation system is based on [5]. We will use both defeasible and strict 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.

Definition 1 (Argumentation System). *An argumentation system is a tuple $\mathcal{AS} = (\mathcal{L}, \mathcal{R})$ with $\mathcal{L}$ the language of first-order logic and $\mathcal{R}$ a set of strict and defeasible inference rules.*

We will use ϕ and ψ as typical elements of $\mathcal{L}$ and say that ϕ and $\neg\phi$ are each other's complements. In the meta-language, $\sim\phi$ denotes the complement of any formula ϕ, positive or negative. Furthermore, $\rightarrow$ denotes the material implication.

Definition 2 (Strict and defeasible rules). *A strict rule is an expression of the form $s(x_1, \ldots, x_n) : \phi_1, \ldots, \phi_m \Rightarrow \phi$ and a defeasible rule is an expression of the form $d(x_1, \ldots, x_n) : \phi_1, \ldots, \phi_m \rightsquigarrow \phi$, with $m \geq 0$ and $x_1, \ldots, x_n$ all variables in $\phi_1, \ldots, \phi_m, \phi$.*

We call $\phi_1, \ldots, \phi_m$ the antecedent, ϕ the conclusion, and both $s(x_1, \ldots, x_n)$ and $d(x_1, \ldots, x_n)$ the identifier of a rule.

Arguments are inference trees constructed from a knowledge-base $\mathcal{K} \subset \mathcal{L}$. If an argument A was constructed using no defeasible inference rules, then A is called a *strict argument*, otherwise A is called a *defeasible argument*.

Example 1. Let $\mathcal{AS} = (\mathcal{L}, \mathcal{R})$ be an argumentation system such that $\mathcal{L} = \{\phi_1, \phi_2, \phi_3\}$ and $\mathcal{R} = \{s() : \phi_1 \Rightarrow \phi_2; d() : \phi_1, \phi_2 \rightsquigarrow \phi_3\}$. From the knowledge-base $\mathcal{K} = \{\phi_1\}$, we can construct 3 arguments. Argument A_1 has conclusion ϕ_1, no premises, and no last applied inference rule. Argument A_2 is constructed by applying $s()$. Consequently, A_2 has premise A_1, conclusion ϕ_2, and last applied inference rule $s()$. Argument A_3 can then be constructed using $d()$ and has premises A_1 and A_2, conclusion ϕ_3, and last applied rule $d()$. Arguments A_1 and

A_2 are strict arguments and argument A_3 is a defeasible argument. A_3 can be visualised as follows:

$$\frac{\phi_1 \quad \frac{\phi_1}{\phi_2}\ s()}{\phi_3}\ d()$$

All arguments can be attacked by rebutting one of their premises. Defeasible arguments can also be attacked by attacking the application of a defeasible rule. For example, let $d(c_1, \ldots, c_n) : \phi_1, \ldots, \phi_m \rightsquigarrow \phi$ be a defeasible inference rule that was applied in argument A. We can attack A in three ways: by rebutting a premise of A, by rebutting A's conclusion, and by undercutting a defeasible inference rule that was applied in A. The application of a defeasible inference rule can be undercut when there is an exception to the rule. An argument concluding $\sim d(c_1, \ldots, c_n)$ undercuts A.

Following [4], argument schemes are formalised as defeasible inference rules. Critical questions point to counterarguments that either rebut the scheme's premises or undercut the scheme. In Section 3.4, we show how to determine what conclusions are justified given a set of arguments.

2.2 Outcomes And Attributes

The notion of *outcomes* is one of the main notions in decision theory [8, 11] and is used to represent the possible consequences of an agent's decisions. The set Ω of possible outcomes should distinguish all consequences that matter to the agent and are possibly affected by its actions. Agents have preferences over outcomes and decision theory postulates that a rational agent should make the decision that leads to the most preferred expected outcome.

The notion of *attribute* is used to denote a feature, characteristic or property of an outcome. For example, when buying a house, relevant attributes could be price, neighbourhood in which it is located, size, or type of house. An attribute has a domain of 'attribute-values' outcomes can have. Every outcome has exactly one attribute-value of each attribute. It cannot be that an outcome has two attribute-values of the same attribute.

Example 2 (Buying a house). There are 2 houses on the market and buying one of them results in one of the two outcomes $\Omega = \{\omega_1, \omega_2\}$. Consider the attributes 'price', 'size', 'neighbourhood', and whether there is a garden. Price is expressed in dollar and size in m^2. The neighbourhood can either be 'downtown' or 'suburb' and 'yes' represents there is a garden and 'no' that there is not.

Outcome ω_1 has the following attribute-values: price is 150.000, size is 50, neighbourhood is 'suburb' and garden is 'yes'. On the other, outcome ω_2's price is 200.000, size is also 50, neighbourhood is 'downtown' and garden is 'no'.

Each attribute is a term in $\mathcal{L}$ and we use $\mathcal{A}$ to denote the set containing all attributes. If x is an attribute, we will also say x-values instead of the attribute values of attribute x. We define several functions concerning attributes and outcomes.

– The function $\mathsf{domain}(x)$ returns a set of attribute-values that the attribute x can have. For example, let attribute nbhd denote the neighbourhood of a house, then $\mathsf{domain}(\mathsf{nbhd}) = \{\mathsf{downtown}, \mathsf{suburb}\}$ or for the attribute price, $\mathsf{domain}(\mathsf{price}) = \mathbb{R}^+$.
– For each attribute x, the function $\overline{x} : \Omega \rightarrow \mathsf{domain}(x)$ gives the attribute-value of the given outcome for the attribute x. For example, $\overline{price}(\omega_1) = 150.000$.

Example 3. Suppose that $\Omega = \{\omega_1, \omega_2\}$ is true, x is an attribute and $\mathsf{domain}(x) = \{1, 2, 3\}$. In that case, the function $\overline{x}$ returns the following: $\overline{x}(\omega_1) = 3$ and $\overline{x}(\omega_2) = 1$.

3 Justification Of Preferences Over Outcomes

Preferences can be expressed in terms of outcomes, e.g. outcome A is preferred to outcome B. The more aspects are involved, the more difficult it becomes to directly express preferences over outcomes. Luckily, it is also natural to express preferences in terms of attributes of outcomes. For example, maximising the attribute profit is preferred. From such statements, preferences over outcomes can be justified, e.g. outcome A is preferred to outcome B because the A's profit is higher. Typically, outcomes have many attributes, yet agents care only about a subset. What set of attributes an agent cares about determines the preferences over outcomes. Using argumentation, agents can discuss why certain attributes should and others should not be used.

Justification for a preference statement like "agent α prefers living downtown to living in a suburb", is useful to better understand α's preferences. Namely, α could argue that the centrality of a house positively influences the amount of fun of a house and that α wants to maximise fun. If it is better understood why α prefers something, then one could disagree (centrality is not fun because it is very noisy) and give alternatives perspectives (living near nature is also fun and do you not also care about quietness).

In complex situations, the preferences of agents depend on multiple attributes. By decomposing an agent's preferences into the different aspects it involves, the number of attributes an aspect depends on becomes smaller. By recursively decomposing preferences, we will arrive at aspects that depend on a single attribute. For example, an agent α decomposes its preferences concerning houses into the aspects costs and comfort. The perspective of costs is determined by the attribute acquisition price. Comfort however, depends on multiple aspects. Therefore, comfort is decomposed into location and size. Location is then connected to the attribute neighbourhood and size to the surface area of the house. One may argue that α forgets that other aspects also influence costs, e.g. maintenance, taxes, heating costs, and so on. On the other hand, another agent may decompose comfort differently. For example, for agent β comfort is influenced by the closeness to highway and whether there is central heating.

In Section 3.1 we will introduce *perspectives* to represent preferences and aspects of preferences, after which we introduce perspectives on attribute values in Section 3.2. In Section 3.3 we introduce influence between perspective to denote that one perspective is an aspect of another. Finally in Section 3.4, we slightly adapt Value-based Argumentation Frameworks, see [3], to determine what conclusions are justified to make.

3.1 Perspectives

An ordering over outcomes can represent an agent's preferences. In that case, if an outcome is higher in the order, the agent prefers that outcome. Similarly, outcomes can be ordered according to some criterion. For example, outcomes can be ordered by how fun they are, or how fair they are. To talk about these different orderings, we introduce the notion of *perspective.* With buying houses, an outcome may be better than another from the perspective of costs, worse from the perspective of its centrality, indifferent from the perspective of comfort, and perhaps incomparable from the perspective of fun.

Definition 3 (Perspective). *A perspective p is associated with a preorder, $\leq_p$ over outcomes Ω. The set $\mathcal{P}$ denotes the set of all perspectives.*

In other words, a perspective p is associated to a binary relation $\leq_p \subseteq \Omega \times \Omega$ that is transitive and reflexive. If $\omega_1 \leq_p \omega_2$ is true, we say that ω_2 is weakly preferred to ω_1 from perspective p. Strong preference from perspective p is denoted as $\omega_1 <_p \omega_2$ and stands for $\omega_1 \leq_p \omega_2$ and $\omega_2 \not\leq_p \omega_1$. Equivalence from perspective p is denoted as $\omega_1 \approx_p \omega_2$ and stands for $\omega_1 \leq_p \omega_2$ and $\omega_2 \leq_p \omega_1$.

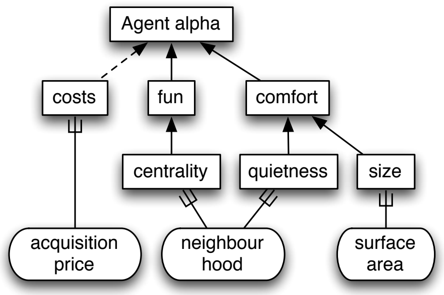
Each agent α is associated with a perspective $\hat{\alpha}$ representing α's preferences over outcomes. If $\omega_1 <_{\hat{\alpha}} \omega_2$ is true, then we either say that ω_2 is preferred to ω_1 from agent α's perspective, or we say that α prefers ω_2 to ω_1. Since perspectives are the main notion in this paper, $\hat{\alpha}$ is abbreviated to α, so that α denotes a perspective.

Not only the preferences of agents can be represented with perspectives, we will also use perspectives to represent aspects of outcomes and the values of agents. For example, the value of 'safety' is represented with a perspective that orders outcomes according to how safe they are or the aspect of comfort is represented with a perspective that orders outcomes by the amount of comfort.

Example 4. Agent α wants to buy a new house and wants to minimise costs, maximise fun and maximise comfort. In Figure 1a, we sketch how the preferences of agent α can be decomposed and how attributes of outcomes can be assigned. Costs are determined by the attribute acquisition price. Fun is influenced by the centrality of the house, i.e. the more central the neighbourhood, the more fun it is. Comfort is influenced by how quiet it is and the size of the house. Again, the attribute neighbourhood is ordered but now by how quiet the neighbourhood is. The size is determined by the surface area of the house.

Agent β just won the lottery and does not care about costs. To β fun is being close to nature, which is completely different from α's idea about fun. Also, because β has a car and α does not, β cares about whether there is enough parking space in the area. Figure 1b sketches the decomposition of β's preferences.

(a) α's preferences



(b) β's preferences

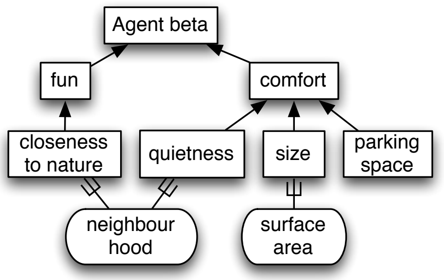


Fig. 1: Different Perspectives Using Different Attributes

3.2 Attributes Determine Perspectives

Attributes can be used to determine how outcomes should be ordered from a perspective. For example, if you want to order houses from the perspective of size, then the attribute 'surface area of the house' is an appropriate attribute. In that case, if house A has a higher surface area than house B, then A is preferred to B from the perspective of size. To use an attribute x to determine a perspective, x's attribute values need to be ordered.

Definition 4 (Attribute Perspective). *An* attribute perspective p_x *is a perspective that is associated with a partial preorder* $\preceq_x^p$ *over the domain of attribute* x*.*

Note that there can be different attribute perspectives on the same attribute.

Example 5. Let the attribute nbhd denote the neighbourhood of the house with $\mathsf{domain}(\mathsf{nbhd}) = \{\mathsf{dwntwn}, \mathsf{vllg}, \mathsf{sbrb}\}$. Furthermore, let $\mathsf{social}_{\mathsf{nbhd}}$ and $\mathsf{quiet}_{\mathsf{nbhd}}$ be attribute perspectives denoting the sociableness and the quietness of the neighbourhood respectively. The different neighbourhoods are preferred from each attribute perspective as follows:

$$\mathsf{sbrb} \prec_{\mathsf{nbhd}}^{\mathsf{social}} \mathsf{vllg} \prec_{\mathsf{nbhd}}^{\mathsf{social}} \mathsf{dwntwn} \qquad\qquad \mathsf{dwntwn} \prec_{\mathsf{nbhd}}^{\mathsf{quiet}} \mathsf{sbrb} \prec_{\mathsf{nbhd}}^{\mathsf{quiet}} \mathsf{vllg}$$

If an attribute value is preferred to another attribute value from p_x, then outcomes with the preferred attribute value are be preferred from p_x. This order between outcomes from an attribute perspective p_x can be inferred with the following strict inference rule.

$$s_{ap}(p_x, \omega_1, \omega_2) : \overline{x}(\omega_1) \prec_x^p \overline{x}(\omega_2) \Rightarrow \omega_1 <_{p_x} \omega_2$$

Example 6. Consider the attributes and attributes perspectives from the previous example. Let there be two outcomes ω_1 and ω_2 such that $\overline{\mathsf{nbhd}}(\omega_1) = \mathsf{dwntwn}$

and $\overline{\mathsf{nbhd}}(\omega_2) = \mathsf{vllg}$. To determine the order between ω_1 from perspective $\mathsf{social}_{\mathsf{nbhd}}$ and from perspective $\mathsf{quiet}_{\mathsf{nbhd}}$, the following two arguments can be constructed.

$$\frac{\overline{\mathsf{nbhd}}(\omega_2) \prec^{\mathsf{social}}_{\mathsf{nbhd}} \overline{\mathsf{nbhd}}(\omega_1)}{\omega_2 <_{\mathsf{social}_{\mathsf{nbhd}}} \omega_1}\; s_{ap} \qquad \frac{\overline{\mathsf{nbhd}}(\omega_1) \prec^{\mathsf{quiet}}_{\mathsf{nbhd}} \overline{\mathsf{nbhd}}(\omega_2)}{\omega_1 <_{\mathsf{quiet}_{\mathsf{nbhd}}} \omega_2}\; s_{ap}$$

3.3 Influence Between Perspectives

Some perspectives involve different aspects such that not one attribute can be assigned. For example, the perspective comfort of a house may involve size, location, the type of heating, and so on. In general, the more abstract a perspective is, the more aspects it has. Furthermore, the more abstract a perspective, the more disputable it may be. Thus it becomes important to specify all the different aspects so that one can communicate clearly.

By decomposing an abstract perspective into several more concrete perspectives, one makes explicit what an abstract perspective means and makes it easier to assign attributes to. For example, although α may not be able to express its preferences over houses, α does want to minimise costs, maximise comfort, and maximise fun. These perspectives may be decomposed further, e.g. fun means maximising the centrality of the house, until an attribute can be assigned.

To decompose a perspective into other perspectives, we introduce two relations between perspectives in $\mathcal{L}$ to denote 'influence' between perspectives:

- the binary relation $\uparrow \subseteq \mathcal{P} \times \mathcal{P}$ is written as $p \uparrow q$ and denotes that perspective p *positively influences* perspective q.
- the binary relation $\downarrow \subseteq \mathcal{P} \times \mathcal{P}$ is written as $p \downarrow q$ and denotes that perspective p *negatively influences* perspective q.

If perspective p positively influences perspective q, then outcomes that are better from perspective p tend to be better from perspective q. In other words, if an outcome is better from p and p positively influences q, then this is a reason to believe that the outcome is better from q. For example, the size of a house positively influences the comfort of the house, i.e. the more size, the more comfort.

The following argument scheme reasons with influence: *outcome ω_2 is preferred to ω_1 from p and p positively influences q, therefore ω_2 is preferred to ω_1 from q.* In other words, if a perspective p positively influences perspective q, then an outcome being preferred from p is a reason to believe that that outcome is also preferred from q. We formalise this argument scheme with the following defeasible inference rule:

$$d_{<,\uparrow}(p, q, \omega_1, \omega_2) : \omega_1 <_p \omega_2, p \uparrow q \rightsquigarrow \omega_1 <_q \omega_2$$

The argument scheme to propagate negative influence is similar, except that if a perspective p negatively influences perspective q, then outcomes that are better from perspective p tend to be worse from perspective q. For example, costs negatively influences agent α's preferences, i.e. the more costs, the less preferable for α. This argument scheme is formalised with the following defeasible inference rule:

$$d_{<,\downarrow}(p, q, \omega_1, \omega_2) : \omega_1 <_p \omega_2, p \downarrow q \rightsquigarrow \omega_2 <_q \omega_1$$

Using these inference rules, arguments can be constructed to justify preferences over outcomes using both positive influence and negative influence between perspectives.

Example 7. Agent α wants to buy a new house and to minimise costs, i.e. $\mathsf{costs} \downarrow \alpha$ is true. There are two outcomes, ω_1 and ω_2, such that the acquisition price, denoted attribute acq, of ω_1 is \$200k and of ω_2 \$150k. The acquisition price of a house positively influences the costs, so $\mathsf{costs}_{\mathsf{acq}} \uparrow \mathsf{costs}$ is true.

$$\dfrac{\mathsf{costs} \downarrow \alpha \qquad \dfrac{\mathsf{costs}_{\mathsf{acq}} \uparrow \mathsf{costs} \qquad \dfrac{\overline{\mathsf{acq}}(\omega_2) \prec^{\mathsf{costs}}_{\mathsf{acq}} \overline{\mathsf{acq}}(\omega_1)}{\omega_2 <_{\mathsf{costs}_{\mathsf{acq}}} \omega_1}\ s_{ap}}{\omega_2 <_{\mathsf{costs}} \omega_1}\ d_{<,\uparrow}}{\omega_1 <_\alpha \omega_2}\ d_{<,\downarrow}$$

The relation $\uparrow$ is irreflexive and transitive, meaning that $p \uparrow p$ is never true and that if $p \uparrow q$ and $q \uparrow r$ are true, then $p \uparrow r$ is true. If $p \uparrow p$ would be true and for any two outcomes $\omega_1 <_p \omega_2$ is true, then we can defeasibly infer that $\omega_1 <_p \omega_2$ is true using inference rule $d_{<,\uparrow}$. Such an argument concludes one of its premises, which is useless. Furthermore, if $p \uparrow p$ can be true, then this may cause infinite loops in implementations. For this reason, the relation $\uparrow$ is irreflexive.

The relation $\uparrow$ should be transitive. Firstly, this is intuitive. For example, let the location of a house positively influence the fun of that house and let the fun of a house positively influence agent α's preferences. Then we can also say that the location of a house positively influences α's preferences. Secondly, this leads to inferences we could already infer. Namely, if $p \uparrow q$, $q \uparrow r$ and $\omega_1 <_p \omega_2$ are true, then we can defeasibly infer that $\omega_1 <_q \omega_2$ is true. Similarly, $\omega_1 <_r \omega_2$ can be defeasibly inferred from $q \uparrow r$ and $\omega_1 <_q \omega_2$. If $\uparrow$ is transitive, then $p \uparrow r$ is also true. From $p \uparrow r$ and $\omega_1 <_p \omega_2$ we can defeasibly infer that $\omega_1 <_r \omega_2$ is true.

The relation $\downarrow$ is irreflexive and antitransitive, meaning that $p \downarrow p$ is not allowed and that if $p \downarrow q$ and $q \downarrow r$ are true, then $p \downarrow r$ is not true. If for some perspective p it would be true that $p \downarrow p$ and for any two outcomes $\omega_1 <_p \omega_2$ is true, then the following argument A can be constructed.

$$A = \dfrac{p \downarrow p \quad \omega_1 <_p \omega_2}{\omega_2 <_p \omega_1}\ d_{<,\downarrow}$$

The conclusion of A conflicts with A's premise $\omega_1 <_p \omega_2$. Consequently, A attacks itself. Allowing $p \downarrow p$ to be true adds nothing useful and can only result in contradictions. Therefore, the relation $\downarrow$ is irreflexive.

The relation $\downarrow$ should be antitransitive. Firstly, this is intuitive. For example, the amount of discount on a house negatively influences the costs of that house (the more discount the less costs) and the costs of a house negatively influences agent α's preferences. Then we can also say that the amount of discount does not negatively influence α's preferences. Secondly, if $\downarrow$ could be transitive, then this could lead to false inferences. Let $p \downarrow q$, $q \downarrow r$ and $\omega_1 <_p \omega_2$ be true. From $p \downarrow q$ and $\omega_1 <_p \omega_2$ we can infer that $\omega_2 <_q \omega_1$ is true and from $\omega_2 <_q \omega_1$ and $q \downarrow r$ we can infer that $\omega_1 <_r \omega_2$ is true. If $p \downarrow r$ would be true, then we could infer that $\omega_2 <_r \omega_1$ is true, which conflicts with $\omega_1 <_r \omega_2$.

3.4 Argumentation Framework

Argumentation Frameworks (AF) were introduced by [6] and provide a formal means to determine what arguments are justified given a set of arguments and a set of attack relations between them. In [3], Value-based Argumentation Frameworks (VAF), an extension of AFs, were introduced.

Our definition of a Perspective-based Argumentation Framework (PerspAF) is largely based on the definition of a VAF (because of space limitations, we refer the reader to [3] for details of VAFs).

Definition 5. *A PerspAF is defined by a tuple* $\langle \mathsf{Args}, R, \mathcal{P}, \eta \rangle$*, where* Args *is the set of all arguments,* R *the attack relations between arguments,* $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$ *a set of* k *perspectives, and* $\eta : \mathsf{Args} \rightarrow 2^{\mathcal{P}}$ *a mapping that associates a set of perspectives with each argument in* Args*. A total ordering* $\trianglelefteq_\alpha$ *of* $\mathcal{P}$ *is associated to each agent* α *for a PerspAF* $\langle \mathsf{Args}, R, \mathcal{P}, \eta \rangle$*.*

In PerspAFs, arguments may use multiple perspectives, whereas arguments can only use a single value in VAFs. To determine whether an attack between argument A and B is successful, only the perspectives are used that A and B do not use both.

The following concepts are related to PerspAFs and are identical to the concepts of VAFs except for when an argument defeats or successfully attacks another argument for a specific audience. Let $\langle \mathsf{Args}, R, \mathcal{P}, \eta \rangle$ be a PerspAF and α an agent.

- argument A *α-defeats* argument B if $(A, B) \in R$ and if there is no perspective $p \in \eta(B) \setminus \eta(A)$ such that $q \lhd_\alpha p$ for each $q \in \eta(A) \setminus \eta(B)$.
- argument A is α-acceptable to the set of arguments $S \subseteq \mathsf{Args}$ if: for every $B \in \mathsf{Args}$ that successfully α-attacks A, there is some $C \in S$ that successfully α-attacks B
- a set $S \subseteq \mathsf{Args}$ is α-conflict-free if: for each $\langle A, B \rangle \in S \times S$, either $\langle A, B \rangle \notin R$ or $\eta(A) \lhd_\alpha \eta(B)$
- a set $S \subseteq X$ is α-admissible if: S is α-conflict-free and every $A \in S$ is α-acceptable to S
- a set is a preferred extension for α if it is maximal α-admissible set

Example 8. Let $\mathcal{P} = \{\mathsf{fun}, \mathsf{comfort}, \alpha\}$ and let audience α order these perspectives as follows: $\mathsf{comfort} \lhd_\alpha \mathsf{fun} \lhd_\alpha \alpha$. Consider the following two arguments:

$$A_f = \frac{\mathsf{fun} \uparrow \alpha \quad \omega_1 <_{\mathsf{fun}} \omega_2}{\omega_1 <_\alpha \omega_2} \qquad A_c = \frac{\mathsf{fun} \uparrow \alpha \quad \omega_2 <_{\mathsf{comfort}} \omega_1}{\omega_2 <_\alpha \omega_1}$$

Let $\langle H(\{A_f, A_c\}, R), \mathcal{P}, \eta \rangle$ be a PerspAF. The conclusions of both arguments conflict, and thus they attack each other and $R = \{(A_f, A_c), (A_c, A_f)\}$. Since argument A_f uses the perspectives α and fun, $\eta(A_f) = \{\alpha, \mathsf{fun}\}$. Similarly, $\eta(A_c) = \{\alpha, \mathsf{comfort}\}$. Consequently, $\eta(A_f) \setminus \eta(A_c) = \{\mathsf{fun}\}$ and $\eta(A_c) \setminus \eta(A_f) = \{\mathsf{comfort}\}$.

Argument A_c does not α-defeat A_f because A_f uses the perspective fun, not used by A_c, and fun is preferred to $\mathsf{comfort}$. On the other hand, argument A_f α-defeats A_c because $(A_f, A_c) \in R$ and there is no perspective used by A_c that is both not used in A_f and is preferred to every perspective in A_f not used in A_c. The set $\{A_f\}$ is the preferred extension for α.

4 Justification Of Goals

In this section, we propose how an agent α can justify having a goal given α's preferences. In [13], Simon views goals as threshold aspiration levels that signal satisfactory of utility. A goal thus does not have to be optimal. Following [16], we see goals as expressions of the desirability of attribute values of a single attribute signaling that these attribute values are 'satisfactory'. For example, an agent may have the goal to live in a house that is located downtown. This expresses that the attribute value 'downtown' of the attribute 'location' is satisfactory to that agent. Another attribute value, e.g. 'suburb', does not achieve that goal and is thus not satisfactory.

The predicate $\mathsf{goal}(\alpha, x, G)$ is introduced in $\mathcal{L}$ and denotes that agent α should have the goal to achieve an outcome that has an x-value in $G \subset \mathsf{domain}(x)$. If agent α has the goal to achieve an x-value in G (i.e. $\mathsf{goal}(\alpha, x, G)$ is true) and outcome ω_1 has an x-value in G, i.e. $\overline{x}(\omega_1) \in G$ is true, then we say that goal $\mathsf{goal}(\alpha, x, G)$ is achieved in outcome ω_1. Consequently, a subset of Ω achieves a goal and the other outcomes in Ω do not achieve that goal.

4.1 Justification Is Subjective

What justification for a goal an agent accepts, depends on the type of agent. For example, a very ambitious but realistic agent only accepts goals that aim for the best achievable x-value, whereas a less ambitious agent may accept goals that just improves the current situation or does better than doing nothing. Another agent may set its standard on a value that is realistic and challenging, i.e. not too easy and not too difficult.

We introduce two argument schemes to distinguish between satisficing goals and optimising goals. The following argument scheme justifies the goal to achieve an x-value that is the best possible. The basis for this justification is that agents should aim to achieve their maximal potential.

Agent α wants to maximise attribute x-values from perspective p_x,
v is most preferred x-value from p_x that is achievable,
therefore, α pursues the goal to achieve x-values of v or better from p_x

If the predicates $\max(\alpha, p_x, v)$ and $\min(\alpha, p_x, v)$ denote that v is the maximal / minimal x-value from p_x that α can achieve, then the optimistic goal argument scheme can be modelled with the following defeasible inference rules:

$$d_{\mathsf{optm},\uparrow}(\alpha, p_x, v) : p_x \uparrow \alpha, \max(\alpha, v, p_x) \rightsquigarrow \mathsf{goal}(\alpha, x, \{g \in \mathsf{domain}(x) \mid v \preceq^p_x g\})$$
$$d_{\mathsf{optm},\downarrow}(\alpha, p_x, v) : p_x \downarrow \alpha, \min(\alpha, v, p_x) \rightsquigarrow \mathsf{goal}(\alpha, x, \{g \in \mathsf{domain}(x) \mid g \preceq^p_x v\})$$

A possible undercutter of the optimistic argument scheme is that achieving the goal is too unlikely. Therefore, the agent should adopt the goal to achieve an easier x-value. Another undercutter would be that achieving v is too costly and that α does not care that much about p_x.

The following argument scheme justifies a goal in a satisficing manner. This scheme's underlying motivation is that agents should adopt goals that achieve outcomes that are satisfactory rather than the best outcome.

Agent α wants to maximise attribute x-values from perspective p_x,
v is a satisfactory and achievable x-value for α,
therefore, α pursues the goal to achieve x-values of v or better from p_x

A possible undercutter for the satisficing argument scheme is that it is too easy and that the agent should adopt a more challenging goal. Another undercutter could be that the perspective p_x is important to α and therefore α should set a higher goal.

Let the predicate $\mathsf{satisf}(\alpha, x, v)$ denote that x-value v is satisfactory for agent α. Then this argument scheme can be modelled with the following defeasible inference rule.

$$d_{\mathsf{satisf},\uparrow}(\alpha, p_x, v) : p_x \uparrow \alpha, \mathsf{satisf}(\alpha, x, v) \rightsquigarrow \mathsf{goal}(\alpha, x, \{g \in \mathsf{domain}(x) \mid v \preceq^p_x g\})$$
$$d_{\mathsf{satisf},\downarrow}(\alpha, p_x, v) : p_x \downarrow \alpha, \mathsf{satisf}(\alpha, x, v) \rightsquigarrow \mathsf{goal}(\alpha, x, \{g \in \mathsf{domain}(x) \mid g \preceq^p_x v\})$$

This only solves part of the problem because how can an agent justify that an attribute value is satisfactory? We can think of several justifications of a satisfaction level: anything better than the current situation is satisfactory, it is better than some standard action such as 'do nothing', it is better than what other agents achieve, or the agent is obliged to achieve at least v. This is however still an open issue that is left for future work.

4.2 Priorities Of Goals

In our PerspAFs, agents have a total ordering over perspectives that represents what perspective they find most important. This information can be used to give goals priorities. Namely, if α has goal G because of perspective p_x and goal H because of perspective q_y and α finds p_x more important than q_y, i.e. $q_y \lhd_\alpha p_x$, then goal H is more important to α.

Goals are created using an attribute perspective that influences an agent. For the same attribute perspective, optimistic goals are stricter than satisficer goals since they do not include satisfactory attribute values upon which the agent can improve. For this reason, achieving an optimistic goal should have a higher priority than achieving a satisficer goal for the same attribute perspective.

5 Buying A House

Agent α, who lives in a suburb, recently got a raise in income and wants to buy a new house to live in. The broker shows two houses that are for sale, one in a village and one downtown, represented with outcomes ω_v and ω_d respectively. Of course, α has the possibility not to buy a new house and stay in its current house. This is represented with outcome ω_0. Consequently, $\Omega = \{\omega_0, \omega_d, \omega_v\}$. Except for its own house, α is unfamiliar with these houses and can therefore not express whether it prefers one of the new houses to its own house.

The broker includes the following attributes of each house: the neighbourhood, the size, and the acquisition price. The attribute nbhd denotes the neighbourhood of the house and $\mathsf{domain}(\mathsf{nbhd}) = \{\mathsf{dwntwn}, \mathsf{sbrb}, \mathsf{vllg}\}$. The attribute area denotes how big the house is in square meters. Consequently, $\mathsf{domain}(\mathsf{area}) = \mathbb{R}^+$. The attribute acq denotes the price of the acquisition of the house and $\mathsf{domain}(\mathsf{acq}) = \mathbb{R}^+$. The set of all attributes is the following: $\mathcal{A} = \{\mathsf{nbhd}, \mathsf{area}, \mathsf{acq}\}$. The attribute values for each outcome can be found in Table 1.

Table 1: Attribute Values Of Outcomes

Attribute	Domain	ω_0	ω_d	ω_v
nbhd	$\{\mathsf{dwntwn}, \mathsf{sbrb}, \mathsf{vllg}\}$	sbrb	dwntwn	vllg
area	$\mathbb{R}^+$ in m^2	60	50	100
acq	$\mathbb{R}^+$ in \$1000	0	220	190

5.1 Decomposing Perspectives

Agent α starts reasoning about its preferences over Ω by expressing what aspects it finds important. Namely, α wants to minimise costs, maximise fun and maximise comfort. By doing so, α's perspective is decomposed into other perspectives that are more concrete. Because α wants to minimise costs, the perspective costs negatively influences the perspective of α, i.e. $\mathsf{costs}\downarrow\alpha$ is true. Also, α wants to maximise fun and comfort, so $\mathsf{fun}\uparrow\alpha$ and $\mathsf{comfort}\uparrow\alpha$ are true.

Agent α figures that the acquisition price attribute is appropriate to determine the perspective of costs such that the higher the acquisition price, the higher the costs. The attribute perspective $\mathsf{costs}_{\mathsf{acq}}$ prefers an acq-value if it is higher. Therefore, $\mathsf{costs}_{\mathsf{acq}}\uparrow\mathsf{costs}$ is true.

For α fun means having people around him. The centrality of a house positively influences fun since α is more likely to out for dinner or drinks with his friends. Therefore, α decomposes the perspective fun into the perspective of the centrality of the neighbourhood, denoted with the attribute perspective $\mathsf{cntrl}_{\mathsf{nbhd}}$ on the attribute nbhd. Consequently, $\mathsf{cntrl}_{\mathsf{nbhd}}\uparrow\mathsf{fun}$ is true.

There is however no attribute that α finds adequate to determine the perspective of comfort. Therefore, comfort is decomposed into the quietness around the house and its size. Size is measured by the attribute perspective $\mathsf{size}_{\mathsf{area}}$ that orders the attribute area (denoting the surface area in m^2) according to size. The attribute perspective $\mathsf{quiet}_{\mathsf{nbhd}}$ orders neighbourhoods by their quietness. Both attributes positively influence comfort, i.e. $\mathsf{size}_{\mathsf{area}} \uparrow \mathsf{comfort}$ and $\mathsf{quiet}_{\mathsf{nbhd}} \uparrow \mathsf{comfort}$ are true.

The attribute perspectives $\mathsf{cntrl}_{\mathsf{nbhd}}$ and $\mathsf{quiet}_{\mathsf{nbhd}}$ both order the attribute values of the attribute 'neighbourhood' and are as follows:

$$\mathsf{sbrb} \prec^{\mathsf{cntrl}}_{\mathsf{nbhd}} \mathsf{vllg} \prec^{\mathsf{cntrl}}_{\mathsf{nbhd}} \mathsf{dwntwn} \qquad\qquad \mathsf{dwntwn} \prec^{\mathsf{quiet}}_{\mathsf{nbhd}} \mathsf{sbrb} \prec^{\mathsf{quiet}}_{\mathsf{nbhd}} \mathsf{vllg}$$

5.2 Arguments About Preference

Now, α starts constructing arguments concerning its preferences over houses. The following argument concludes that α should prefer staying in its house, outcome ω_0, to buying the house downtown, ω_d, because the costs of not buying are lower.

$$A_{\mathsf{costs}} = \dfrac{\mathsf{costs} \downarrow \alpha \qquad \dfrac{\overline{\mathsf{acq}}(\omega_0) \prec^{\mathsf{costs}}_{\mathsf{acq}} \overline{\mathsf{acq}}(\omega_d)}{\omega_0 <_{\mathsf{costs}_{\mathsf{acq}}} \omega_d}\, s_{ap}}{\omega_d <_\alpha \omega_0}\, d_{\downarrow,<}$$

However, the following argument concludes that α should prefer ω_d, which conflicts with A_{costs}'s conclusion, because ω_d is more fun since it is located in a more central neighbourhood.

$$A_{\mathsf{fun}} = \dfrac{\mathsf{fun} \uparrow \alpha \qquad \dfrac{\mathsf{cntrl}_{\mathsf{nbhd}} \uparrow \mathsf{fun} \qquad \dfrac{\overline{\mathsf{nbhd}}(\omega_0) \prec^{\mathsf{cntrl}}_{\mathsf{nbhd}} \overline{\mathsf{nbhd}}(\omega_d)}{\omega_0 <_{\mathsf{cntrl}_{\mathsf{nbhd}}} \omega_d}\, s_{ap}}{\omega_0 <_{\mathsf{fun}} \omega_d}\, d_{\uparrow,<}}{\omega_0 <_\alpha \omega_d}\, d_{\uparrow,<}$$

Agent α keeps thinking and comes up with the following argument that concludes that its current house is actually more comfortable since it is in a neighbourhood that is more quiet.

$$A_{\mathsf{comfort}} = \dfrac{\mathsf{comfort} \uparrow \alpha \qquad \dfrac{\mathsf{quiet}_{\mathsf{nbhd}} \uparrow \mathsf{comfort} \qquad \dfrac{\overline{\mathsf{nbhd}}(\omega_d) \prec^{\mathsf{quiet}}_{\mathsf{nbhd}} \overline{\mathsf{nbhd}}(\omega_0)}{\omega_d <_{\mathsf{quiet}_{\mathsf{nbhd}}} \omega_0}\, s_{ap}}{\omega_d <_{\mathsf{comfort}} \omega_0}\, d_{\uparrow,<}}{\omega_d <_\alpha \omega_0}\, d_{\uparrow,<}$$

Given these three arguments, we want to determine what conclusions are justified. For this we construct the PerspAF $\langle H(\mathsf{Args}, R), \mathcal{P}, \eta\rangle$, with:

$$\begin{aligned}
\mathsf{Args} &= \{A_{\mathsf{costs}}, A_{\mathsf{fun}}, A_{\mathsf{comfort}}\}\\
R &= \{(A_{\mathsf{costs}}, A_{\mathsf{fun}}), (A_{\mathsf{fun}}, A_{\mathsf{costs}}), (A_{\mathsf{comfort}}, A_{\mathsf{fun}}), (A_{\mathsf{fun}}, A_{\mathsf{comfort}})\}\\
\mathcal{P} &= \{\alpha, \mathsf{fun}, \mathsf{comfort}, \mathsf{costs}, \mathsf{quiet}_{\mathsf{nbhd}}, \mathsf{cntrl}_{\mathsf{nbhd}}, \mathsf{costs}_{\mathsf{acq}}, \mathsf{size}_{\mathsf{area}}\}
\end{aligned}$$

and function η, that maps an argument to the perspectives it contains, is as follows.

$$\begin{aligned}\eta(A_{\mathsf{costs}}) &= \{\alpha, \mathsf{costs}, \mathsf{costs}_{\mathsf{acq}}\}\\ \eta(A_{\mathsf{fun}}) &= \{\alpha, \mathsf{fun}, \mathsf{cntrl}_{\mathsf{nbhd}}\}\\ \eta(A_{\mathsf{comfort}}) &= \{\alpha, \mathsf{comfort}, \mathsf{quiet}_{\mathsf{nbhd}}\}\end{aligned}$$

Let α find fun more important than $\mathsf{comfort}$ and costs more important than fun, i.e. $\mathsf{comfort} \lhd_\alpha \mathsf{fun}$ and $\mathsf{fun} \lhd_\alpha \mathsf{costs}$ are true. Then A_{fun} α-defeats A_{comfort} and A_{costs} α-defeats A_{fun}. Then the set $\{A_{\mathsf{comfort}}, A_{\mathsf{costs}}\}$ is the preferred extension, so the conclusion that α prefers staying in its current house to buying a house downtown is justified.

5.3 Goals

If α will also visit other brokers and thus considers more houses, it can be computationally efficient for α to generate a number of goals that can easily be checked when evaluating a new house. Given a number of goals, evaluating an outcome involves checking whether its attribute values are in the goals. If no goals are used, then evaluating an outcome involves constructing arguments for all relevant perspectives to check whether it is better than some other outcome(s).

The current house of α is $60m^2$, i.e. $\overline{\mathsf{area}}(\omega_0) = 60$, and α finds this size satisfactory. Since α does not feel very strongly about the size of its house, α uses the satisfycing argument scheme to justify the following goal.

$$\frac{\mathsf{size}_{\mathsf{area}} \uparrow \alpha \quad \mathsf{satisf}(\alpha, \mathsf{area}, 60)}{\mathsf{goal}(\alpha, \mathsf{area}, \{g \in \mathsf{domain}(\mathsf{area}) \mid g \geq 60\})}\ d_{\mathsf{satisf}}$$

With its new job, α can maximally lend 200 thousand dollar for the acquisition of a house and therefore α sets its aspiration level for the acquisition on 200. Given this information, α justifies having the following goal:

$$\frac{\mathsf{costs}_{\mathsf{acq}} \downarrow \alpha \quad \mathsf{satisf}(\alpha, \mathsf{acq}, 200)}{\mathsf{goal}(\alpha, \mathsf{acq}, \{g \in \mathsf{domain}(\mathsf{acq}) \mid g \leq 200\})}\ d_{\mathsf{satisf}}$$

The current house of α is in a suburb and α wants to maximise neighbourhood with respect to both centrality and quietness, i.e. $\mathsf{cntrl}_{\mathsf{nbhd}} \uparrow \alpha$ and $\mathsf{quiet}_{\mathsf{nbhd}} \uparrow \alpha$ are true. Agent α cares a lot about the centrality of its house and less about its quietness. Therefore, α uses the optimising argument scheme to justify its goal to live downtown:

$$\frac{\mathsf{cntrl}_{\mathsf{nbhd}} \uparrow \alpha \quad \max(\alpha, \mathsf{dwntwn}, \mathsf{cntrl}_{\mathsf{nbhd}})}{\mathsf{goal}(\alpha, \mathsf{nbhd}, \{\mathsf{dwntwn}\})}\ d_{\mathsf{optim}}$$

About the quietness α cares less and therefore uses the satisfycing argument scheme:

$$\frac{\mathsf{quiet}_{\mathsf{nbhd}} \uparrow \alpha \quad \mathsf{satisf}(\alpha, \mathsf{nbhd}, \mathsf{sbrb})}{\mathsf{goal}(\alpha, \mathsf{nbhd}, \{\mathsf{sbrb}, \mathsf{vllg}\})}$$

It is impossible for α to achieve both goals. However, α finds costs more important than fun and fun more important than comfort. Consequently, α finds the goal to live downtown more important than the goal to live in a quiet suburb.

6 Discussion

6.1 Outcomes Compared To States

In [3, 2], states are used to reason about decisions over actions, rather than outcomes. A state is a truth assignment to a set of propositions. In a state r, an agent can perform an action a, which results in another state s. If the agent performs a, there is a state transition from state r to state s.

In decision theory, making a decision results in an outcome. Outcomes represent all possible consequences of a decision. Outcomes can represent the state resulting from the action performed, effects in the far future, how pleasant the action was, and possibly the history of all preceding states. An outcome is thus a more general notion than a state, because outcomes can contain all information in states and even more.

6.2 Values Versus Perspectives

In [2], there is a valuation function δ that takes a state transition and a value and returns whether that state transition either promotes, demotes, or is neutral towards that value. More specifically, $\delta : S \times S \times V \to \{+, -, 0\}$ with S the set of states and V the set of values. Note that a state transition either promotes, demotes, or is neutral towards a value resembles Simon's simple valuation function, which values an outcome either as 'satisfactory', 'indifferent' or 'unsatisfactory'.

The valuation function must be specified for all state transitions and all values, which can become time consuming when the number of states or values increases. Namely, if there are n states and m values, then the valuation function must be specified for $m \cdot n^2$ different inputs. Furthermore, if two agents disagree about whether a state transition is promotes a value, e.g. whether performing an action promotes the value of fun, then they can only explain that that is the outcome of their valuation function. Since values typically are abstract, it is important to explain and discuss what a value means. This is not possible in the approach of [2].

In our approach, a value is represented with a perspective, which is associated with an ordering over outcomes. A perspective can be decomposed into other perspectives and a perspective can be associated with an attribute of outcomes. This allows agents to explain and argue why a transition or goal promotes one of their values. For example, an agent can explain that its value of 'fun' means maximising spending time with friends and minimising time at work. whereas another agent can then explain that to him fun means spending time in nature and accomplishing things at work.

Furthermore, decomposing an abstract perspective into more specific perspectives for which an ordering is more easy to specify, makes it less demanding to specify whether a transition promotes, demotes or is neutral towards a value.

If a perspective p represents a value, then its associated ordering $\leq_p$ can be used to the define the valuation function δ for p in the following way

$$\delta(q_1, q_2, v) = \begin{cases} + & \text{if } q_1 <_v q_2 \\ - & \text{if } q_2 <_v q_1 \\ 0 & \text{if } q_1 \equiv_v q_2 \end{cases}$$

If $\leq_v$ is a total order, i.e. no elements are incomparable, then δ is a normal function, otherwise δ is a partial function.

7 Conclusion

In this paper we have proposed several argument schemes to argue about what decision is best for an agent based on its preferences over outcomes. An agent's preferences are expressed in terms of values and goals and we propose a model to represent what a value means and how it affects an agent's preferences. If the meaning of a value is clear, goals can be justified or attacked by arguing that they promote or demote a value.

We represent values as perspectives over outcomes. By recursively decomposing the different aspects of a perspective into other perspectives until they are decomposed in attribute perspectives, the meaning of a perspective and thus a value is made explicit. In this way, an agent can explain what a value exactly means to him, which allows other agents to argue that some aspect is wrong or forgotten or that the wrong attribute is used. Agents can justify pursuing a goal using the perspectives that are important to an agent and the attributes that are associated to those perspectives. We have discussed a satisficing and a optimistic argument scheme to justify a goal. Furthermore, priorities between goals can be justified using the priorities agents have over perspectives.

In future work, the relation between values and goals may be explored further. Different agent types and different situations may lead to pursuing different goals. An optimistic goal may be undercut by stating that it is too hard to achieve, but when is a goal too hard to achieve? Moreover, how can an agent justify that an attribute value is satisfactory and how is that influenced by circumstances?

When an agent finds costs more important than the centrality of the neighbourhood, and a house in a suburb is \$1 cheaper than a house downtown, then the costs argument is stronger than the centrality argument. By extending the formalism in this paper with 'distances' between attribute values, such weird results might be solved.

Bibliography

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

[2] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.

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

[4] F. Bex, H. Prakken, C. Reed, and D. Walton. Towards a formal account of reasoning about evidence: Argumentation schemes and generalisations. *Artificial Intelligence and Law*, 11(2):125–165, 2003.

[5] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[6] 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.

[7] A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. *Proc. of 2nd Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 883–890, 2003.

[8] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives.* Wiley, New York, 1976.

[9] R.L. Keeney. *Value-Focused Thinking: A Path to Creative Decisionmaking.* Harvard University Press, 1992.

[10] M. Rokeach. *The nature of human values.* Free Press, New York, 1973.

[11] L.J. Savage. *The foundations of statistics.* Dover Pubns, 1972.

[12] SH Schwartz. Universals in the content and structure of values: theoretical advances and empirical tests in 20 countries. *Advances in experimental social psychology*, 25:1–65, 1992.

[13] Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, pages 99–118, 1955.

[14] D.N. Walton. *Argumentation Schemes for Presumptive Reasoning.* Lawrence Erlbaum Associates, 1996.

[15] T. L. van der Weide, F. Dignum, J.-J. Ch. Meyer, and G. A. W. Prakken, H. Vreeswijk. Practical reasoning using values. In P. McBurney, I. Rahwan, S. Parsons, and P. Moraitis, editors, *Proceedings of the Sixth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2009), Budapest, Hungary*, pages 225–240, 2009.

[16] M.P. Wellman and J. Doyle. Preferential semantics for goals. *Proceedings of the National Conference on Artificial Intelligence*, pages 698–703, 1991.