University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Alina Powała

Modeling Dialogues in Multiagent Systems: a Paraconsistent Approach

PhD dissertation

Supervisor
prof. dr hab. Barbara Dunin-Kęplicz

Institute of Mathematics
University of Warsaw

March 2016

Author's declaration:
aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

March 23, 2016 ....
date *Alina Powała*

Supervisor's declaration:
the dissertation is ready to be reviewed

March 23, 2016 ....
date *prof. dr hab. Barbara Dunin-Kęplicz*

Acknowledgements

First of all I would like to thank my supervisor prof. Barbara Dunin-Kęplicz for patiently directing my work, for her time and invaluable advice during the whole of my doctoral studies, as well as for co-authoring all articles contained in this dissertation.

I would also like to thank prof. Andrzej Szałas for his insightful comments and discussions, which greatly influenced this work.

Special thanks go to my Husband, Mom, Dad and Grandpa for their endless support, understanding and encouragement.

I would also like to thank colleagues from companies NuTech Solutions, Netezza, IBM and iQor for fruitful discussions, sharing of experience and showing understanding at the last stages of work on this dissertation.

Finally I would like to thank also my friends, including Graham Richardson and Rick Handt for their help in preparing the English version, and, last but not least, Marysia Przyłucka and Irmina Jakubiak for the fact that I can always count on you.

Modeling Dialogues in Multiagent Systems: a Paraconsistent Approach

PhD dissertation summary

Alina Powała (formerly Strachocka)

March 23, 2016

Supervisor
prof. dr hab. Barbara Dunin-Kęplicz
Institute of Mathematics
University of Warsaw

Contents

1 Communication in Information-Rich Multiagent Settings

1.1 Multiagent Settings

This research should be considered in the context of collaboration in autonomous systems like *multiagent systems (MAS)*. An *intelligent agent* is a software program, capable of carrying out actions in its environment via *actuators*, upon perceiving percepts from that environment via *sensors*, in order to complete its objectives. Baring that simple definition, the *perception*, *reasoning*, *communication* and *action*, are the four main functions an agent must implement. In resource-bounded applications, these functions naturally compete for time, computational power, memory, bandwidth, etc. Since some objectives are difficult to achieve single-handedly, multiagent systems are created to exercise the synergy of agents' collaboration. This complex subject [16,35,52,73,97,99] inevitably faces balancing reasoning with communication. Obtaining this balance is the focus of this research. Another essential aspect of collaboration is resolving discrepancies arising from the clash of individual agents' opinions and objectives. Agent theories should acknowledge such emerging conflicts by providing agents with tools to resolve them. Conflict and its different facets in communication is the leitmotif of this dissertation.

Typically, regardless architecture type, an agent maintains:

- an **informational stance**, reflecting its view of the environment, other agents and itself,
- a **motivational stance**, which concerns agent's objectives and supporting motivations.

In multiagent literature informational stance is expressed in terms of *beliefs* (underlining the contrast with sure, provable and invariable *knowledge*) while motivational stance typically encompasses notions such as *intentions*, *goals* or *commitments*. Indeed, the most studied BDI (beliefs, desires, intentions) model of agency [20, 47, 72] originated from Bratman's human rational choice and action theory [11]. As (multi)modal logics allowed for expressive modeling of intensional concepts (e.g., belief, knowledge, necessity or possibility), they became a natural tool for formalizing individual and group informational (e.g., *common belief* [43]) and motivational (e.g., *social/collective intention*, *social/collective commitments* [35]) notions, providing a deep understanding of phenomena occurring in agency. However, this came at a price of high complexity [41]. Contemporary real-world applications call for theories which can be executed directly. This dissertation fulfills this requirement providing expressive yet tractable solutions for communication in multiagent settings together with an adequate methodology.

1.2 Research Objectives

Most multiagent environments are dynamic and unpredictable because sure, up-to-date and unambiguous data is seldom available. Agents situated in these environments receive information from multiple sources of varying significance, credibility or quality which results in an *incompleteness* and *inconsistency* of their beliefs. These factors demand modeling agents' belief bases as *paraconsistent* (i.e., tolerating inconsistency) and *paracomplete* (i.e., tolerating lack of information) ones. Thus, in *realistic modeling* agents are supposed to:

- reason with and communicate about contradictory and missing information,
- deal with inconsistent and incomplete conclusions (as opposed to terminating reasoning upon obtaining such conclusions),
- apply relevant nonmonotonic techniques like *default* or *autoepistemic reasoning*, *circumscription*, *(Local) Closed World Assumption ((L)CWA)* to fill the missing or disambiguate inconsistent information.

Since complex communication patterns are essential in agency, we relinquish rigid communication protocols, and lean towards more relaxed communication forms, acknowledging agents' flexibility. An adequate model of agents' reasoning is the subject of ongoing research at Institute of Informatics at University of Warsaw, led by professors Barbara Dunin-Kęplicz and Andrzej Szałas (see Section 2.2). Drawing upon their solutions, the overall objective of this dissertation was a *creation of a paraconsistent, paracomplete, dynamic and tractable formal model of communication* including

- G1. a formal model of communication forms:
  - G1.1. *elementary* (known as *speech-acts* [3, 76]), concerning beliefs [39],
  - G1.2. *advanced*, concerning reasoning schemes [28, 29],
- G2. formalization of *inquiry* as a dialogue type aiming at *knowledge acquisition* [37],
- G3. formalization of *persuasion* as a dialogue type aiming at *conflict resolution* [36].

Persuasion and *inquiry* have been selected as the two most vital dialogues in multiagent applications. The essence of this research is a deep analysis of complex phenomena appearing in these dialogues, leading to their *dynamic models* and the *complexity* results. **Importantly, the tractability requirement guided our approach.** Indeed, in our research the focus has changed from theoretical modeling of multiagent systems in multimodal logics of high complexity [35][1] to employing tractable rule-based approach suitable for practical applications. In Sections 3.3.5 and 3.4.4 both communication and computational complexity results are summed up.

The overall outcome of this research is a **methodology of complex yet computationally-friendly dialogues in information-rich settings that can be directly implemented as a part of multiagent application**. Moreover, the dissertation provides foundation for investigating other dialogue types and verification of their properties.

This dissertation is composed of the following five papers:

- **P1**. **Paraconsistent Semantics of Speech Acts** [39], published in *Neurocomputing*, with B. Dunin-Kęplicz, A. Szałas and R. Verbrugge, 2015.
- **P2**. **Perceiving Rules under Incomplete and Inconsistent Information** [28], published in the *Proceedings of 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA)*, with B. Dunin-Kęplicz, 2013.
- **P3**. **Paraconsistent Argumentation Schemes** [38], published in *Web Intelligence*, with B. Dunin-Kęplicz, 2016.
- **P4**. **Tractable Inquiry in Information-Rich Environments** [37], published in the *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, with B. Dunin-Kęplicz, 2015.
- **P5**. **Paraconsistent Multi-party Persuasion in** `TalkLOG` [36], published in the *Proceedings of the 18th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, with B. Dunin-Kęplicz, 2015.

Although the name `TalkLOG` has been introduced only in the last paper **P5**, we will use it to relate to the whole framework.

---

[1] It was also the context of my Master's thesis and paper [31].

My other publications include:

1. **Deliberation Dialogues during Multiagent Planning** [31], published in the *Proceedings of the 19th International Symposium on Foundations of Intelligent Systems (ISMIS)*, with B. Dunin-Kęplicz, R. Verbrugge, 2011.
2. **Perceiving Speech Acts under Incomplete and Inconsistent Information** [30], published in the *Proceedings of the International KES Conference on Agents Mutli-Agent Systems: Technologies and Applications (KES-AMSTA)*, with B. Dunin-Kęplicz, A. Szałas, R. Verbrugge, 2013.
3. **Computationally-Friendly Argumentation Schemes** [29], published in the *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, with B. Dunin-Kęplicz, 2014.
4. **The Polish School of Argumentation: A Manifesto** [12], published in *Argumentation*, with K. Budzyńska, B. Dunin-Kęplicz, et al., 2014.
5. **Multi-Party Persuasion: a Paraconsistent Approach**, submitted to *Fundamenta Informaticae*, with B. Dunin-Kęplicz, 2016.

2 Underpinnings

2.1 Theory of Speech-Acts and Theory of Dialogue

Contemporary approaches to communication in multiagent systems draw upon Walton and Krabbe's semi-formal theory of dialogue [94], adapting the normative models of human communication, including paradigmatic *dialogue types* (*inquiry*, *information seeking*, *deliberation*, *persuasion* and *negotiation*) to multiagent settings. See [2,6,8,13,18,25,31,35,57,60,63,64,67,74,90] for investigations in multiagent argumentation-based dialogue, and [94] for the definitions of dialogue types. Each model of dialogue is defined by its initial situation, the participants' individual goals, and the aim of the dialogue as a whole (see Table 1).

Table 1: Types of dialogue recalled from [92]

Type of Dialogue	**Initial Situation**	**Participants' Goal**	**Goal of Dialogue**
Persuasion	Conflict of Opinions	Persuade Other Party	Resolve or Clarify Issue
Inquiry	Need to Have Proof	Find and Verify Evidence	Prove (Disprove) Hypothesis
Negotiation	Conflict of Interests	Get What You Most Want	Reasonable Settlement Both Can Live With
Information Seeking	Need Information	Acquire or Give Information	Exchange Information
Deliberation	Dilemma or Practical	Choice Coordinate Goals and Actions	Decide Best Available Course of Actions
Eristics	Personal Conflict	Verbally Hit Out at Opponent	Reveal Deeper Basis of Conflict

In collaborative efforts [98], such as teamwork [35], agents need to communicate intensively starting from *team formation*, through *social planning* to *team action* and *reconfiguration*. At every stage, a different dialogue type prevails, e.g., *information seeking* during potential recognition, *persuasion* during team formation and action allocation or *deliberation* during means-end-analysis.

Complex dialogues are composed with the use of *speech acts* – the basic building blocks of communication. Contemporary understanding of speech acts comes form the works of Austin and Searle [3, 76] including the most popular taxonomy of speech acts, identifying:

- *assertives*, committing to the truth of a proposition, e.g., stating,
- *directives*, which get the hearer to do something, e.g., asking,
- *commissives*, committing the speaker to some future action, e.g., promising,
- *expressives*, expressing a psychological state, e.g., thanking,
- *declaratives*, changing reality according to the proposition e.g., baptising.

Importantly, Austin specified the *effects* of speech acts on the *attitudes* and *actions* of the hearer. Thus, in computational approach, **speech acts** are viewed as *actions "that make you change your mind"* [87], and **dialogues** are viewed as *communicative games* between two or more agents, who try to *expand*, *contract*, *update*, and *revise* their beliefs through communication. Indeed, the application of speech acts and dialogue theory to communication in multiagent systems dates back to late 20th century [19]. The semantics of dialogues (games) and speech acts (moves in the game) was studied in multiagent literature mainly from two angles:

- **mentalistic**: based on modeling changes in agents' internal mental attitudes. Speech acts as *typical actions* are defined by their *pre-* and *post-conditions* [21, 39, 44, 45, 51].
- **social**: departing from internal mental attitudes, building upon concepts such as *commitment* or *convention*, aiming at obtaining verifiable dialogue protocols [77, 78] (see also [17] and references therein).

Jason [9], an implementation of the AgentSpeak [71] language based on social commitments [78], is an example of combining together the mentalistic and societal view of agency.

In this dissertation both approaches have been explored. The initial works (**P1**, **P2**) were conducted in the mentalistic spirit. As mentalistic semantics does not lend itself to verification, when defining complex dialogues we followed the social approach. In principle we adopted the blackboard metaphor where dialogue participants are aware of both moves made in the dialogue and dialogue rules. To this end, in TalkLOG we exercised an architecture including:

- *dialogue stores*, persisting the effects of participants' moves on the state of the discussion,
- a *move relevance function*, relying only on the publicly available information.

2.1.1 Bi-Party Dialogues

Both inquiry and persuasion have enjoyed much attention from the researchers:

- Walton and Krabbe [94] introduced two types of semi-formal persuasion dialogues: *permissive persuasion* (PPD - everyday conversations) and *rigorous persuasion* (RPD - model of reasoned argument). In these dialogues each player's move has to pertain to the adversary's preceding move, so replies cannot be postponed. Therefore these dialogues do not offer a more nuanced handling of the burden of proof, which is vital for flexibility of interlocutors.
- PWA (Parsons, Wooldridge & Amgoud) protocol [64], although suffered from similar modeling limitations (see [67] for a discussion) was a formal approach (unlike [94]) allowing to analyze the properties of termination and dialogue outcomes.
- In his system [66], Prakken first allowed alternative replies and postponing replies, permitting much flexibility in persuasion. As regards conflict resolution, in [64] it hinged on the preference relation between arguments, while in [66] on the priorities of reasoning rules.

- In [7] a framework for representing dialogues was given, together with models of two subtypes of inquiry dialogues and a strategy for making moves in the dialogue. The authors used Defeasible Logic Programming (DeLP) to deal with ignorance and inconsistency in agents' knowledge bases.

Only recently, Walton [54, 93] signalized the importance of the inconsistent information and ignorance for the theory of dialogue. Currently there is a number of formalisms that do not trivialize when inconsistent premises (for a survey see [5, 95]). In [80] the logic of multi-valued argumentation (LMA) is used and agents argue using multi-valued knowledge base. In [68] ASPIC+, a framework for structured argumentation with possibly inconsistent knowledge bases and defeasible rules is given, while in [18] ALIAS agents use abductive reasoning to negotiate over incomplete knowledge. However, none of these formalisms handles inconsistency and ignorance the way `TalkLOG`'s underlying formalism does.

2.1.2 Multi-Party Dialogues

As indicated in [26, 82], several new issues arise when contemplating the plurality of dialogue participants. In two landmark papers [26, 82], Dignum and Vreesijk, and Traum et al. discuss related issues, like:

1. **Open vs. Closed System**: are all parties permanently present during the whole dialogue?
2. What are the **roles** of dialogue participants, are they fixed?
3. **Medium and Addressing**: one-to-one, one-to-many, one-to-all communication channels.
4. **Coordination**: turn-taking or asynchronous.
5. **Termination**: who decides?
6. **Properties**: e.g., comparing multi-party dialogue conclusions to those obtained by the union of participants' belief bases.
7. **Internal properties** of agents participating in such a dialogue.

Multi-party dialogues is a fairly young research field in Argumentation and Computational Dialectics [8, 26, 49, 90, 96, 100] and Intelligent Virtual Agents (IVA) [83, 83, 84] domains:

- In [90] a simple multi-party inquiry dialogue has been proposed, where the participants are equivalent (without roles) and exchange public messages via forum. Communication happens in turns with no termination criterion.
- In [8] agents share the set of arguments, but differ on the attack relations. Although agents were privately assigned to the two adverse groups, they independently proposed moves to the central authority who selected the move to play. Some of the investigated conditions include: what outcomes will be reached if agents follow the protocol; under which conditions the debate is pre-determined; and whether the outcome coincides with the result obtained by merging the argumentation systems of the participants.
- In [49] two protocols for regulating debates among agents based on a bi-polar argumentation (i.e., using two types or relations between arguments: *defeat* and *support*) are proposed: so-called *category-based* and *cluster-based*. Arguments were exchanged via a common game board. Comparisons to merged argumentation systems were studied.
- In [96] a framework for resolving disputes concerning categorisation of particular cases with the use of multi-party dialogues was given along with a discussion of mechanisms and strategies used to facilitate multi-party argumentation.

- In [100] a distributed argumentation system was given together with a multi-party dialogue game for computing the defensibility of an argument from consistent knowledge bases.
- In [83, 84] a model of negotiation for virtual agents, including negotiators with different goals, negotiating over multiple options, has been proposed. Agents can dynamically change their negotiating strategies based on the course of negotiation.

However, to our best knowledge, no complexity results for multi-party dialogues are given in the literature. Most of the current research regards simple protocols with limitations concerning both interlocutors (e.g., all are equivalent, with consistent belief bases) and dialogue mechanisms (e.g., turn taking, no termination, usually one-to-all communication channel).

2.2 Logical Foundations of `TalkLOG`

To model phenomena such as lack and inconsistency of information, a commonly used logic is Belnap's four-valued logic [4], but it turned out that in multiagent settings it sometimes provides unintuitive results (see e.g., [27]).[2] The logic system underlying the realistic model of agency we employed, encompasses and naturally treats unknown and inconsistent information and does not share such problems (for our survey of nonmonotonic and paraconsistent techniques see **P3**).

2.2.1 Logical Language

In `TalkLOG` the solution is founded on the four-valued logic of [89], equipped with two new truth values: unknown ($\mathfrak{u}$) and inconsistent ($\mathfrak{i}$) with the intuitive reading:

- a is *true* ($\mathfrak{t}$) if all sources claim a,
- a is *false* ($\mathfrak{f}$) if all sources claim $\neg a$,
- a is *unknown* ($\mathfrak{u}$) if no sources claim a nor $\neg a$,
- a is *inconsistent* ($\mathfrak{i}$) if some sources claim a, other claim $\neg a$.

The semantics of propositional connectives is summarized in Table 2. The definitions of $\wedge$ and $\vee$ reflect minimum and maximum with respect to the truth ordering

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}. \tag{1}$$

The employed truth ordering 1 illustrates the degrees of truth of a proposition. While $\mathfrak{f}$ expresses no possibility of the proposition being true, $\mathfrak{u}$ admits some possibility, $\mathfrak{i}$ expresses there is at least one witness of the truth of the proposition and $\mathfrak{t}$ means that the proposition is definitely true. Whenever truth values are restricted to $\{\mathfrak{f}, \mathfrak{t}\}$, the semantics is compatible with the semantics of classical first-order logic.

In what follows all sets are finite except for sets of formulas. We deal with the classical first-order language over a given vocabulary without function symbols. We assume that *Const*

---

[2] Example recalled from [89]. Assume a family owns two cars: a and b. The question, whether the family has a safe car corresponds to the logical value of the expression $safe(a) \vee safe(b)$. Car a has gone through safety tests at two different stations s_1 and s_2. It has passed the safety tests at s_1 but failed the tests at s_2. Car b has not gone through any safety test yet. The results of the tests determine the truth values of $safe(a)$ and $safe(b)$: $safe(a)$ has the value $\mathfrak{i}$ while $safe(b)$ has the value $\mathfrak{u}$. If the join operation $\vee$ is defined by Belnap's truth ordering, then $safe(a) \vee safe(b) = \mathfrak{i} \vee \mathfrak{u} = \mathfrak{t}$. However, the safety of car a is unclear, since the results of both safety tests are contradictory, and we know nothing about safety of car b! A more intuitive result here would be $\mathfrak{i}$. Asking instead, if all cars of the family are safe, $safe(a) \wedge safe(b)$, evaluates to $\mathfrak{f}$ in Belnap's logic ($\mathfrak{i} \wedge \mathfrak{u}$). However, actually we do not have any information about the safety of car b. If in reality it would have failed the safety tests then the expression above would evaluate to $\mathfrak{f}$. But, if car b would have passed the tests then the expression would become $\mathfrak{i}$. Therefore, the above case seems to be better described by $\mathfrak{u}$ than by the answer obtained in the Belnap's logic.

Table 2: Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$.

$\wedge$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$
$\mathfrak{u}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$

$\vee$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$

$\rightarrow$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{f}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$

	$\neg$
$\mathfrak{f}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$

is a fixed set of constants, Var is a fixed set of variables and Rel is a fixed set of relation symbols. Though we use classical first-order syntax, the semantics substantially differs from the classical one as truth values $\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}$ (true, inconsistent, unknown, false) are explicitly present. The semantics (see Def. 2) is based on sets of ground literals rather than on relational structures.

Definition 1 A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, $\bar{\tau}$ being a sequence of parameters, $\bar{\tau} \in (Const \cup Var)^k$, where k is the arity of $R \in Rel$. *Ground literals over Const*, denoted by $\mathcal{G}(Const)$, are literals without variables, with all constants in *Const*. If $\ell = \neg R(\bar{\tau})$ then $\neg\ell \stackrel{\text{def}}{=} R(\bar{\tau})$. $\triangleleft$

Let $v : Var \rightarrow Const$ be a *valuation of variables*. For a literal ℓ, by $\ell(v)$ we mean the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$.

Definition 2 The *truth value* $\ell(L, v)$ of a literal ℓ w.r.t. a set of ground literals L and valuation v, is defined by:

$$\ell(L,v) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$

$\triangleleft$

For a formula $\alpha(x)$ with a free variable x and $c \in Const$, by $\alpha(x)^x_c$ we understand the formula obtained from α by substituting all free occurrences of x by c.

Table 3: Semantics of first-order formulas.

- if α is a literal then $\alpha(L, v)$ is defined in Definition 2;
- $(\neg\alpha)(L, v) \stackrel{\text{def}}{=} \neg(\alpha(L, v))$;
- $(\alpha \circ \beta)(L, v) \stackrel{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x\alpha(x))(L, v) = \min\limits_{a \in Const} (\alpha^x_a)(L, v)$, where min is the minimum w.r.t. ordering (1);
- $(\exists x\alpha(x))(L, v) = \max\limits_{a \in Const} (\alpha^x_a)(L, v)$, where max is the maximum w.r.t. ordering (1).

Definition 2 is extended to all formulas in Tab. 3, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Tab. 2 w.r.t the truth ordering 1.

2.2.2 Modeling Agents

The current line of research follows an important shift in perspective proposed by Dunin-Kęplicz and Szałas: *rather than drawing conclusions from complex modal theories we reason from paraconsistent belief bases*. In this approach, the way an individual agent deals with conflicting or lacking information is encoded in its *epistemic profile* [32]. This concept embodies agent's reasoning capabilities encompassing techniques suitable for different aspects of agent's activities. Technically speaking, agents' *beliefs* are represented as sets of literals constituting paraconsistent belief bases. Epistemic profiles are represented as agent-specific *rules* operating on possibly complex *belief structures* in order to draw individual conclusions.

The definition of an epistemic profile is recalled from [33]. If S is a set, then $FIN(S)$ represents the set of all finite subsets of S.

Definition 3 Let $\mathbb{C} \stackrel{\text{def}}{=} \textsc{Fin}(\mathcal{G}(\mathit{Const}))$ be the set of all finite sets of ground literals over constants in *Const*. Then:

- a *constituent* is any set $C \in \mathbb{C}$;
- an *epistemic profile* is any function $\mathcal{E} : \textsc{Fin}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over epistemic profile* $\mathcal{E}$ is meant a structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$; here $\mathcal{C} \subseteq \mathbb{C}$ is a nonempty set of constituents and $F \stackrel{\text{def}}{=} \mathcal{E}(\mathcal{C})$ is the *consequent* of $\mathcal{B}^{\mathcal{E}}$. $\triangleleft$

The constituents and consequents reflect the processes of agents' belief acquisition and formation. An agent starts with constituents, i.e., sets of beliefs acquired by perception, expert-supplied knowledge, communication with other agents, and many other ways. Next, the constituents are transformed into consequents according to the agent's individual epistemic profile. Consequents contain final, "mature" beliefs (see Figure 1). More formally, an epistemic profile corresponds to a function mapping finite sets of ground literals to ground literals. Therefore, the epistemic profile, being any function, can encode any reasoning schema (especially when we disregard complexity issues). This dissertation aims at tractable solutions, thus complexity will matter.

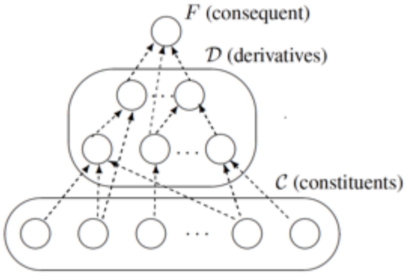

Figure 1: Belief formation (from [34] with permission).

The above definition of belief structures leads to a very important property: epistemic profiles can be devised analogously for individuals and groups, ensuring a uniform treatment of individual and group beliefs and thus facilitating reasoning in groups of complex topology. Namely, in multiagent settings, for each group, the *group epistemic profile* is set up, where consequents of group members become constituents at the group level and such constituents are further transformed into group consequents. This way, various perspectives of agents involved are taken into consideration and merged.

2.3 4QL as Implementation Tool

Many important aspects of classical agency have to be adjusted when adopting a paraconsistent semantics. First of all, the AGM postulates for Belief Revision [14] are no longer valid[3]. On the other hand, some assumptions underlying classical formalizations of agency (Dynamic Epistemic Logic [85], intention logic [20]; BDI [47] & KARO [86] frameworks) are unpractical: real agents do not have infinite resources (like time) available for reasoning and they are not logically perfect reasoners. Therefore a formalism which does not require such assumptions would be preferred.

Our approach is strongly influenced by ideas underlying 4QL: a four-valued, rule-based, paraconsistent query language introduced by Małuszyński and Szałas. 4QL[4] was defined in [56], further developed in [79] and based on a 4-valued logic of [89]. It allows for negation both in premises and conclusions of rules. Even though openness of the world is assumed, rules can be used to close the world locally or globally. 4QL features:

- Possibly many, perhaps distributed information sources.
- Four logical values ($\mathfrak{t}$, $\mathfrak{f}$, $\mathfrak{i}$, $\mathfrak{u}$).
- Unrestricted negation (in premises and conclusions of rules).
- Simple tools: rules, modules and multi-source formulas to formulate and enrich (lightweight versions of) (L)CWA, autoepistemic, default, or defeasible reasoning.
- Modular architecture to deal with unknown or inconsistent conclusions without enforcing termination of reasoning.
- PTIME complexity of computing queries while capturing all tractable queries.

The formal language underlying 4QL was introduced in Section 2.2.1.

Definition 4 A *multisource formula* is an expression of the form: $m.A$ or $m.A \in T$, where:

- m is a module name;
- A is a first-order or a multisource formula;
- v is a valuation;
- $T \subseteq \{\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}\}$.

We write $m.A = v$ (resp., $m.A \neq v$) to stand for $m.A \in \{v\}$ (resp., $m.A \notin \{v\}$). ◁

The intuitive meaning of a multisource formula $m.A$ is: *"return the answer to query expressed by formula A, computed within the context of module m"*. The value of '$m.A \in T$ ' is:

$$\begin{cases} \mathfrak{t} & \text{when the truth value of } A \text{ in } m \text{ is in the set } T; \\ \mathfrak{f} & otherwise. \end{cases}$$

Let $A(X_1, \ldots, X_k)$ be a multisource formula, $X_1, \ldots, X_k$ be its all free variables and D be a finite set of literals (a belief base). Then A, understood as a query, returns tuples $\langle d_1, \ldots, d_k, tv \rangle$, where $d_1, \ldots, d_k$ are database domain elements and the value of $A(d_1, \ldots, d_k)$ in D is tv.

Definition 5

- *Rules* are expressions of the form: ℓ :- $b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$. where the *premises* $b_{11}, \ldots, b_{1i_1}, \ldots, b_{m1}, \ldots, b_{mi_m}$ are multisource formulas and the *conclusion* ℓ is a positive or negative literal and ',' and '$|$' abbreviate conjunction and disjunction, respectively. If δ is a rule, by $head(\delta)$ we mean the rule conclusion.

---

[3] But see [70, 81].

[4] Open-source implementation of 4QL is available at `4ql.org`.

- A *fact* is a rule with empty premises (evaluated to $\mathbf{t}$). If δ is a fact, $head(\delta) = \delta$.
- A *module* is a syntactic entity encapsulating a finite number of facts and rules.
- A 4QL *program* is a set of modules, without cyclic references to modules involving multisource formulas of the form $m.A \in T$. ◁

In the sequel, Γ denotes the set of all facts; Π denotes the set of all rules.

The semantics of 4QL is defined by *well-supported models*, i.e., models consisting of (positive or negative) ground literals, where each literal is a conclusion of a derivation starting from facts. Each module can be treated as a finite set of literals. For any set of rules, such a set is uniquely determined and computable in deterministic polynomial time $O(N^k)$ where N is the size of domain and $k = max(s, t)$, where s is the maximal arity of relations and t is the maximum number of free variables.

Thanks to the correspondence between 4QL models and finite sets of literals, and due to the fact that 4QL captures PTIME [55], the constituents and consequents of Definition 3 being PTIME-computable, can be directly implemented as 4QL modules [34].

The complexity of 4QL is of great importance for this dissertation. On account of 4QL, TalkLOG dialogues are both expressive (as 4QL captures all tractable queries) and feasible (as 4QL enjoys polynomial computational complexity of computing queries). Such features distinguish 4QL from other formalisms, e.g., Answer Set Programming (ASP) [46]. ASP is based on the trivalent semantics ($true$, $false$, $unknown$), and does not admit inconsistency. Computing a so-called *"answer set"* (stable model) is NP-complete. The answer sets may contain conclusions that are not grounded in facts, which may be suitable for ASP primary applications (specification and computation of problems from the NP class), however is not appropriate in our case.

Definition 6 Let P be a 4QL program, A a formula, and $\mathcal{M}_P$ the well-supported (unique) model of P. Then, $P \models A$ iff for any valuation v we have $\mathcal{M}_P \models v(A)$.

Definition 7 Let ℓ be a literal and P a 4QL program. The *derivation* of ℓ from P is the well-supported model $\mathcal{M}_P$.

Example 1 *Consider program $P = \{top, su\}$ consisting of two modules top and su*[5].

$$\begin{array}{rrcl} top = \{ & enter(b) & \text{:-} & isAt(s,b), \neg has(s,h)., \\ & isAt(s,b) & \text{:-} & isArmed(s), hearShotsAt(b)., \\ & isAt(s,b) & \text{:-} & su.isAt(s,b) \in \{\mathfrak{u}, \mathfrak{i}, \mathbf{t}\}., \\ & & & \neg has(s,h), \\ & & & has(s,h), \\ & & & isArmed(s)\} \\ su = \{ & isAt(s,b) & \text{:-} & see(s,b), \neg conditions(fog)., \\ & & & see(s,b), \\ & & & \neg conditions(fog)\} \end{array} \tag{2}$$

The literals s, b, h represent suspect, building and hostage, respectively. The program uniquely determines the following well-supported model for module su:

$$\mathcal{M}_{su} = \{\neg conditions(fog), see(s,b), isAt(s,b)\} \tag{3}$$

and the following well-supported model for module top:

$$\begin{array}{l} \mathcal{M}_{top} = \{enter(b), \neg enter(b), isAt(s,b), \\ isArmed(s), has(s,h), \neg has(s,h)\}. \end{array} \tag{4}$$

---

[5] su stands for 'surveillance'.

3 Results Obtained in the Dissertation

3.1 Conversing Agents in 4QL

Argumentation schemes [95], originating from Legal Argumentation, attempt to classify different types of everyday arguments, utilizing the ideas underlying nonmonotonic formalisms. Each scheme is accompanied by a set of *critical questions*, used to evaluate the argument. Although particular schemes may *represent* different types of reasoning (e.g., deduction, induction, abduction, presumption), in general they aim to model plausible, thus defeasible, reasoning.

Table 4: Original Expert Opinion Scheme [95]

	Original Argumentation Scheme
Scheme	A is an expert in domain D A asserts that X is true X is within D
	X is true
Critical Questions	How credible is A as an expert source? Is A an expert in domain D? What did A assert that implies X? Is A personally reliable as a source? Is X consistent with what other experts assert? Is A's assertion of X based on evidence?

In principle, heterogeneity of agents w.r.t. reasoning means that when presented with the same evidence, agents may draw different conclusions. The notion of epistemic profile directly exposes this concept. In its abstract form, epistemic profile, being arbitrary function, conveys all reasoning capabilities of an agent. Thanks to this generic definition, also non-deductive reasoning methods like argumentation schemes, can be included as a part of epistemic profiles.

In this dissertation, *paraconsistent argumentation schemes (PAS)* have been analyzed in the context of constructing epistemic profiles of agents. To this end, following the approach of [69], we simplified the set of critical questions to those pointing to the specific undercutters, further called *exceptions*. Next, we encoded the scheme *premises*, *exceptions* and *conclusion* using four-valued literals (see Table 5 for an example of our paraconsistent adaptation of the Expert Opinion scheme). To our best knowledge, our paraconsistent approach to argumentation schemes is a novelty in the literature, encompassing the dual definition of PAS:

- as a part of an agent's *epistemic profile* utilizing the notions of *constituents*, *consequents* and *belief structures* (see Section 3.1.1),
- directly translated into 4QL, with the use of the notions of *modules* and *well-supported models* (see Section 3.1.2).

Technically, any 4QL program presented in **P3** can be implemented and interpreted using the 4QL interpreter `inter4ql`.

3.1.1 PAS as a Part of Epistemic Profile

In **P3**, paraconsistent argumentation schemes are modeled with the use of the two dedicated sets of *premises* and *exceptions*. Intuitively, when all premises are present and none of the exceptions is present, the conclusion of the scheme can be drawn.

Table 5: Paraconsistent Expert Opinion Scheme

	Paraconsistent Argumentation Scheme
P-Scheme	isExpert(A,D) assert(A,X,V) inDomain(X,D)
	$v(is(X)) = V$
Exceptions	¬isReliable(A) ¬evidenceBased(A,X,V)

Consider three sets of ground literals: Premises (P), Exceptions (E) and Conclusions (Con), and a function $\mathcal{PAS}(\{P, E\}) = Con$, which represents the *paraconsistent argumentation scheme*. The set P contains *candidates for conclusion* of the scheme. They are obtained by means specific to every argumentation scheme. The elements of E are *triggers* that, when present, forbid the respective candidate conclusion from being drawn. Intuitively, a conclusion c cannot be obtained when the exceptions indicate $\neg c$. Ultimately, the conclusion of the scheme is obtained as follows. If there exists a tetravalent candidate for a conclusion $c \in P$ (value of c is not $\mathfrak{u}$), check whether there exists a trigger $\neg\, c \in E$ blocking this candidate (value of $\neg c$ is $\mathfrak{t}$). If the trigger:

- does not exist, the candidate conclusion becomes the final scheme conclusion,
- exists, the scheme cannot be applied causing the value of $c \in Con$ to be $\mathfrak{u}$.

In short, a conclusion c is established based on the supporting arguments given by the set P (i.e., $c(P, v) \neq \mathfrak{u}$) and (lack of) rebutting triggers provided by the set E (i.e. $\neg c(E, v) \neq \mathfrak{t}$).

The definition below presents the paraconsistent argumentation scheme as a partial function: a fragment of agent's epistemic profile that expresses agent's argumentative skills. The translation of PAS to 4QL is presented in Definition 9, together with the analogy between both definitions.

Definition 8 (PAS) Recall that

- $\mathbb{C} = \text{F}{\small\text{IN}}(\mathcal{G}(\mathit{Const}))$ stands for the set of all finite sets of ground literals over the finite set of constants Const,
- $v : Var \longrightarrow \mathit{Const}$ is a valuation of variables.
- by a constituent we understand any set $C \in \mathbb{C}$.

Let

- P and E be two constituents, representing the set of premises and exceptions, respectively,
- $\mathcal{S} = \{P, E\}$ be a nonempty set of constituents ($\mathcal{S} \subseteq \mathbb{C}$),
- $Con \in \mathbb{C}$ be a finite set of ground literals, representing the conclusions.

Then, *Paraconsistent Argumentation Scheme (PAS)* is a partial function $\mathcal{PAS} : \text{F}{\small\text{IN}}(\mathbb{C}) \rightarrow \mathbb{C}$, $\mathcal{PAS}(\{P, E\}) = Con$, such that

$$c(Con, v) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} & \text{iff} \quad c(P, v) = \mathfrak{t} \text{ and } \neg c(E, v) \neq \mathfrak{t}; \\ \mathfrak{i} & \text{iff} \quad c(P, v) = \mathfrak{i} \text{ and } \neg c(E, v) \neq \mathfrak{t}; \\ \mathfrak{u} & \text{iff} \quad c(P, v) = \mathfrak{u} \text{ or } \neg c(E, v) = \mathfrak{t}; \\ \mathfrak{f} & \text{iff} \quad c(P, v) = \mathfrak{f} \text{ and } \neg c(E, v) \neq \mathfrak{t}. \end{cases}$$

By belief structure over $\mathcal{PAS}$ we mean $\mathcal{B}^{\mathcal{PAS}} = \langle \mathcal{S}, Con \rangle$, where:

- $\mathcal{S} = \{P, E\}, \mathcal{S} \subseteq \mathbb{C}$ is a nonempty set of constituents;
- $Con \stackrel{\text{def}}{=} \mathcal{PAS}(\mathcal{S})$ is the *consequent* of $\mathcal{B}^{\mathcal{PAS}}$.

We identify $\mathcal{B}^{\mathcal{PAS}}$ with the instance of a paraconsistent argumentation scheme. ◁

Although epistemic profiles serve as a useful abstraction for characterizing agents' reasoning capabilities, in the remainder of this dissertation we assume that agents' reasoning is grounded in belief bases, rather than in arbitrary theories. That is, in reasoning we allow rules and facts and consider well-supported models only. The methods we apply have to be simple and effective. Therefore the main results: inquiry and persuasion dialogues (see Sections 3.3 and 3.4) are already directly implemented in 4QL.

3.1.2 Dual Definition of PAS in 4QL

Recall Example 1 from Section 2.3. The two modules of program P, top and su, define the sets of ground literals $\mathcal{M}_{top}$ (4) and $\mathcal{M}_{su}$ (3), which are the well-supported models of top and su, respectively. Taking the epistemic profile perspective, P can be seen as a belief structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{M}_{su}, \mathcal{M}_{top} \rangle$ with one constituent su and a consequent top while the epistemic profile $\mathcal{E}$ is defined by the rules of module top. Such an approach allows to encode agents' informational stance, expressed in the terms of belief structures, directly in the rule-based query language 4QL.

An argumentation scheme in 4QL is implemented via a dedicated `Scheme` module, containing two specific rules:

```
is(X):- Premises.is(X),
       -Exceptions.is(X) in {false,unknown,incons}.
```

and

```
-is(X):- -Premises.is(X),
         -Exceptions.is(X) in {false,unknown,incons}.
```

The multisource formulas in the bodies of the rules pertain to two other specific sub-modules: one corresponding to the premises and one to the exceptions. Intuitively, these rules express the mechanism of drawing the scheme conclusions in the way described in the previous Section. Altogether, the 3-modular 4QL architecture reflects the structure of the argumentation scheme:

- the set of $premises$ is translated to the module `Premises`,
- the set of $exceptions$ is captured in the module `Exceptions`,
- the $conclusion$ (`is(X)`) is evaluated within the `Scheme` module.

Definition 9 Let L stand for the set of ground literals with constants in Const. Let $Exceptions$, $Premises$ and $Scheme$ be three sets of rules. A *Paraconsistent Argumentation Scheme* is a tuple $PAS = \langle Exceptions, Premises, Scheme \rangle$, such that if

- $\mathcal{M}_{\mathcal{S}}$ is the *well supported model* of $Scheme$,
- $\mathcal{M}_{\mathcal{P}}$ is the *well supported model* of $Premises$,
- $\mathcal{M}_{\mathcal{E}}$ is the *well supported model* of $Exceptions$,

then $\exists c \in L$, such that

$$\mathcal{M}_{\mathcal{S}}(\mathrm{c}) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} & \text{iff } \mathcal{M}_{\mathcal{P}}(\mathrm{c}) = \mathfrak{t} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg \mathrm{c}) \neq \mathfrak{t}; \\ \mathfrak{i} & \text{iff } \mathcal{M}_{\mathcal{P}}(\mathrm{c}) = \mathfrak{i} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg \mathrm{c}) \neq \mathfrak{t}; \\ \mathfrak{u} & \text{iff } \mathcal{M}_{\mathcal{P}}(\mathrm{c}) = \mathfrak{u} \text{ or } \mathcal{M}_{\mathcal{E}}(\neg \mathrm{c}) = \mathfrak{t}; \\ \mathfrak{f} & \text{iff } \mathcal{M}_{\mathcal{P}}(\mathrm{c}) = \mathfrak{f} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg \mathrm{c}) \neq \mathfrak{t}. \end{cases}$$

Is such a case we define c as the *conclusion* of the paraconsistent argumentation scheme. ◁

Def. 9 presents PAS as a tuple of specific 4QL modules (sets of rules) while Def. 8 expresses PAS as a partial function: a fragment of an agent's epistemic profile. The set of conclusions of the scheme (Con) corresponds to the well-supported model of the $Scheme$ module ($\mathcal{M}_{\mathcal{S}}$).

Any argumentation scheme that can be represented as PAS can be translated to 4QL. Obtaining conclusion c of the scheme is in polynomial time, as it amounts to computing the well-supported model of the module $Scheme$. This complexity result allows to reason and compare conclusions obtained with the use of multiple different argumentation schemes at once.

3.2 Communication Forms

Human communication involves building complex models of interlocutors possibly in an unconscious way, which then play an implicit role in human reasoning. Explicit models of others is a necessary component in agency. Even though we cannot grasp the entire subtlety of human communication, adequate, yet complex, communication forms can be proposed, like the following.

1. Informing about (transient) *opinions* helps agents to form high-level models of others, i.e., involving expressed current beliefs.
2. Communication including revealing one's (persistent) *facts* and *reasoning rules* permits to create a (more adequate) model of others.
3. Finally, communicating whole reasoning/argumentation schemes resembles installing new software packages on an intelligent agent to create the models of others or to augment its own reasoning capabilities.

Accordingly, the research goals G1.1. and G1.2., *formalization of elementary and complex communication forms*, were realized in papers **P1**, **P2** and **P3**, with the focus on *speech acts*. Depending on different type of their *content* we discerned:

- *elementary communication forms* concerned with (4-valued) opinions (**P1**),
- *complex communication forms* covering reasoning rules (**P2**) or argumentation schemes (**P3**).

The scope of this research covered a model of three atomic speech acts: $assert$, $concede$ and $request$, and a compound speech act $challenge$. They were modeled by specifying their:

- *preconditions*, including the *communicative relation* with the sender (*authority*, *peer*, *subordinate*),
- *complex post-action*, covering belief revision and responding with a proper speech act.

As we concentrate on *perceiving* speech acts, the issue of *initiating* them was out of the scope.

3.2.1 Elementary Communication Forms

Communicating agents routinely adjudicate whether to adopt a new piece of information obtained via communication. As agents are heterogeneous, even when perceiving the same news they may *draw different conclusions* and *react differently*. In multiagent literature, there are three general approaches to model agent's reaction to a new piece of information:

- based on the message content φ,
- based on the message sender S, or
- based on both.

For example, in [64] the authors proposed an *acceptance attitude* which determines when a message can be accepted on the basis of its content solely, identifying three such attitudes: *credulous*, *cautious* and *skeptical*. They were defined with respect to agent's ability to construct an argument against φ regardless the features of the sender.

In richer communication models, also other aspects have been studied. In [91], the problem whether agent's statement (message) should be considered or disregarded was adjudicated with the use of a high-level concept: *credibility* of an agent, expressed by a *credibility function*. Qualities such as: veracity, prudence, perception, cognitive skills also played a role in judging agent's credibility.

In a slightly different approach of [62], the role of credibility plays *trust*, which is considered to be *"a mechanism for managing the uncertainty about autonomous entities and the information they deal with"*. Indeed, trust [15] is essential in creating a team via communication [35].

In **P1**, to generalize various aspects of communication such as *trust*, *credibility*, *power-relations*, the notion of **communicative relations** was introduced. Three types of such relations were characterized in terms of the receiver's reaction to the sender's messages: *communication with authority*, *peer to peer communication*, *communication with subordinate*. In short, communicative relations act as filters, determining when a new percept triggers *belief revision* or *conflict resolution*.

When considering message content φ, due to the 4-valued approach the number of possible disagreements concerning opinions about φ rises. In **P1** we distinguished two such cases:

- *strong disagreements (conflicts)*: when agents have contradictory opinions on φ;
- *mere discrepancies*: when one agent holds an inconsistent opinion about φ and the other believes φ is true or false.

To sum up, modeling an agent's reaction to a new piece of information in **P1** hinges upon *both message content φ and message sender S* and draws upon speech acts theory of Austin and Searle.

In **P1** we proposed a 4-valued model of speech acts $assert$, $concede$, $request$ and $challenge$, with the focus on assertions as the main concept. According to Searle and Vanderveken [76], the sincerity conditions of *assertions* require that the agent believes in what it asserts. Classically, assertions of beliefs were represented as $assert_S(\varphi)$ meaning that S asserts that φ holds. As the 4-valued case required extending notation, therefore the content of speech act is a literal together with its value. Next, the semantics of speech acts was specified by their *preconditions* and *complex post-actions*:

$$\{precondition\}\langle speech\ act\rangle[complex\ post-action].$$

The speech acts semantics has been detailed in the form of comprehensive tables in **P1** .

For assertions, six new cognitive situations have been characterized from the receiver's point of view, based on the communicative relation with the sender, and on the message content φ.

1. *Perceiving previously unknown information*, where the receiver, ignorant about φ, is informed about φ's value.
2. *Perceiving information that is unknown*, where the sender informs the receiver that φ is unknown.
3. *Perceiving previously inconsistent information*, where the receiver, believing φ is inconsistent, is informed about φ's value.
4. *Perceiving inconsistent information*, where the sender informs the receiver that φ is inconsistent.

5. *Perceiving compatible information*, where the sender and receiver agree on φ's value.
6. *Perceiving contradictory information*, where the sender and receiver strongly disagree on φ's value.

The semantics of the remaining speech acts: $concede$, $request$ and $challange$ is given in **P1**.

3.2.2 Advanced Communication Forms

In **P2** and **P3** advanced communication forms concerning more refined content such as agents' reasoning schemes were analyzed. These schemes were communicated via assertions.

In **P2** we analyzed how agents should react to perceiving assertions about *reasoning rules*: should they adopt, reject, ignore or maybe challenge the new rule? Obviously, adopting a new reasoning rule may induce conflicts, inconsistencies and deep changes in agents' belief structure. This subject has hitherto received little attention. In particular, in [23], a cooperative rule learning approach for exchanging sets of rules among agents has been presented. Formalism given in [58] concerned acceptability of inference rules. However, none of these approaches deals explicitly with unknown and possibly inconsistent information. Therefore the problem of rule adoption principle for communicating agents in the paraconsistent and paracomplete approach was investigated in G1.2. This goal was realized via a *rule admissibility criterion*, based on compatibility of the rule conclusions with the current belief structure. The compatibility was founded on the special knowledge-preserving ordering of truth values $\leq_k$ (see Figure 2) and was a solution for mitigating possible future conflicts incurred by adopting a new rule.

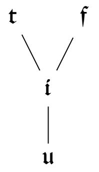

Figure 2: Knowledge ordering $\leq_k$

Incompatible belief structures were not the only threat to the system's stability investigated in **P2**. We made an original observation that dealing with unknown information is a delicate matter because accepting rules containing unknown literals is risky for the receiver. This problem was solved on a meta-level via the communication relations: rules containing unknown premises are considered only when the sender is an authority, otherwise, the unknown premises need to be resolved first. This leads to distinguishing the following cases.

- *Rule head is unknown, rule body is known.* The novel assembly of literals leads to a new, unknown beforehand conclusion and may be viewed as learning a new concept.
- *Rule head is known, rule body is unknown.* That case may be described as widening the knowledge, or making it more detailed.
- *Rule head is known, rule body is known.* Philosophically, such situation pertains to two different cases: the new rule is known as it is already, or the new rule combines previously known literals as premises (*Eureka!*). Such case may be described as knowledge discovery.
- *Rule head is unknown, rule body is unknown.* In that case, the agent is overburdened with new information and typically should start from resolving the unknown premises first.

The execution of the admissibility criterion is at the heart of the ***Algorithm for Perceiving Assertions About Rules***, a generalized 4-step procedure, realized via: *Filtering*, *Parsing*, *Evaluation* and *Belief Revision*. The impact of communicative relations is made explicit in the Algorithm:

- in the Parsing phase, if the rule premises are not recognized,
- in the Evaluation phase, if the rule conclusion is not recognized.

Avoiding conflicts is not a satisfactory solution when deep changes in agents' epistemic profiles turn out to be necessary as the rule admissibility criterion prevents them. Formalization of several argumentation schemes in **P3** allows for a more ontology-oriented approach to adjudicating about admissibility of a communicated rule. Simply, if a rule belongs to a special module encoding an argumentation scheme it should be accepted as it expresses a canonical reasoning method. This approach can be extended to allow communication about modules (sets of rules).

To sum up, we took two approaches at defining rule admissibility: a semantic one, based on compatibility of belief structures (in **P2**), and a syntactic one, based on structure/labeling of rules or whole modules (in **P3**).

3.3 Formalizing Inquiry in TalkLOG

Research goal G2, i.e., formalization of *inquiry*: a dedicated dialogue type for knowledge acquisition, was realized in the paper **P4** [37], where a paraconsistent and paracomplete model of *inquiry dialogue* was proposed. The purpose of inquiry, as formulated by Walton and Krabbe, is to collectively solve a theoretical problem [94]. In multiagent settings, inquiry *"starts when some agents are ignorant about the solution to some question or open problem. The main goal is the growth of knowledge, leading to agreement about the conclusive answer of the question. This goal may be attained in many different ways, including an incremental process of argument which builds on established facts in drawing conclusions beyond a reasonable doubt. Both information retrieval and reasoning may be intensively used in this process"* [35]. Inquiry is not only a vital dialogue in multiagent systems' team formation (e.g., during action allocation or as an embedded dialogue in task division [35]) or team action. It also offers intelligent agent interface to human users for intelligent information retrieval through searching and collectively arriving at a final answer/recommendation for the user.

3.3.1 Better Discernment between Inquiry Types

In its paradigmatic form, inquiry seeks to prove a statement as true or false, e.g.:

I1. $is(suspect, guilty) = \mathfrak{t}$,

I2. $is(suspect, guilty) = \mathfrak{f}$.

These options do not exhaust all possibilities in realistic modeling. To overcome the limitations of 2-valued approach, in TalkLOG we additionally considered influence of incomplete and contradictory information on inquiry scenarios, providing their more granular discernment:

I3. $is(suspect, guilty) = \mathfrak{i}$: the information about suspect's guilt is inconsistent

I4. $is(suspect, guilty) = \mathfrak{u}$: is suspect guilty?

A scenario where the subject s of inquiry is unknown resembles a so-called *discovery dialogue*, where *"the question whose truth is to be ascertained may only emerge in the course of the dialogue itself"* [59]. In TalkLOG, it is just another variation of inquiry (see Definition 13):

- **Inquiry-WHAT**: when initial value of s is $\mathfrak{u}$,
- **Inquiry-THAT**: when initial value of s is $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$.

As a group activity (in contrast to *information seeking* dialogue), inquiry aims at obtaining the *common belief* (C-BEL$_G(s)$, see [35] for the definition) in the group G about the dialogue topic s [35]. Unlike *common knowledge*, C-BEL$_G(s)$ does not entail that s is true. Indeed, rather than proving a statement to be true or false, in TalkLOG, the goal of inquiry is, technically speaking, to jointly *find a proof* of s. Such a commonly believed proof implies commonly believed s.

Inquiry-THAT can be viewed as a collaborative proof-searching process which starts from a *working hypothesis* and aims at confirming it. Thus, Inquiry-THAT succeeds if the final value v_f is equal to the initial value v_i of s (see Table 6 and Definition 13). When the initial subject value v_i is *unknown*, in the lack of a working hypothesis, *any* final valuation different from $\mathfrak{u}$ is a success. Thus Inquiry-WHAT succeeds if $v_f \neq \mathfrak{u}$ (see Table 6 and Definition 13).

Table 6: Summarised view of Inquiry-WHAT and Inquiry-THAT.

	Inquiry-WHAT		**Inquiry-THAT**	
Initial Topic Value	$\mathfrak{u}$		$\mathfrak{t}, \mathfrak{i}, \mathfrak{f}$	
Final Topic Value	$\mathfrak{u}$	$\mathfrak{t}, \mathfrak{i}, \mathfrak{f}$	Same as Initial	Otherwise
Outcome	failure	success	success	failure

Although the primary outcome of TalkLOG inquiry is the proof of the goal, for simplicity (and to maintain compatibility with other approaches) we formalize the inquiry outcome as a pair: the *final value of the goal* **and** the *proof* of it (where the *proof* is the primary notion and the *value* of the goal is a derivative).[6]

3.3.2 Principles of TalkLOG Inquiry

In the literature there are different approaches to dialogue: some people assume that dialogue protocols should only enforce coherence of dialogues, others also assume rationality and trustworthiness of the agents involved in a dialogue [66]. We lean towards an approach where *"the idea is to have protocols specify the minimum rules the agents should respect. Agents need to be free to do whatever they want as long as the main rules are respected"* [17]. Although, as mentioned by Dignum [17], being flexible probably means being computationally expensive, our research shows that does not necessarily has to be true as shown in Section 3.3.6.

TalkLOG inquiry is governed by the following principles.

[Cooperativeness] We deal with a finite set of n cooperative agents who do not withhold information[7]. Agents' belief bases are encoded as finite, ground 4QL programs $P_1, \ldots, P_n$, that share a common ontology[8]. Agents communicate one-to-all without coordination[9] and their final beliefs are expressed by the well-supported models $\mathcal{M}_{P_1}, \ldots, \mathcal{M}_{P_n}$ of their programs.

[Activeness] In between joining and leaving a dialogue an agent must make at least one relevant move.

[Compliance] Agents' programs do not change during dialogue[10].

[Sincerity] Agents do not lie about their beliefs nor contents of their programs.

[Pragmatism] Agents cannot repeat assertions.

---

[6] Recall that computing the value of the goal is in polynomial time wrt. the size of the proof, thus storing the pre-computed value greatly reduces lookup time.

[7] This implicitly constraints the number of queries to dialogue stores per one locution as agents would not query a dialogue store when they can provide input.

[8] Practical realization of common ontology assumption is non-trivial, see [53].

[9] Cp. [90] where agents act, i.e., listen, reason and speak, in turns, for a fixed number of rounds.

[10] This requirement can be relaxed as discussed in Section 3.3.5.

Inquiry starts when an initiator opens the dialogue on topic s. This topic becomes the root of the backward chaining tree. All participants may contribute to the dialogue by providing their (partial) proofs of s constructed via backward chaining. Next, the elements of the proofs uttered by the agents are added to the scope of the dialogue allowing all participants to further provide the (partial) proofs for these elements. Agents cannot retract their locutions nor repeat them. Since their programs are finite, we finally get to a point where no agent has a move to make and dialogue terminates naturally.

Due to [17], dialogue evaluation criteria should consider not only the communication mechanism (the protocols), but also *agents' argumentative attitudes*, which express their preference to give support or to attack, when multiple moves are possible. Accordingly, in TalkLOG we distinguish the following attitudes:

1. **compliant**, when agent attacks only as a last resort,
2. **offensive**, when agent seizes each opportunity to attack immediately.

Agent's attitude is **indefinite** when it exhibits no clear preference for attacks or supports. In the sequel we postulate this attitude. Unlike other approaches, we do not assume that the distributed knowledge of the group is complete. If the statement s cannot be proved by the agents, the conclusion would simply be $\mathtt{u}$.

3.3.3 Architecture of Multi-Party Inquiry

Typically dialogues are modeled by specifying the format and semantics of permissible messages. As inquiry is a joint search for a proof, the relevant speech acts are: *assertions* and *questions* (or *requests*, see Table 7). Sequences of locutions uttered by the same sender are called *moves*. Recall that according to the speech acts theory, the sincerity condition of assertions states that an agent is *committed* to the truth of the communicated proposition. As a novelty in TalkLOG, assertions about *4-valued facts* and *2-valued rules* are admitted. Naturally, this yields additional questions as compared to the classical case, e.g., regarding accepting or rejecting perceived information (see Section 3.2 for a summary). In general, these issues can be investigated *from the individual* and *from the group* perspective.

On the individual level, a common technique is to equip agents with an 'attitude', which determines how to 'behave' in the dialogue (i.e., when to accept a proposition or to utter one, see e.g., *acceptance* and *assertion* attitudes of [64]). However, such an attitude-based solution is not satisfactory as we focus on the conversing *group*. In TalkLOG, group's behavior is regulated by the *dialogue rules* where the group-level equivalents of the 'acceptance' and 'assertion' attitudes are specified within the dialogue protocol.

Table 7: Formats and intended meaning of permissible locutions in inquiry.

Speech Act	Format	Intended Meaning
assert	$assert(S_i, r, d)$	Participant S_i asserts a fact or rule r in dialogue d.
request	$requestAll(S_i, d)$	Participant S_i requests all open questions in dialogue d.

A commonly accepted model of inquiry [7, 77] hinges upon two stores:

- *Query Store* (QS), containing *current open questions* (associated with the dialogue),
- *Commitment Store* (CS), containing the *statements (commitments)* expressed in the dialogue (associated with each individual agent).

To guard **focus** and **coherence** of dialogues, typically a *move relevance function* [65] is given, to determine, on the basis of the move and Query Store content, which statements can be added to the Commitment Store. In TalkLOG, there is a single Query Store (QS_d) associated with the dialogue (d) as usual (see Definition 12). Initially it contains the dialogue goal s as a single entry. The manner of adding new and removing old questions is the essence of QS_d definition. In this respect, the novelty in TalkLOG is that the issue when questions should be added to or removed from QS is studied deeper due to the 4-valued approach. Section 3.3.4 is dedicated to this matter.

However, to satisfy both the social postulates [17] and requirements of the 4-valued approach, we had to redefine the notion of the Commitment Store (see Definition 11). Therefore, in contrast to the existing approaches, we do not consider particular individual agents' Commitment Stores but maintain a *single*, *associated with dialogue*, Commitment Store (CS_d). CS_d is created empty when the dialogue begins (as no locutions have been uttered yet) and updated with every relevant assertion. In short, the inquiry Commitment Store is just an evolving 4QL program (see also [1]).

Arguably, TalkLOG's inquiry architecture can be viewed as an adaptation of blackboard metaphor (like [26]'s *forum* or *newsgroup*) rather than exact *blackboard architecture* [22, 42].

Definition 10 (Locution Relevance) Locution m^t is *relevant* to inquiry dialogue d at time t iff.

$$m^t = assert(S, M_i.\ell \text{ :- } b, d) \text{ and } (\neg)M_i.\ell \in QS_d^t,$$

where QS_d^t is the Query Store of d at time t. We alternate between the notions of locution, message and utterance. ◁

Definition 11 (Commitment Store) *Commitment Store* of a dialogue d at time t is a 4QL program denoted as $CS_d^t = \langle M_1^t, \dots, M_k^t \rangle$:

- $CS_d^0 = \emptyset$
- $CS_d^t = CS_d^{t-1} \cup \{M_i.\ell \text{ :- } b\}$, such that $m^t = assert(S, M_i.\ell \text{ :- } b, d)$ is relevant to d at time t,
- $CS_d^t = CS_d^{t-1}$ otherwise. ◁

Definition 12 (Query Store) Let:

- CS_d^t be the Commitment Store of dialogue d at time t,
- m^t be the message received at time t,
- $close : \textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \to \textsc{Fin}(\mathbb{C})$ be a method for removing entries from the Query Store,
- $open : \textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \to \textsc{Fin}(\mathbb{C})$ be a method for adding entries to Query Store.

Then, *Query Store* of an inquiry dialogue d on subject s at time t is a finite set of literals denoted as QS_d^t such that:

- $QS_d^0 = \{s\}$
- $QS_d^t = (QS_d^{t-1} \cup B') \setminus B''$, if $m^t = assert(S, M_i.\ell \text{ :- } b, d)$, where

$$B' = open(b, CS_d^t),$$
$$B'' = close(QS_d^{t-1} \cup B', CS_d^t),$$

- $QS_d^t = QS_d^{t-1}$ otherwise. ◁

Definition 13 (Inquiry Types) For an inquiry terminating at time t, with the goal s of initial valuation v_i, the value of the dialogue conclusion is $v_f = v(s, \mathcal{M}_{CS^t_d})$, where $\mathcal{M}_{CS^t_d}$ is the well-supported model of CS^t_d. Dialogue is:

- **successful** iff
  - $v_i = \mathfrak{u} \wedge v_f \neq \mathfrak{u}$ [Inquiry-WHAT], or
  - $v_i \neq \mathfrak{u} \wedge v_f = v_i$ [Inquiry-THAT],
- **unsuccessful** otherwise. ◁

3.3.4 Refined Inquiry Strategies

TalkLOG provides a possibility to develop various *strategies* for conducting inquiry. Technically, an inquiry strategy (ST) is a *pair of methods* for adding (method $open$) and removing (method $close$) entries from the Query Store ($ST = \langle open, close \rangle$).

The precise definition of $open$ and $close$ methods may be custom and may originate from various approaches, among others, numerical (e.g., restricting the number of simultaneously open threads w.r.t. current CS size), semantic (e.g., opening threads based on the truth value of the literal *representing* the thread) or based on social choice theory (e.g., voting). In TalkLOG we adopted the semantic approach, concerning the way of dealing with inconsistent or unknown information. Intuitively, a new thread can be opened (or closed) based on its current truth value.

In TalkLOG we distinguished two methods for adding (ADD1, ADD2) and two methods for removing (REM1, REM2) literals from QS, leading to four inquiry strategies (see Table 8), which were thoroughly analyzed in **P4**.

Table 8: Inquiry strategies as pairs of methods for updating QS.

	REM1	REM2
ADD1	**narrow-minded**	**pragmatic**
ADD2	**forgetful**	**open-minded**

Implementation and analysis of our inquiry dialogues required a formal specification of concepts of *dependence set* and *proof* in 4QL. A dependence set of a literal ℓ from a program P consists of literals reachable via backward chaining on P from ℓ. Thus, TalkLOG inquiry is a *paraconsistent and paracomplete distributed version of backward chaining*.

Definition 14 Let ℓ be a literal and P a 4QL program. A *dependence set* of ℓ from P, denoted $\mathcal{D}_{P,\ell}$ is a set of literals such that:

- $\neg\ell, \ell \in \mathcal{D}_{P,\ell}$,
- if there is a rule ℓ' :- $b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$ in P, such that $\ell' \in \mathcal{D}_{P,\ell}$ then $\forall_{j \in 1..m} \forall_{k \in 1..i_j}\ b_{jk}, \neg b_{jk} \in \mathcal{D}_{P,\ell}$. ◁

A proof of a literal ℓ from a program P is a subprogram S of P generated from the dependence set $\mathcal{D}_{P,\ell}$ by taking all rules and facts of P whose conclusions are in $\mathcal{D}_{P,\ell}$. We also formally specified the size of proof and size of proof domain, concepts needed for complexity considerations.

Definition 15 Let ℓ be a literal, P a 4QL program. A *proof* of l from P is a 4QL program $S \subseteq P$ such that $\delta \in S$ iff $head(\delta) \in \mathcal{D}_{P,\ell}$, where δ is a fact or a rule. The *size of the proof S* is the size of the program S. The *size of domain of the proof S* is the size of the dependence set $\mathcal{D}_{P,\ell}$. ◁

3.3.5 Verified Properties

There are several typically investigated dialogue properties.

- Termination:
  - guaranteed (regardless of the sequence of messages) or
  - possible (in at least one sequence of messages)
- Computational complexity of terminating dialogues: how quickly do they terminate?
- Convergence to the merged outcome:
  - necessary - if regardless the sequence of messages each dialogue converges to the merged outcome?
  - possible - is there at least one sequence of messages, s.t. the dialogue converges to the merged outcome?
- Winning strategies of participants - if, regardless of other participants' moves, an agent can execute a sequence of dialogue moves which guarantees the preferred outcome?
- Soundness and completeness.

As in inquiry we did not define conditions for an agent to win or loose in the dialogue, we were not interested in investigating strategies of individual agents, but rather the properties of designed dialogues. In this regard, **soundness** of a strategy means that *whenever a dialogue terminates with a given conclusion, the same result would be obtained from the union of all the agents' belief bases*. In other words, **sound dialogues necessarily converge to the merged outcome**.

Definition 16 A strategy ST is *sound* iff whenever dialogue d on subject s conducted under this strategy terminates at t with conclusion k, then:

$$\text{if } v(s, \mathcal{M}_{CS_d^t}) = k \text{ then } v(s, \mathcal{M}_{\bigcup_{i \in 1..n} P_i}) = k.$$

If a solution is obtainable from the union of agents beliefs, an inquiry under a **complete** strategy will reach it.

Definition 17 A strategy ST is *complete* iff whenever dialogue d on subject s conducted under this strategy terminates at t with conclusion k, then:

$$\text{if } v(s, \mathcal{M}_{\bigcup_{i \in 1..n} P_i}) = k \text{ then } v(s, \mathcal{M}_{CS_d^t}) = k.$$

Properties of soundness and completeness are the most important ones in research on dialogue as they relate the outcomes of dialogues to the merged belief bases of participants. However, choosing a correct merging operator is not trivial. Merging operators investigated in the dialogue and argumentation literature cover the range from simple union of agents' beliefs [7] to complex consensual merges with expansion [24]. Selecting a merging operator not suitable for the problem at hand is a common hindrance in proving soundness and completeness of protocols (e.g., [8]). We postulate that the definition of merging operator should reflect the nature of dialogue. In case of TalkLOG inquiry, it is a simple union of all the agents' belief bases. In this respect, the main results from **P4** are the following.

Theorem 1 *Narrow-minded strategy is neither sound nor complete. Moreover, it is type 1 nondeterministic.*[11]

---

[11] Type 1 nondeterminism in logic programs means freedom to choose the rule to apply [75].

Theorem 2 *Forgetful and narrow-minded strategies are equal.*

Theorem 3 *Pragmatic and open-minded strategies are equal.*

Theorem 4 *Open-minded strategy is sound and complete.*

Although in **P4** we assumed that agents' programs **do not change during dialogue**, this requirement is superfluous. Indeed, the move relevance function admits to CS all communicated facts and rules whose heads match the entries in QS syntactically, without considering their current truth values in CS. The manner of adding/removing entries to QS depends on the dialogue strategy. Adding new facts and rules from CS to an agent's program cannot reduce the dependence set of the dialogue topic s from the program. In open-minded strategies QS grows monotonically. It's easy to see that in such strategies, at a given timepoint t, the union of the dependence sets of the entries in QS^t from an agent's program P^t equals the dependence set of the dialogue topic from $P^t \cup CS^t$:

$$\bigcup_{\ell \in QS^t} \mathcal{D}_{P^t,\ell} \equiv \mathcal{D}_{P^t \cup CS^t,\, s}$$

This follows from the definition of dependence set and QS under open-minded strategy:

- $\forall_t \;\; s \in QS^t$
- if $M_i.l$:- $b \in CS^t$, s.t. $b = b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$, then $\{b_{j_k} | j \in 1..m, k \in 1..i_j\} \subset QS^t$

3.3.6 Complexity of Inquiry

The ultimate elements of the proposed solution are complexity considerations. This happens to be yet another distinguishing feature of our research. In related works, precise complexity results are usually:

- high, e.g., ”since the protocols are based in logic we know that the complexity will be high” [64] or [17], or
- not given, e.g., when ”system is complicated and involves many interacting components” [6].

However, some approaches try to tackle these problems, e.g., [40, 64]. Although, due to [17], flexibility in dialogues comes at a price of high computational complexity, our research proves differently.

When analyzing complexity of dialogues, several aspects come to play, such as:

1. the complexity of selecting optimal locution at each step of the dialogue,
2. how many dialogue steps are needed for dialogue to terminate, and
3. what is the complexity of obtaining conclusion of a terminated dialogue.

In general, the complexity measures of inquiry are divided into two sorts:

- *communication complexity*, concerning only the amount of communication among agents (who have unlimited computational power) [50],
- *computational complexity* (data complexity), concerning the amount of computation (when communication is free) required to:
    - achieve dialogue termination,

- obtain a conclusion of a terminated dialogue.

Computational complexity of both problems is expressed in terms of *data complexity* [61, 88], i.e., complexity of evaluating a fixed query (here: inquiry goal) on an arbitrary database (here: Commitment Store of the dialogue). Thus data complexity is given as a function of the size of Commitment Store, which is an evolving 4QL program. Recall that the semantics of 4QL is defined by *well-supported models* [56, 79], i.e., models consisting of (positive or negative) ground literals, where each literal is a conclusion of a derivation starting from facts. For any set of rules, such a model is uniquely determined and computable in deterministic polynomial time $O(N^k)$ where N is the size of domain and $k = max(s, t)$ where s is the maximal arity of relations and t is the maximum number of free variables. As in our research we deal with ground programs, $t = 0$. When s is a bound constant, which is the case in practical applications of 4QL (qualitative not quantitative reasoning), we achieve tractability.

Below, the complexity results from **P4** are recalled and summarized in Table 9..

Theorem 5 *If the size of the domain of the proof of* s *is* N*, then the size* $|QS|$ *of the Query Store at the end of the open-minded inquiry is* $N/2 \leq |QS| \leq N$.

Theorem 6 *If the size of the proof of* s *is* M*, then the size* $|CS|$ *of the Commitment Store at the end of the open-minded inquiry is* $|CS| = M$.

Theorem 7 *Communication complexity of inquiry is* $O(nM)$.

Theorem 8 *Computational complexity of a narrow-minded inquiry is* $M \times O(N^k)$.

Theorem 9 *Computational complexity of termination of open-minded inquiry is* $O(1)$.

Theorem 10 *Obtaining the conclusion of a terminated open-minded inquiry is* $O(N^k)$.

Table 9: Results for open- and narrow-minded inquiries

Characteristics	**Open-minded**	**Narrow-minded**
Open vs. Closed System	open (at least one $assert$ per $join$)	
Addressing	one-to-all	
Coordination	asynchronous	
Properties	sound and complete	not sound and not complete
Communication Complexity	$O(nM)$	$O(nM)$
Computational Complexity (Termination)	$O(1)$	$O(MN^k)$
Computational Complexity (Obtaining Conclusion)	$O(N^k)$	$O(1)$
Total Store Size	$M + N$	

3.3.7 Conclusions

To meet the requirements of information-rich environments as well as demands of realistic modeling of agency, we presented a formalization of multi-party, paraconsistent and paracomplete inquiry in nonmonotonic, dynamic TalkLOG setting, concluding the following.

1. The spectrum of situations in which TalkLOG inquiry is applicable was widened comparing to the existing approaches.
2. Two additional logical values ensure a better discernment between inquiry types.

3. TalkLOG inquiry architecture builds upon the existing standard solutions with the following novel elements:
    - Commitment Store is associated with the dialogue and not with the particular agents,
    - both facts and rules are admitted to Commitment Store, which is a monotonically growing 4QL program,
    - definition of Query Store relies on two generic methods for adding and removing entries to the store, together interpreted as an inquiry strategy,
    - inquiry strategies can be instantiated to fit the circumstances in question.
4. Specifically, TalkLOG architecture allowed to obtain a protocol with public semantics, suitable for an arbitrary number of participants holding possibly different initial opinions regarding the statement to prove.
5. The concept of TalkLOG inquiry strategy permits to calibrate the depth of inquiry.
6. Our methodology allows one to provide:
    - new inquiry strategies and evaluate their relationships (e.g., (in)equality),
    - analysis of termination, soundness and completeness of inquiry under given strategies,
    - communication and complexity results of inquiry,
    - limitations on the dialogue store sizes.
7. In TalkLOG inquiry agents' beliefs may change in the course of dialogue according to the state of Commitment Store.

3.4 Formalizing Persuasion in TalkLOG

Research goal G3, aiming at a formal model of *persuasion* as a dedicated dialogue type for *conflict resolution*, was realized in the paper **P5** [36][12], covering a paraconsitent and paracomplete model of *persuasion dialogue*.

Classically, the initial situation of Walton and Krabbe's persuasion is a conflict of opinions. The goal of the dialogue is *"a resolution of the initial conflict by verbal means"* [94]. Typical approaches to modeling persuasion are based on 2-valued logic and polarized the set of participants into two **parties**: the proponents (PRO) of a proposition φ (a group of agents believing φ is t) and the opponents (OPP, sometimes also called CON, a group of agents believing φ is $\mathfrak{f}$). Both parties fight to persuade the others to change their opinions. As a novelty, within TalkLOG, we investigated *multi-party* persuasion in the new 4-valued modeling perspective. Indeed, in TalkLOG a dialogue party is a group of dialogue participants holding the common goal: *to bring about the preferred state of the world*.

Walton and Krabbe characterize three types of opinions: *positive*, *negative* and *one of doubt*. The classical understanding of conflict of opinions is based on the dichotomy of *truth* and *falsity*. Such an approach is not informative enough in the 4-valued logical settings. Thus in TalkLOG:

- the 'doubtful' case is expanded, distinguishing situations when the doubt results from *ignorance* or from *inconsistency*,
- the *opinions* are represented as 4-valued literals and the *conflict of opinions* is defined as inequality of these literals's truth values (see Definition 19).

---

[12] This Section is based on an extended version of **P5** submitted to Fundamenta Informaticae.

Therefore the whole spectrum of persuasion scenarios has been widened.

Although classical persuasion deals with conflicts of *opinions*, in multiagent settings the conflict may concern also other attitudes, like intentions [35]. This inspired us to deepen the study of persuasion towards the *reasons* underlying the conflict of opinions. In reality it often happens that the same opinion may be motivated by different, in extreme cases even antithetic, reasons. In autonomous systems such a situation may lead to unintended/unintuitive results, especially in cooperation, when reconciling motivational attitudes is vital. Heretofore no method allowing to isolate and separately analyze the cases of conflicts of opinions and their motivations existed. To reveal such cases we introduce *Deep Persuasion*, which:

- can commence even when opinions agree,
- aims at resolving conflicts of *motivations of opinions* (see Definition 19).

Like for inquiry, soundness and completeness of persuasion was evaluated by comparing the outcomes of an n-agent dialogue with the outcomes obtained by merging knowledge of these n agents. The key point was a proper construction of the merge operator. Moreover, termination and some complexity results were given.

3.4.1 Modeling Opinion and Motivation

In classical logical formalizations of multiagent systems, beliefs (or opinions) are modeled via an epistemic modal operator BEL. Thus, $BEL(i, \varphi)$ stands for agent i believes that φ holds. In TalkLOG, agent A is associated with a 4QL program P_A and beliefs are represented by 4-valued literals from the well-supported model. In order to account for two additional truth values, an opinion will be represented as a pair $o = \langle \varphi, v \rangle$, where $\varphi \in \mathcal{G}(Const)$ is a literal and $v \in \mathbb{T}$ is a truth value with the intended meaning that o is an opinion v on φ (or: o is a belief that φ is v).

As regards *motivation of an opinion* v on φ, we identify them with an explanation or justification for φ being v. Such justification may express how the value v for φ has been reached. In logical systems it simply means a *proof* for φ. In TalkLOG, the *motivation* of the opinion, called the *proper proof* of the literal representing the opinion (Definition 18), hinges upon:

- the *dependence set of a literal* ℓ *from a program* P (introduced in **P4**, see Definition 14), which consists of literals reachable via backward chaining on P from ℓ,
- the *proof of a literal* ℓ *from a program* P, containing rules (facts) whose conclusions are elements of the dependence set. Note that the proof may contain rules whose premises evaluate to $\mathfrak{f}$ or $\mathfrak{u}$, thus do not influence the value of ℓ. The definition of *proper proof* in **P5** disregards such rules.

Definition 18 (Proof, Proper Proof) Let

- ℓ be a literal,
- P be a 4QL program, $\delta \in \Gamma \cup \Pi$ be a fact or a rule,
- $S_{l,P}$ be the proof of l from P,
- $\mathcal{M}_{S_{l,P}}$ be the well-supported model of the proof.

The *proper proof* (p-proof) of l from P denoted $\Phi_{l,P}$, is a subset of $S_{l,P}$ such that $\delta \in \Phi_{l,P}$ iff $body(\delta)(\mathcal{M}_{S_{l,P}}) \in \{\mathfrak{t}, \mathfrak{i}\}$. $\lhd$

We define a conflict of opinions as inequality of truth values of the literals representing the opinions. By a conflict of motivation of opinion, we understand unequal p-proofs for the literal representing the opinion. Obviously, equality of p-proofs entails equality of opinions, but not the other way around.

Definition 19 (Initial Conflict on Topic) Let

- $\varphi \in \mathcal{G}(\mathit{Const})$ be a ground literal, representing the topic of the dialogue,
- $\langle \varphi, v \rangle$: $v \in \mathbb{T}$ be an opinion v on φ,
- P_1 and P_2 be two 4QL programs of agents A_1 and A_2,
- $\mathcal{M}_{P_1}$ and $\mathcal{M}_{P_2}$ be the well-supported models of P_1 and P_2 respectively,
- Φ_{φ,P_1} and Φ_{φ,P_2} be the p-proofs of φ from P_1 and P_2 respectively.

Then

- an *initial conflict on topic* φ between A_1 and A_2 occurs when:
  - $\varphi(\mathcal{M}_{P_1}) \neq \varphi(\mathcal{M}_{P_2})$, or [conflict of opinion]
  - $\Phi_{\varphi,P_1} \neq \Phi_{\varphi,P_2}$ [conflict of motivation]
- A_1 and A_2 share a *common opinion on* φ if $\varphi(\mathcal{M}_{P_1}) = \varphi(\mathcal{M}_{P_1})$,
- A_1 and A_2 share a *common motivation on* φ if $\Phi_{\varphi,P_1} = \Phi_{\varphi,P_2}$. $\triangleleft$

When dealing with an arbitrary number of dialogue participants, defining *success condition* of a multi-party persuasion is a subject of different potential formulations. In TalkLOG:

- **Classical Persuasion** is successful if a *common opinion* is obtained, i.e., all agents share the same opinion v on the topic. This robust success criterion requires a consensus, while there may me other possible solutions built upon single-winner voting methods, e.g., Borda count or Approval voting.
- **Deep Persuasion** is successful if a *common motivation* is obtained. This means, that in addition to the common opinion, all participants' p-proofs for that opinion must equal.

Similarly to inquiry, the conclusion of persuasion is a pair: the final value of the topic and the motivation (p-proof) for that value, obtained in the dialogue.

Communication about Opinion and Motivation In TalkLOG persuasion we discern between defeasible *beliefs* (opinions) and persistent *pieces of evidence* (building blocks of motivation). While beliefs are literals from the well-supported model, pieces of evidence are the facts and rules from agent's 4QL program. We introduce a membership function μ (see Definition 20) expressing the degree of membership of a piece of evidence to a program.

Definition 20 Let P be a 4QL program, $\delta \in \Gamma \cup \Pi$ be a fact or a rule and $\mathit{Lab} = \{t, f, b, n\}$ be the set of labels. Then $\mu_P(\delta) : \Gamma \cup \Pi \to \mathit{Lab}$ encodes the membership of δ to P:

$$\mu_P(\delta) \overset{\text{def}}{=} \begin{cases} t & \text{when } \delta \in P \wedge \neg\delta \notin P; \\ f & \text{when } \delta \notin P \wedge \neg\delta \in P; \\ b & \text{when } \delta, \neg\delta \in P; \\ n & \mathit{otherwise}. \end{cases}$$

To sum up, in TalkLOG persuasion, the *content* of a locution is of two sorts:

- an *opinion* (*belief*) $o = \langle \varphi, v \rangle$,
- a *piece of evidence*: a fact/rule δ together with $\mu(\delta)$.

Conflicts of evidence are distinguished when membership functions disagree. In TalkLOG, we provided considerably different conflict resolution mechanisms.

- *Conflicts of opinions* are resolved via argumentation (and embedded 4QL mechanisms), utilizing approaches known from deductive argumentation [5, 7, 64, 67, 68]. Importantly, all three types of attack known from Argumentation Theory, i.e., *on the premises*, *on the conclusions* and *on the inference step*, are realized in TalkLOG persuasion.

- *Conflicts of evidence* are adjudicated upon via dedicated conflict resolution methods, like social choice theory methods [10] selected here (particularly, voting).

As persuasion is a much more complex dialogue than inquiry, the list of permissible locutions was extended (Table 10) to allow for:

- *questioning* statements of other agents,
- expressing change of opinion via *retraction*,
- expressing change in motivation via *retraction* or *adoption*,

Table 10: Intended meaning of permissible locutions in TalkLOG persuasion.

Speech Act	**Format**	**Intended Meaning**
assert	$assert_x^d\langle\delta, \mu(\delta)\rangle$	asserting attitude towards evidence δ
concede	$concede_x^d\langle\delta, \mu(\delta)\rangle$	conceding/agreeing with evidence δ
assertBel assertBel	$assertBel_x^d\langle\varphi, v\rangle$ $assertBel_x^d\langle B, v\rangle$	asserting opinion $\langle\varphi, v\rangle$ asserting opinion $\langle B, v\rangle$
why	$why_x^d\langle\varphi, v'\rangle$	questioning opinion $\langle\varphi, v'\rangle$, $v' \neq \mathfrak{u}$
retract	$retract_x^d\langle\delta\rangle$	retracting evidence δ
adopt	$adopt_x^d\langle\delta\rangle$	adopting evidence δ
retractBel	$retractBel_x^d\langle\varphi, v\rangle$	retracting opinion $\langle\varphi, v\rangle$

3.4.2 Principles of TalkLOG Persuasion

TalkLOG persuasion is governed by the following principles.

[Cooperativeness] The setting consists of a finite set of n cooperative agents who do not withhold information[13]. Agents' belief bases are encoded as finite, ground 4QL programs $P_1, \ldots, P_n$, that share a common ontology. Agents communicate one-to-all without coordination; their final beliefs are expressed by the well-supported models $\mathcal{M}_{P_1}, \ldots, \mathcal{M}_{P_n}$ of the programs.

[Activeness] In between joining and leaving a dialogue an agent must make at least one relevant move.

[Compliance] Agents' programs change during dialogue according to the group decisions.

[Sincerity] Agents do not lie about their beliefs nor contents of their programs.

[Pragmatism] Particular agents cannot repeat $assert$ and $concede$ locutions, however, they **can** repeat $assertBel$ locutions (if separated by $retractBel$) as in the light of new evidence agents' beliefs may change.

To allow maximum flexibility only the above principles restrict agents communicating in TalkLOG. Persuasion starts when agent initiator asserts its initial opinion v_i about the dialogue topic s. This assertion becomes the subject of attacks and defenses for all participants who can:

- question the belief (why),
- attack the belief by asserting a different opinion ($assertBel$),
- support the belief by providing evidence and beliefs ($assert, assertBel$),
- attack the belief by providing evidence and beliefs ($assert, assertBel$).

---

[13] This implicitly constraints the number of queries to dialogue stores per one locution.

The details of particular moves are presented in **P5** in Section 4.

Notice that when outlining differences between Classical and Deep persuasion, we followed a *prescriptive* methodology, where we looked at initial conflicts and main dialogue goals from an omniscient outside observer perspective. Obviously this is not a perspective of agents in TalkLOG. Indeed, unlike **P4** where the discernment between Inquiry-WHAT and Inquiry-THAT can be done at the dialogue opening on the basis of initial topic value, here, distinguishing Deep from Classical persuasion cannot be made at the beginning. The formal model of both persuasion types differs only on the definition of the dialogue conclusion (cp. Definition 23 and 22).

3.4.3 Architecture of Multi-Party Persuasion

TalkLOG persuasion architecture follows the social approach [17] and was built upon TalkLOG inquiry, sharing the following features:

- single *Query Store* and *Commitment Store* associated with dialogue,
- *Commitment Store* as an evolving (monotonically) 4QL program,
- *move relevance* function to ensure focus and coherence of dialogues.

TalkLOG persuasion features four dialogue stores: Query Store (QS), Dispute Store (DS), Resolved Dispute Store (RDS) and Commitment Store (CS). QS and DS are defined through the auxiliary Store Update and One-step Update functions for specifying the effects of moves and particular locutions, respectively, on the stores' content. RDS and CS are defined in a straightforward way. Stores' definitions are included in **P5**; in this Section we provide explanations and intuitions. The format and meaning of entries in particular stores are recalled in Table 11.

Table 11: Formats and intended meaning of entries in TalkLOG persuasion stores.

Dialogue Store	**Format of Entries**	**Intended Meaning**
Query Store	$\langle bel, \varphi, v, A\rangle$ $\langle why, \varphi, v, \bot\rangle$	Agent A asserted belief $\langle\varphi, v\rangle$. Some agent questioned belief $\langle\varphi, v\rangle$.
Dispute Store	$\langle\delta, n_t, n_f, n_b, n_n\rangle$	There are n_t votes for accepting δ, n_f votes for accepting $\neg\delta$, n_b votes for accepting both δ and $\neg\delta$ and n_n votes for rejecting both δ and $\neg\delta$.
Resolved Dispute Store	$\langle(\neg)\delta, i\rangle$ $\langle(\neg)\delta, o\rangle$	$(\neg)\delta$ was accepted $(\neg)\delta$ was rejected
Commitment Store	δ	An accepted fact or a rule

Query Store The original functionality of Query Store in TalkLOG *inquiry* is here realized by two separate stores: *Query Store* and *Dispute Store*, each concerned with the different sort of locution content: belief or evidence, respectively. Agents may inspect QS to find:

- *questions* that need answering (why-tuples, see Table 11), or
- *beliefs* of others that can be questioned (bel-tuples).

Why-tuples do not persist the sender of the question about $\langle\varphi, v\rangle$. Due to the public character of the Query Store, any response to a question is visible to all participants, therefore who asked the question is irrelevant.

Dispute Store DS contains pieces of evidence δ that are put forward by agents to support a belief or respond to a question in QS. Agents query DS to find δ submitted by another agent, which they can support or dispute.

DS contains tuples of form $\langle \delta, n_t, n_f, n_b, n_n \rangle : (\Gamma \cup \Pi) \times \mathbb{N}^4$ where n_k is so-called *support counter* for label k of δ indicating how many votes for/against δ have been casted. In **P5** we introduced notation shortcuts for manipulating DS, which are elaborated on here. Importantly, there is only one entry for δ or $\neg\delta$ in DS. The exact form of the tuple depends on which issue (δ or $\neg\delta$) was first asserted in the dialogue.

1. We use the following notation concerning labels $k \in Lab$:

   $$\neg t = f; \neg f = t; \quad \neg b = b; \neg n = n$$

   Simply, n_t for δ is treated as n_f for $\neg\delta$. Moreover n_b and n_n for δ are the same as the respective values for $\neg\delta$.

2. We use $DS[\delta, k]$ to access the value of the support counter for k of δ:

   $$DS[\delta, k] \overset{\text{def}}{=} n_k, \quad DS[\neg\delta, k] \overset{\text{def}}{=} DS[\delta, \neg k].$$

3. We use $DS[\delta]$ to test if an entry for δ exists in DS:

   $$DS[\delta] \overset{\text{def}}{=} \begin{cases} \mathsf{t} & \text{iff } \langle \delta, n_t, n_f, n_b, n_n \rangle \in DS; \\ \mathsf{f} & \text{iff } \langle \delta, n_t, n_f, n_b, n_n \rangle \notin DS; \end{cases}$$

4. We use $DS[\delta, k]$++ to increment the support counter for k of δ:

   $$\begin{aligned} DS[\delta, k]{+}{+} \overset{\text{def}}{=} \quad & \{\langle x, n_t, n_f, n_b, n_n \rangle \in DS | x \neq \delta\} \cup \\ & \{\langle \delta, x_t, x_f, x_b, x_n \rangle \in DS : x_k = DS[\delta, k] + 1\} \end{aligned}$$

DS Update Function defines the way DS is updated after each move:

- $assert\langle \delta, \mu(\delta) \rangle$ results in creating a new tuple for δ (unless already exist) and increasing the support counter for $\mu(\delta)$,
- $concede\langle \delta, \mu(\delta) \rangle$ increases the support counter for $\mu(\delta)$ if the relevant tuple exist in DS.

Figure 3 illustrates serving an assertion of evidence by persuasion dialogue stores.

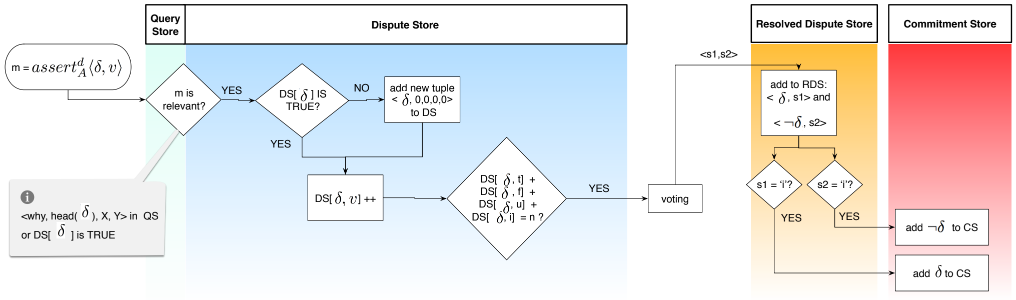

Figure 3: Serving an assertion of evidence.

Voting When a particular entry in DS received enough votes, the *voting function* (see Definition 21) is executed, determining the status (i or o) of the piece of evidence in question:

- if *accepted (in)*: status i,
- if *rejected (out)*: status o.

Notation-wise, if $\langle a, b\rangle$ is the outcome of voting for δ, then a is the status of δ and b of $\neg\delta$. For example, if $\langle o, o\rangle$ is the outcome of voting for δ then neither δ nor $\neg\delta$ are accepted. The voting mechanism is customizable, with the following restrictions:

- all votes have to be casted to obtain the voting outcome,
- antithetic rules cannot be accepted.

The first condition is required to obtain important properties of persuasion dialogues. The second condition reflects our modeling decision to avoid meta-level conflicts.

Definition 21 (Voting Function) Let $Status = \{i, o\}$, $\delta \in \Gamma \cup \Pi$, and $n \in \mathbb{N}$ be the number of dialogue participants. Then, a *voting function VF* is any function $VF : (\Gamma \cup \Pi) \times \mathbb{N}^5 \rightarrow Status^2$, such that

- $VF(\delta, n_t, n_f, n_b, n_n, n) = \emptyset$ iff $n_t + n_f + n_b + n_n \neq n$, and
- if $\delta \in \Pi$ (a rule) then $VF(\delta, n_t, n_f, n_b, n_n, n) \neq \langle i, i\rangle$. $\triangleleft$

Resolved Dispute Store The elements of RDS are tuples of the form $\langle(\neg)\delta, s\rangle : (\Gamma \cup \Pi) \times Status$, encoding the outcomes of voting over $(\neg)\delta$. This way, we can discern between a situation where $(\neg)\delta$ was considered but rejected (entry $\langle(\neg)\delta, o\rangle$) and a situation where $(\neg)\delta$ was not yet even considered (no entry for $(\neg)\delta$ in RDS). RDS is publicly available for participants to query it to learn the status of pieces of evidence. Agents are required to comply with the decisions encoded in RDS:

- adopt a piece of evidence with status i (and inform others using $adopt$), or
- abandon a piece of evidence with status o (and inform others using $retract$).

Monitoring these acknowledgments helps to verify agents' compliance with the dialogue protocol.

Commitment Store The elements of CS are facts or rules accepted by voting, i.e., the entries from RDS with status "in" (i). Like in inquiry, once accepted, entries are never removed from CS. Unlike inquiry, final CS contains statements to which all dialogue participants are committed to. Importantly, the current *well-supported model* of CS contains the beliefs that are currently justifiable on the grounds of accepted evidence. Obviously, the *well-supported model* changes non-monotonically, according to the course of discussion.

In TalkLOG, conclusion of persuasion on topic s with initial value v_i is the final value v_f of s, together with the motivation (p-proof) Φ_{s,CS^t_d} for that value, obtained in the dialogue. The final value of Deep Persuasion topic is obtained from the well-supported model of CS after dialogue termination (see Definition 22).

Definition 22 (Deep Persuasion Conclusion) Let CS^t_d be the Commitment Store of Deep Persuasion d terminating at t, with the topic s of initial value v_i. Then, the *conclusion* of d is $c = \langle v_f, S\rangle$ where

- $v_f = s(\mathcal{M}_{CS^t_d})$, where $\mathcal{M}_{CS^t_d}$ is the well-supported model of CS^t_d,
- $S = \Phi_{s,CS^t_d}$, i.e., S is the p-proof of s from CS^t_d.

However, for termination of Classical Persuasion, it suffices that all agents share the same belief about the topic (see Definition 23). Thus, the final value of Classical Persuasion is obtained from the Query Store and so it does not have to agree with the value following from the p-proof (Φ_{s,CS_d^t}). The manner of obtaining dialogue conclusion is definitely different in Deep and Classical Persuasion.

Definition 23 (Classical Persuasion Conclusion) Let QS_d^t be the Query Store and CS_d^t be the Commitment Store of Classical Persuasion d terminating at t with n participants $A_1, \ldots, A_n$, with the topic s of initial value v_i. Then, the *conclusion* of d is $c = \langle v_f, S \rangle$ where

- $n = |\langle bel, s, v_f, X \rangle \in QS_d^t : X \in \{A_1, \ldots A_n\}|$,
- $S = \Phi_{s,CS_d^t}$, i.e., S is the p-proof of s from CS_d^t.

Move Relevance Function The final element of TalkLOG persuasion architecture is *move relevance function* [65]. It is provided to guide coherence and focus of dialogues. As move is a sequence of locutions, relevance of each locution is verified on the basis of dialogue stores' content at the given timepoint t, and the locution content. For ease of presentation, dialogue stores are updated after each relevant locution, so that locution relevance can be defined without reference to preceding locutions in the move. Relevant locutions are listed below.

1. Assertions $assertBel_S^d\langle \psi, v \rangle$ of a belief if it concerns
   (a) the topic of the dialogue,
   (b) a belief asserted by another agent,
   (c) body or head of a rule present in DS, or a fact present in DS.
2. Questions $why_S^d\langle \varphi, v \rangle$ if the belief $\langle \varphi, v \rangle$: $v \in \{\mathtt{t}, \mathtt{f}, \mathtt{i}\}$ is in QS.
3. Assertions $assert_S^d\langle \delta, v_l \rangle$ of a fact or rule if
   (a) a question about a belief concerning $head(\delta)$ is in QS,
   (b) it is present in DS (an assertion works as a concession then).
4. Concessions $concede_S^d\langle \delta, v_l \rangle$ of a fact or a rule present in DS.
5. Retractions $retractBel_S^d\langle \varphi, v \rangle$ of a belief if it is in QS.
6. Adoptions $adopt_S^d\langle \delta \rangle$ of a fact or a rule if $\langle \delta, i \rangle \in RDS$.
7. Rejections $reject_S^d\langle \delta \rangle$ of a fact or a rule if $\langle \delta, o \rangle \in RDS$.

3.4.4 Verified Properties

To analyze termination of dialogues, we distinguished two termination conditions

- **Impasse**: when no agent has a relevant move to make,
- **Common Opinion**: when all agents agree on the value of the topic,

and investigated their relationship with the two persuasion types.

Theorem 11 *Persuasion terminating on Impasse is a Deep Persuasion.*

Theorem 12 *Persuasion terminating on Common Opinion is a Classical Persuasion.*

Theorem 13 *Persuasion dialogues terminate.*

Next, we went on to analyze the outcomes of dialogues. Informally, **soundness** of persuasion means that any conclusion obtained in the dialogue equals the conclusion obtained by a single agent reasoning from a merged belief bases of dialogue participants. On the other hand, **completeness** of persuasion means that any conclusion obtained by reasoning from **merged belief bases** of participants is obtainable by persuasion carried out by these agents. The merging operator $\sum(s)$ reflects the nature of dialogue and is a consensual merge (see [24]) exploiting a voting mechanism (see Definition 21) for conflict resolution.

Definition 24 Persuasion dialogue d on subject s is *sound* iff whenever it terminates at t with conclusion $c = \langle v_f, S \rangle$, then if $s(\mathcal{M}_{CS_d^t}) = v_f$ then $s(\mathcal{M}_{(\sum_{i=1}^n P_i)(s)}) = v_f$.

Definition 25 Persuasion dialogue d on subject s is *complete* iff whenever it terminates at t with conclusion $c = \langle v_f, S \rangle$, then if $s(\mathcal{M}_{(\sum_{i=1}^n P_i)(s)}) = v_f$ then $s(\mathcal{M}_{CS_d^t}) = v_f$.

Merging is an iterative procedure, achieved by joining p-proofs of the merge parameter s and resolving conflicts on the way. The result of merging is a 4QL program defined as follows:

$$\left(\sum_{i=1}^{n} P_i\right)(s) \overset{\text{def}}{=} \bigcup_{IT=0}^{ITMAX} \left(\bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) \right),$$

where for $IT \geq 0$, $k \in Lab$

- $\bigcup \Phi^{IT} \overset{\text{def}}{=} \bigcup_{i=1..n} \Phi_{s, P_i^{IT}}$, meaning the union of all agents' p-proofs for s from their current program in iteration IT;
- $ITMAX \overset{\text{def}}{=} IT : \forall_{i=1..n} : P_i^{IT+1} = P_i^{IT}$, meaning the final iteration after which all agents' programs stop changing;
- $P_i^{IT+1} = P_i^{IT} \cup \bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) \setminus \bigcup_{\delta \in \bigcup \Phi^{IT}} OUT(\delta)$, meaning agent's i program in the next iteration is its program from the previous iteration changed in a way that:
    - those facts and rules from the union of all agents' p-proofs of s from the previous iterations which were accepted are added and
    - those facts and rules from the union of all agents' p-proofs of s from the previous iterations which were rejected are removed;
- $IN(\delta) = \{\delta[a = i], \neg\delta[b = i] : \langle a, b \rangle \in VF(\delta, n_t^\delta, n_f^\delta, n_b^\delta, n_n^\delta, n)\}$, is the set of facts or rules $(\neg)\delta$ to be added to agents' programs;
- $OUT(\delta) = \{\delta[a = o], \neg\delta[b = o] : \langle a, b \rangle \in VF(\delta, n_t^\delta, n_f^\delta, n_b^\delta, n_n^\delta, n)\}$ is the set of facts or rules $(\neg)\delta$ to be removed from agents' programs;
- $n_k^\delta = |\{i \in 1..n : \mu_{P_i^{IT}}(\delta) = k\}|$, meaning the value of support counter for label k for δ in iteration IT.

We define the merge (4QL program) by imitating the creation of the Commitment Store: by adding entries (elements of the union of all agents' p-proofs for the persuasion topic s) as they get accepted via voting. In each iteration, the conflicts in the union of all agents' p-proofs are resolved by voting[14], whose outcomes (sets IN and OUT) are then used to update the programs. The procedure stops naturally when agents' programs stop changing. The proof of s from such a merge is $\Phi_{s,(\sum_{i=1..n} P_i)(s)}$ while the value of the topic s is $s(\mathcal{M}_{(\sum_{i=1}^n P_i)(s)})$.

Theorem 14 *Classical Persuasion is not sound and not complete.*

---

[14] Note P not Φ in the subscript $\mu_{P_i^{IT}}(\delta)$, since one may vote for δ even if absent from the p-proof.

Theorem 15 *Iterated Deep Persuasion is sound and complete.*

Theorem 16 *Deep and Classical Persuasion possibly converges to the merged outcome.*

A possible way to extend this research is an investigation of dialogue strategies of individual agents in the spirit of [48].

3.4.5 Complexity

Like inquiry, the complexity measures of persuasion are divided into two sorts:

- *communication complexity*, concerning only the amount of communication among agents (who have unlimited computational power) [50],
- *computational complexity* (data complexity), concerning the amount of computation (when communication is free) required to:
  - achieve dialogue termination,
  - obtain a conclusion of a terminated dialogue.

Obtaining conclusion of terminated Deep Persuasion is expressed in terms of *data complexity* [61, 88], i.e., complexity of evaluating a fixed query (here: persuasion goal) on an arbitrary database (here: Commitment Store). It amounts to computing the well-supported model of CS_d^t just once, at the end of dialogue. Thus, this problem is in $O(N^k)$ where N is the size of domain and k is the maximal arity of relations.

Obtaining conclusion of terminated Classical Persuasion amounts to counting the occurrences of agents' beliefs concerning the dialogue topic in Query Store, which holds agents' expressed opinions, thus it is in $O(nN) = |QS_d^t|$ (as QS_d^t may possibly contain all beliefs of all n agents at termination time). When implemented with a dedicated data structure it can be in $O(1)$[15].

3.4.6 Conclusions

To meet the requirements of information-rich environments as well as demands of realistic modeling of agency, we presented a formalization of multi-party, paraconsistent and paracomplete persuasion in nonmonotonic, dynamic TalkLOG setting, concluding the following.

1. Two additional logical values ensure a better discernment between initial opinions.
2. Paraconsistent framework admits a new interpretation of conflict of opinion, based on the inequality of truth values of literals representing the opinion, rather than on inconsistency.
3. In our Classical Persuasion the novelty lies in more refined types of initial conflicts of opinion and in permitting an arbitrary number of participants.
4. Reaching to motivations of opinions in our Deep Persuasion allows to study semantically deeper conflicts. This new type of persuasion, when successful, leads to a common motivation entailing a common opinion in the group.
5. TalkLOG framework permits to obtain a uniform treatment of both persuasion types, which are differentiated by the termination criterion solely.
6. Conflict of opinion is typically a subject of argumentation, while as a novelty in TalkLOG:
   - agents argue about beliefs with the use of pieces of evidence,
   - when conflict of evidence appears, solutions from different scientific fields can be of help (like different voting mechanisms from computational social choice theory).

---

[15] For example, when maintaining counters for each type of opinion (truth value) on the dialogue subject, it suffices to test whether all counters but one are 0 and one equals to the number of dialogue participants.

7. TalkLOG persuasion architecture builds upon TalkLOG inquiry architecture and deals with four dialogue stores, retaining the effects of agents' dialogue moves and resolved conflicts.
8. TalkLOG architecture allowed to obtain a protocol with public semantics, suitable for an arbitrary number of participants holding possibly different initial opinions or motivations.
9. Our methodology allows one to provide:
    - a dedicated voting mechanism as a method for information fusion in persuasion,
    - customization of the complex merge operator with the designed voting mechanism, for analyzing important properties of dialogues,
    - analysis of termination, soundness and completeness of persuasion,
    - communication and complexity results of persuasion,
    - limitations on the dialogue store sizes.
10. The dynamic model of TalkLOG's persuasion encompasses agents' beliefs change in the course of dialogue according to the state of the Resolved Dispute Store.

4 Final Conclusions

This research was set out to explore the fundamental problems occurring in communication in multiagent systems situated in information-rich settings with an emphasis on:

- realistic modeling,
- low computational complexity,
- obtaining desirable, provable properties of proposed solutions,
- dynamism of communication as an inherent element of dialogue.

Indeed, in practical applications agents do not freeze their beliefs when engaging in communication: they acquire information *from and about interlocutors*, which may be further exploited in the course of dialogue. Verifying properties of dialogues is important for several reasons. First, it allows to verify properties of larger agents systems that use the proposed dialogues at various phases of team formation or group action. Next, it proves whether the theoretical solutions are suitable for practical applications. Therefore, when creating models of complex dialogues, such as inquiry and persuasion, it is important to obtain protocols with proven properties as opposed to ad-hoc dialogues. Indeed, soundness and completeness relate the dialogue outcomes to the conclusions reached by well-defined merging of interlocutors' belief bases.

To our best knowledge this dissertation is the first such advanced research on agent communication in the paraconsistent and paracomplete 4-valued setting. The obtained results allow to conclude that exploration of this new direction was fruitful. Importantly, as the tractability requirement guided our approach, flexible and expressive dialogues that are not computationally complex were obtained. As modeling the environment and phenomena appearing in multiagent systems in the new perspective proved to be legitimate, contemporary, reverting to the 2-valued logics seems unreasonable.

References

[1] J. J. Alferes, A. Brogi, J. A. Leite, and L. M. Pereira. Evolving logic programs. In *Proceedings of JELIA 2002*, volume 2424 of *LNCS*, pages 50–61. Springer, 2002.

[2] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152:157–206, 2005.

[3] J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, second edition, 1975. Edited by J. O. Urmson and M. Sbisa.

[4] N. Belnap. A useful four-valued logic. In G. Epstein and J. Dunn, editors, *Modern Uses of Many Valued Logic*, pages 8–37. Reidel, 1977.

[5] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[6] E. Black. *A generative framework for argumentation-based inquiry dialogues.* PhD thesis, University College London, 2007.

[7] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[8] E. Bonzon and N. Maudet. On the outcomes of multiparty persuasion. In P. McBurney, S. Parsons, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems*, volume 7543 of *LNCS*, pages 86–101. Springer Berlin Heidelberg, 2012.

[9] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.

[10] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. *Multiagent systems*, pages 213–283, 2012.

[11] M. Bratman. *Intention, plans, and practical reason*. Harvard University Press, Cambridge, MA, 1987.

[12] K. Budzyńska, B. Dunin-Kęplicz, A. Strachocka, et al. The polish school of argumentation: A manifesto. *Argumentation*, 28(3):267–282, 2014.

[13] K. Budzyńska, M. Kacprzak, and P. Rembelski. Perseus. software for analyzing persuasion process. *Fundamenta Informaticae*, 93(1-3):65–79, 2009.

[14] P. G. C. Alchourrón and D. Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50:510–530, 1985.

[15] C. Castelfranchi. The social nature of information and the role of trust. *International Journal of Cooperative Information Systems*, 11(3):381–403, 2002.

[16] C. Castelfranchi, F. Dignum, C. M. Jonker, and J. Treur. Deliberate normative agents: Principles and architecture, 1999.

[17] A. K. Chopra, A. Artikis, J. Bentahar, M. Colombetti, F. Dignum, N. Fornara, A. J. I. Jones, M. P. Singh, and P. Yolum. Research directions in agent communication. *ACM TIST*, 4(2):20, 2013.

[18] A. Ciampolini, E. Lamma, P. Mello, F. Toni, and P. Torroni. Cooperation and competition in alias: a logic framework for agents that negotiate. *Annals of Mathematics and Artificial Intelligence*, 37(1-2):65–91, 2003.

[19] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. Technical Report 433, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1988.

[20] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[21] P. R. Cohen and H. J. Levesque. Performatives in a rationally based speech act theory. In *In Meeting of the Association for Computational Linguistics*, pages 79–88, 1990.

[22] D. D. Corkill. Blackboard systems. *AI Expert*, 6:40–47, 1991.

[23] S. Costantini and A. Tocchio. Learning by knowledge exchange in logical agents. In *Proc. of WOA 2005: Dagli*, 2005.

[24] S. Coste-Marquis, C. Devred, S. Konieczny, M. Lagasquie-Schiex, and P. Marquis. On the merging of dung's argumentation systems. *Artif. Intell.*, 171(10-15):730–753, 2007.

[25] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9:145–158, 2001.

[26] F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In F. Dignum, editor, *Workshop on Agent Communication Languages*, volume 2922 of *LNCS*, pages 212–230. Springer, 2003.

[27] D. Dubois. On ignorance and contradiction considered as truth-values. *Logic Journal of the IGPL*, 16(2):195–216, 2008.

[28] B. Dunin-Kęplicz and A. Strachocka. Perceiving rules under incomplete and inconsistent information. In *Computational Logic in Multi-Agent Systems*, volume 8143 of *LNCS*, pages 256–272. Springer Berlin Heidelberg, 2013.

[29] B. Dunin-Kęplicz and A. Strachocka. Computationally-friendly argumentation schemes. In *International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM*, volume 3, pages 167–174, Aug 2014.

[30] B. Dunin-Kęplicz, A. Strachocka, A. Szałas, and R. Verbrugge. Perceiving speech acts under incomplete and inconsistent information. In *KES-AMSTA*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 255–264. IOS Press, 2013.

[31] B. Dunin-Kęplicz, A. Strachocka, and R. Verbrugge. Deliberation dialogues during multi-agent planning. In *Foundations of Intelligent Systems - 19th International Symposium, ISMIS 2011, Warsaw, Poland, June 28-30, 2011. Proceedings*, volume 6804 of *LNCS*, pages 170–181. Springer, 2011.

[32] B. Dunin-Kęplicz and A. Szałas. Epistemic profiles and belief structures. In *6th KES AMSTA International Conference on Agent and Multi-Agent Systems. Technologies and Applications*, volume 7327 of *LNCS*, pages 360–369. Springer-Verlag, 2012.

[33] B. Dunin-Kęplicz and A. Szałas. Taming complex beliefs. *Transactions on Computational Collective Intelligence XI*, LNCS 8065:1–21, 2013.

[34] B. Dunin-Kęplicz, A. Szałas, and R. Verbrugge. Tractable reasoning about group beliefs. In *2nd international Workshop on Engineering Multi-Agent Systems (EMAS 2014)*, LNAI. Springer Berlin Heidelberg, 2014.

[35] B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley, Chichester, 2010.

[36] B. Dunin-Kęplicz and A. Strachocka. Paraconsistent multi-party persuasion in TalkLOG. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, volume 9387 of *LNCS*, pages 265–283. Springer, 2015.

[37] B. Dunin-Kęplicz and A. Strachocka. Tractable inquiry in information-rich environments. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 53–60. AAAI Press, 2015.

[38] B. Dunin-Kęplicz and A. Strachocka. Paraconsistent argumentation schemes. In *Web Intelligence*, volume 14, pages 43–65. IOS Press, 2016.

[39] B. Dunin-Kęplicz, A. Strachocka, A. Szałas, and R. Verbrugge. Paraconsistent semantics of speech acts. *Neurocomputing*, 151:943–952, 2015.

[40] P. E. Dunne and P. McBurney. Optimal utterances in dialogue protocols. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 608–615, New York, NY, USA, 2003. ACM.

[41] M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz. Complexity issues in multiagent logics. *Fundamenta Informaticae*, 75(1-4):239–262, 2007.

[42] R. Englemore and A. Morgan. *Blackboard Systems; Edited by Robert Engelmore, Tony Morgan (the Insight Series in Artificial Intell*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1988.

[43] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[44] FIPA, 2002. `http://www.fipa.org/`.

[45] M. Fisher. Representing and executing agent-based systems. In *Proceedings of the ECAI- 94 Workshop on Agent Theories, Architectures, and Languages*, pages 307–324, 1994.

[46] M. Gelfond and Y. Kahl. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents - The Answer-Set Programming Approach*. Cambridge University Press, 2014.

[47] M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. In *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, ATAL '98, pages 1–10, London, UK, UK, 1999. Springer-Verlag.

[48] M. Kacprzak, M. Dziubiński, and K. Budzyńska. Strategies in dialogues: A game-theoretic approach. In *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 333–344. IOS Press, 2014.

[49] D. Kontarinis, E. Bonzon, N. Maudet, and P. Moraitis. Regulating multiparty persuasion with bipolar arguments: Discussion and examples. In *Modles Formels de l'interaction (MFI'11)*, 2011.

[50] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, New York, NY, USA, 1997.

[51] Y. Labrou and T. Finin. Semantics and conversations for an agent communication language. In M. N. Huhns and M. P. Singh, editors, *Readings in Agents*, pages 235–242. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[52] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *AAAI*, pages 94–99. AAAI Press / The MIT Press, 1990.

[53] W. Lorkiewicz, R. Kowalczyk, R. Katarzyniak, and Q. Bao Vo. On topic selection strategies in multi-agent naming game. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*, pages 499–506. IFAAMAS, 2011.

[54] F. Macagno and D. Walton. Reasoning from paradigms and negative evidence. *Pragmatics & Cognition*, 19(1):92–116, 2011-01-01T00:00:00.

[55] J. Małuszyński and A. Szałas. Logical foundations and complexity of 4QL, a query language with unrestricted negation. *Journal of Applied Non-Classical Logics*, 21(2):211–232, 2011.

[56] J. Małuszyński and A. Szałas. Partiality and inconsistency in agents' belief bases. In *KES-AMSTA*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 3–17. IOS Press, 2013.

[57] P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22(1):95–132, 2007.

[58] P. Mcburney and S. Parsons. Tenacious tortoises: A formalism for argument over rules of inference. In *Computational Dialectics (ECAI 2000 Workshop*, 2000.

[59] P. McBurney and S. Parsons. Chance discovery using dialectical argumentation. In *New Frontiers in Artificial Intelligence*, volume 2253 of *LNCS*, pages 414–424. Springer, 2001.

[60] P. McBurney and S. Parsons. A denotational semantics for deliberation dialogues. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 86–93, Washington, DC, USA, 2004. IEEE Computer Society.

[61] C. H. Papadimitriou and Y. Mihalis. On the complexity of database queries. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* ACM, 1997.

[62] S. Parsons, K. Atkinson, K. Z. Haigh, K. N. Levitt, P. McBurney, J. Rowe, M. P. Singh, and E. Sklar. Argument schemes for reasoning about trust. In *COMMA*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 430–441. IOS Press, 2012.

[63] S. Parsons and P. McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12:415–439, 2003.

[64] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Log. Comput.*, 13(3):347–376, 2003.

[65] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15:1009–1040, 2005.

[66] H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.

[67] H. Prakken. Models of persuasion dialogue. In *Argumentation in Artificial Intelligence*, pages 281–300. Springer US, 2009.

[68] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[69] H. Prakken. On the nature of argument schemes. In *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning*. College Publications, 2010.

[70] G. Priest. Paraconsistent belief revision. *Theoria*, 67(3):214–228, 2001.

[71] A. S. Rao. Agentspeak(l): Bdi agents speak out in a logical computable language. In *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-agent World : Agents Breaking Away: Agents Breaking Away*, MAAMAW '96, pages 42–55, Secaucus, NJ, USA, 1996. Springer-Verlag New York, Inc.

[72] A. S. Rao and M. P. Georgeff. Modeling rational agents within a bdi-architecture, 1991.

[73] A. S. Rao, M. P. Georgeff, and E. A. Sonenberg. Social plans: A preliminary report, 1992.

[74] C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agent Systems*, 11(2):173–188, 2005.

[75] U. Schöning. *Logic for Computer Scientists*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

[76] J. Searle and D. Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, 1985.

[77] M. P. Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, Dec. 1998.

[78] M. P. Singh. A social semantics for agent communication languages. In *Proceedings of the IJCAI Workshop on Agent Communication Languages*. Springer-Verlag, 2000.

[79] A. Szałas. How an agent might think. *Logic J. IGPL*, 21(3):515–535, 2013.

[80] T. Takahashi and H. Sawamura. A logic of multiple-valued argumentation. In *AAMAS*, pages 800–807. IEEE Computer Society, 2004.

[81] K. Tanaka. The AGM theory and inconsistent belief change. *Logique Et Analyse*, 48(189-192):113–150, 2005.

[82] D. Traum. Issues in multiparty dialogues. *Advances in Agent Communication*, 2004.

[83] D. Traum, J. Gratch, A. Hartholt, S. C. Marsella, and J. Lee. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In *Proceedings of the 8th International Conference on Intelligent Virtual Agents*, pages 117–130, Tokyo, Japan, Sept. 2008.

[84] D. Traum and J. Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Bologna, Italy, July 2002.

[85] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[86] B. van Linder, W. van der Hoek, and J.-C. Meyer. Formalizing abilities and opportunities of agents. *Fundamenta Informaticae*, 34(1-2):53–101, 1998.

[87] B. van Linder, W. van der Hoek, and J.-J. Meyer. Actions that make you change your mind. In *KI-95: Advances in Artificial Intelligence*, volume 981 of *LNCS*, pages 185–196. Springer Berlin Heidelberg, 1995.

[88] M. Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 137–146, New York, NY, USA, 1982. ACM.

[89] A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.

[90] G. Vreeswijk and J. Hulstijn. A free-format dialogue protocol for multi-party inquiry. In *In Proc. of the Eighth Int. Workshop on the Semantics and Pragmatics of Dialogue (Catalog '04)*, pages 273–279, 2004.

[91] D. Walton. Ethotic arguments and fallacies: The credibility function in multi-agent dialogue systems. *Pragmatics and Cognition*, 7(1):177–203, 1999.

[92] D. Walton. Types of dialogue and burdens of proof. In *Proceedings of the 2010 Conference on Computational Models of Argument: Proceedings of COMMA 2010*, pages 13–24, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[93] D. Walton. Reasoning about knowledge using defeasible logic. *Argument & Computation*, 2(2-3):131–155, 2011.

[94] D. Walton and E. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.

[95] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[96] M. Wardeh, T. J. M. Bench-Capon, and F. Coenen. Multi-party argument from experience. In *Argumentation in Multi-Agent Systems, 6th International Workshop, ArgMAS 2009, Budapest, Hungary, May 12, 2009. Revised Selected and Invited Papers*, pages 216–235, 2009.

[97] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009.

[98] M. Wooldridge and N. Jennings. The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.

[99] M. Wooldridge and N. R. Jennings. The cooperative problem solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.

[100] J. Yuan, L. Yao, Z. Hao, F. Liu, and T. Yuan. Multi-party dialogue games for distributed argumentation system. In *IAT*, pages 329–332. IEEE Computer Society, 2011.

Paraconsistent Semantics of Speech Acts

Barbara Dunin-Kęplicz[a], Alina Strachocka[a], Andrzej Szałas[a,b],
Rineke Verbrugge[c]

[a] Institute of Informatics, Warsaw University, Poland
[b] Dept. of Computer and Information Science, Linköping University, Sweden
[c] Institute of Artificial Intelligence, University of Groningen, The Netherlands

Abstract

This paper discusses an implementation of four speech acts: *assert*, *concede*, *request* and *challenge* in a paraconsistent framework. A natural four-valued model of interaction yields multiple new cognitive situations. They are analyzed in the context of *communicative relations*, which partially replace the concept of trust. These assumptions naturally lead to six types of situations, which often require performing conflict resolution and belief revision.

The particular choice of a rule-based, DATALOG$^{\neg\neg}$-like query language 4QL as a four-valued implementation framework ensures that, in contrast to the standard two-valued approaches, tractability of the model is achieved.

Keywords: paraconsistent modeling, communication, speech acts, tractable models, four-valued logic, conflict resolution

1. A Four-valued Formalism in Modeling Speech Acts

The development of multiagent systems (MAS) demands an adequate modeling of the environment and agents involved. Suitable knowledge representation methods should be selected on an application-specific basis. When confining to logic-based approaches and formalisms, traditionally two-valued logics prevail. They fail, however, to express in a natural manner richer modeling aspects when some values or properties are simply unknown, or when the available information is inconsistent. A natural remedy for such situations is introducing four logical values. This work aligns with a whole line of research concerning logical modeling, reasoning and communicating about the surrounding environment, under the assumption that we deal with four types of situations, encoded in the four logical

Preprint submitted to Elsevier *March 4, 2016*

values: (i) fact *a* holds, (ii) fact *a* does not hold, (iii) it is unknown whether *a* holds, (iv) information about *a* is inconsistent.

This paper continues our research program on paraconsistent modeling of communication in the four-valued framework. In real-world systems with many information sources, lack and inconsistency of information is a rule rather than exception. To model such phenomena, a commonly used logic is the four-valued logic proposed in [4]. However, as discussed, e.g., in [11, 34], the approach of [4] is problematic. In fact, in areas we focus on it often delivers results deviating from intuitions (see [11, 34] and Example 1 for details). Our approach is strongly influenced by ideas underlying the 4QL query language [23, 24] which does not share such problems.

We consider argumentation-based dialogues as communicative games between two or more agents and treat the sender and the receiver as two independent information sources, which try to expand, update, and revise their beliefs through communication. Instead of adopting the computationally hard theory of trust [8], we consider three communicative relations between the agents involved: *communication with authority*, *peer to peer communication* and *communication with subordinate*.

The complex dialogues that we investigate, such as persuasion, deliberation, information seeking, negotiation or inquiry are composed of so-called *speech acts* (see [1, 7, 10, 13, 18, 27, 29] for investigations in multi-agent argumentation-based dialogue, and [35] for the definitions of various dialogue types). Austin's observation that some utterances cannot be verified as true or false [2] led to the first division of speech acts into *constatives*, which can be assigned a logical truth value, and the remaining group of *performatives*. Searle created their most popular taxonomy, identifying: *assertives*, *directives*, *commissives*, *expressives* and *declaratives* [30]. Austin specified also their effects on the attitudes and actions of the hearer. Various speech acts, viewed as *typical actions*, can be represented in dynamic logic, by characterizing their pre- and post-conditions. We define them in terms of the changes in agents' beliefs and actions (see also [10, 18, 13]).

There have been many approaches to defining semantics of speech acts [27, 1], some based on Belnap's four-valued logic [21]. Still, some researchers view them as primitive notions [28]. Within the most popular *mentalistic* approach, reflected in languages such as KQML and FIPA ACL, speech acts are defined through their impact on agents' mental attitudes. The current paper clearly falls in that approach.

The four-valued, natural model of interaction yields multiple new cognitive situations. Therefore we distinguish six interaction types and analyze them one by one, providing a semantics of four selected speech acts: *assert*, *concede*, *request* and *challenge*. Our semantics is expressed in terms of triples consisting of *preconditions*, *speech acts* and *complex post actions*. Along with defining rules for perceiving speech acts, we indicate their detailed impact on the receiver's informational stance.

To demonstrate perceiving speech acts in the four-valued framework, throughout this paper we use a running example that concerns a rescue-agent (receiver) capable of putting out fire and detoxicating an area (see the 4QL module illustrated in Figure 1 which reflects the situation, where the agent is aware of the fire but due to faulty sensors cannot sense the heat). For simplicity, we will focus on the case when the sender S is an authority and the receiver R is a subordinate.

This paper is an extended version of our conference paper [12]. Comparing to [12], we:

- elaborate on the group beliefs achieved via communication;
- expand upon belief revision in 4QL;
- more directly integrate our approach with belief structures;
- revise and extend discussions and examples.

The paper is structured as follows. Section 2 is devoted to a four-valued logic used throughout the paper and provides basic information on 4QL. Section 3 discusses preliminary solution details. Section 4 contains the main technical contribution, illustrated by an example in Section 5. Section 6 concludes the paper.

2. 4QL: an Implementation Tool

In what follows all sets are finite except for sets of formulas.

We deal with the classical first-order language over a given vocabulary without function symbols and assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols. We shall use this notation in the following definitions.

Definition 1. A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with $\bar{\tau} \in (Const \cup Var)^k$, where k is the arity of R. *Ground literals over Const*, denoted by $\mathscr{G}(Const)$, are literals without variables, with all constants in *Const*. $\triangleleft$

If $\ell = \neg R(\bar{\tau})$ then $\neg \ell \overset{\text{def}}{=} R(\bar{\tau})$. Let $v : Var \longrightarrow Const$ be a *valuation of variables*. For a literal ℓ, by $\ell(v)$ we mean the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$. The semantics of propositional connectives is summarized in Table 1.

Table 1: Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$ (see [34, 23, 24]).

$\wedge$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\vee$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\rightarrow$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$		$\neg$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$

Note that the definitions of $\wedge$ and $\vee$ reflect minimum and maximum w.r.t. the ordering:

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}. \tag{1}$$

The ordering (1) differs from orderings used in Belnap-like approaches. The following example of [34] shows that in application areas we deal with these approaches lead to conclusions deviating from intuitions.

Example 1. Assume a family owns two cars: c_1 and c_2. The question, whether the family has a safe car corresponds to the logical value of the expression

$$safe(c_1) \vee safe(c_2).$$

Car c_1 has gone through safety tests at two different stations s_1 and s_2. It has passed the safety tests at s_1 but failed the tests at s_2. Car c_2 has not gone through any safety test yet. The results of the tests determine the truth value of $safe(c_1)$ to be $\mathfrak{i}$ while of $safe(c_2)$ to be $\mathfrak{u}$. If the join operation $\vee$ is defined by Belnap's truth ordering, then

$$safe(c_1) \vee safe(c_2) = \mathfrak{i} \vee \mathfrak{u} = \mathfrak{t}.$$

However, the safety of car c_1 is unclear, since the results of both safety tests are contradictory, and we know nothing about safety of car c_2! A more intuitive result here would be $\mathfrak{i}$.

Asking instead, if all cars of the family are safe, $safe(c_1) \wedge safe(c_2)$, evaluates to $\mathfrak{f}$ in Belnaps logic ($\mathfrak{i} \wedge \mathfrak{u}$). However, actually we do not have any information about the safety of car c_2. If in reality it would have failed the safety tests then the expression above would evaluate to $\mathfrak{f}$. But, if car c_2 would have passed the tests then the expression would become $\mathfrak{i}$. Therefore, the above case seems to be better described by $\mathfrak{u}$ than by the answer obtained in the Belnaps logic. $\lhd$

Definition 2. The *truth value* of a literal ℓ w.r.t. a set of ground literals L and valuation v, denoted by $\ell(L,v)$, is defined as follows:

$$\ell(L,v) \overset{\text{def}}{=} \begin{cases} \mathfrak{t} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases} \qquad \lhd$$

For a formula $\alpha(x)$ with a free variable x and $c \in Const$, by $\alpha(x)^x_c$ we understand the formula obtained from α by substituting all free occurrences of x by c. Definition 2 is extended to all formulas in Table 2, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 1.

Table 2: Semantics of first-order formulas.

- if α is a literal then $\alpha(L,v)$ is defined in Definition 2;
- $(\neg\alpha)(L,v) \overset{\text{def}}{=} \neg(\alpha(L,v))$;
- $(\alpha \circ \beta)(L,v) \overset{\text{def}}{=} \alpha(L,v) \circ \beta(L,v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x \alpha(x))(L,v) = \min\limits_{a \in Const} (\alpha^x_a)(L,v)$,
  where min is the minimum w.r.t. ordering (1);
- $(\exists x \alpha(x))(L,v) = \max\limits_{a \in Const} (\alpha^x_a)(L,v)$,
  where max is the maximum w.r.t. ordering (1).

It is worth noting that whenever truth values are restricted to $\{\mathfrak{f}, \mathfrak{t}\}$, the semantics we consider is compatible with the semantics of classical first-order logic.

From several languages designed for programming BDI agents (for a survey see, e.g., [25]), none directly addresses belief formation, in particular nonmonotonic or defeasible reasoning techniques. 4QL enjoys tractable query computation and captures all tractable queries. It supports a modular and layered architecture, providing simple, yet powerful constructs for expressing nonmonotonic rules reflecting "lightweight" versions of many known formalisms [23]. 4QL applies the Open World Assumption, however the world can be (locally or globally) closed using rules.

In 4QL rules are distributed among modules, where a layered architecture is required.[1] A *layer* is a set of modules. We also assume a partial order $\preceq$ on modules, where $m \preceq n$ means that m is in a layer lower than n. An *external literal* is an expression of one of the forms:

$$m.R,\ \neg m.R,\ m.R \text{ IN } T,\ \neg m.R \text{ IN } T,$$

where m is a module name, R is a positive literal and $T \subseteq \{\mathfrak{f}, \mathfrak{u}, \mathfrak{i}, \mathfrak{t}\}$. The intended meaning of $m.R \text{ IN } T$ is that the truth value of $m.R$ is in the set T. If R is not defined in the module m then the value of $m.R$ is $\mathfrak{u}$. The use of external literals is restricted by the following requirement:

> an external literal $m.\ell'$ may only appear in a module n, provided that $m \preceq n$.

Definition 3. By a *rule* we mean any expression of the form:

$$\ell :- b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}. \tag{2}$$

where ℓ is a (positive or negative) literal, called *conclusion of the rule* and *rule's premisses* $b_{11}, \ldots, b_{1i_1}, \ldots, b_{m1}, \ldots, b_{mi_m}$ are (positive or negative) literals or external literals or truth values. $\triangleleft$

In rules ',' and '|' abbreviate conjunction and disjunction, respectively. By a *fact*, denoted by 'ℓ.', we mean a rule of the form '$\ell :- \mathfrak{t}$.'.

A 4QL *program* is a finite set of modules, each of which contains a finite number of rules.

---

[1] The layered architecture can be relaxed as in [31].

Definition 4. Let a set of constants, *Const*, be given. A set of ground literals L with constants in *Const* is a *model of a set of rules S* iff for each rule (2) and any valuation v mapping variables into constants in *Const*, we have that:

$$\big(((b_{11} \wedge \ldots \wedge b_{1i_1}) \vee \ldots \vee (b_{m1} \wedge \ldots \wedge b_{mi_m})) \rightarrow \ell\big)(L,v) = \mathsf{t},$$

where it is assumed that the empty antecedent is t in any interpretation. $\lhd$

The semantics of 4QL is defined by well-supported models [23, 24]. Intuitively, a model is *well-supported* if it consists of ground literals such that each literal in the model is supported by a reasoning grounded in facts. For any 4QL module there is a unique such model and it can be computed in polynomial time [24]. That is, any set of 4QL rules and facts can be understood as its well-supported model being a finite set of ground literals.

```
module r:
  relations: is(literal).
             do(literal).
  rules:
    is(risky) :- is(poison) | is(heat).
    -is(risky) :- -is(poison), -is(heat).
    is(heat) :- is(fire).
    -is(poison) :- is(safe).
    is(busy) :- do(detox) | do(extinguish).
    -do(detox) :- is(heat).
    -is(heat) :- is(lowtemp).
    -is(fire) :- is(lowtemp).
    is(X) :- perceived.is(X).
    -is(X) :- -perceived.is(X).
    do(extinguish) :- is(heat) | is(fire).
    -do(extinguish) :- -is(heat), -is(fire).
end.
module perceived:
  relations:   is(literal).
  facts:
    is(fire).
    -is(heat).
end.
```

Figure 1: Example of a 4QL program.

To illustrate 4QL, consider Figure 1, where we use self-explanatory syntax of the 4QL interpreter inter4QL.[2] The program shown in Figure 1 consists of two

---

[2] The interpreter, initiated by P. Spanily and further developed by Ł. Białek, can be downloaded from http://www.4ql.org/ .

modules: r and *perceived*[3]. The program uniquely determines the following well-supported model for module *perceived*:

$$\{is(fire), \neg is(heat)\} \tag{3}$$

and the following well-supported model for module r:

$$\begin{gathered}\{is(heat), \neg is(heat), is(fire), is(busy), is(risky), \\ do(detox), \neg do(detox), do(extinguish)\}.\end{gathered} \tag{4}$$

3. Introductory Solution Details

3.1. Epistemic Profiles and Communicative Relations

An essential question is how to realize heterogeneity of agents in multiagent systems. Clearly, being different, when seeing the same thing the agents may draw different conclusions. A notion of *epistemic profile* [14] explicitly models this problem.[4] First, it defines the way an agent reasons (e.g., by the use of rules). Second, it permits expressing the granularity of reasoning (e.g., by varying the level of certain attributes or accuracy of rules expressing the modeled phenomena). Third, it also characterizes the manner of dealing with conflicting or lacking information by combining various forms of reasoning, including belief fusion, disambiguation of conflicting beliefs or completion of lacking information. The following definitions are adapted from [14], where also more intuition and examples can be found.

If S is a set then by $\textsc{Fin}(S)$ we understand the set of all finite subsets of S.

Definition 5. Let $\mathbb{C} \overset{\text{def}}{=} \textsc{Fin}(\mathscr{G}(Const))$ be the set of all finite sets of ground literals over the set of constants *Const*. Then:

- by a *constituent* we understand any set $C \in \mathbb{C}$;
- by an *epistemic profile* we understand any function $\mathscr{E} : \textsc{Fin}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over an epistemic profile* $\mathscr{E}$ we mean $\mathscr{B}^{\mathscr{E}} = \langle \mathscr{C}, F \rangle$, where:
    - $\mathscr{C} \subseteq \mathbb{C}$ is a nonempty set of constituents;

---

[3] For a detailed discussion of this example see Section 5.

[4] Belief structures are further developed in [16] but here we use the deterministic version.

- $F \overset{\text{def}}{=} \mathscr{E}(\mathscr{C})$ is the *consequent* of $\mathscr{B}^{\mathscr{E}}$. $\triangleleft$

For example, Figure 1 presents two modules: r and *perceived*. These modules define sets of ground literals (3) and (4) and can be seen as a belief structure with one constituent *perceived* and consequent r. The epistemic profile is defined by rules of module r.

As well-supported models are sets of ground literals (represented by 4QL programs with deterministic polynomial time data complexity), we alternate between the notions of the set of consequents of 4QL programs and well-supported models.

To express beliefs we introduce operator $Bel()$.

Definition 6. Let $\mathscr{E}$ be an epistemic profile. The *truth value of formula* α w.r.t. belief structure $\mathscr{B}^{\mathscr{E}} = \langle \mathscr{C}, F \rangle$ and valuation v, denoted by $\alpha(\mathscr{B}^{\mathscr{E}}, v)$, is defined by $\alpha(\mathscr{B}^{\mathscr{E}}, v) \overset{\text{def}}{=} \alpha(\bigcup\limits_{C \in \mathscr{C}} C, v)$. Beliefs are evaluated in F: $Bel(\alpha)(\mathscr{B}^{\mathscr{E}}, v) \overset{\text{def}}{=} \alpha(F, v)$. $\triangleleft$

Note that the semantics of $Bel()$ differs from the one given in [14, 15]. Namely, In [14, 15], $Bel(\mathfrak{u}) = \mathfrak{f}$, while in Definition 6 we assume that $Bel(\mathfrak{u}) = \mathfrak{u}$. The choice we made here is better suited to model nuances of agent communication and the general matter of our paper.

Of course, in the case of many agents and groups, we may have separate belief structures for these agents/groups and, consequently, we may have many $Bel()$ operators related to belief structures. We do not introduce them formally, as in the rest of the paper we only deal with agents' beliefs, and to simplify notation we skip belief operators, assuming that they are implicitly present and the context will always uniquely indicate respective consequents.

Agents may react variously to the perceived information. We distinguish three types of *communicative relations*, considered from the receiver's perspective:

1. *communication with authority*: an agent (receiver) is prone to adopt the interlocutor's (sender, authority) theses. In strong disagreement, instead of abandoning its beliefs totally, it would rather investigate the reasons of the conflict, but in case of a mere discrepancy of opinions it would give up on its prior beliefs;
2. *peer to peer communication*: both parties are viewed as equally credible and important information sources, therefore nobody's opinion prevails a priori. This shows up when dealing with inconsistent information, which taints everything: whenever one party believes a proposition is inconsistent, the

other party's prior beliefs do not matter. The peer, upon receiving information that introduces inconsistency to its beliefs, is obliged to comply with it. Therefore every discrepancy of opinions boils down to inconsistency,

3. *communication with subordinate*: when dealing with a less reliable source of information, the receiver with authority would not be willing to abandon its beliefs in favor of the interlocutor's. It would value its observations higher and protect true and false propositions from being infected by inconsistency. However, in case of strong disagreements, it engages in conflict resolution.

3.2. Belief Revision

Introducing inconsistency as a first-class citizen entails a need of **conflict resolution**. Namely, perceiving new information, whether it is some previously unknown fact or a new valuation of a proposition, may require revising beliefs. The communicative relation functions as a filter, determining when a new percept triggers **belief revision**. Importantly, such a belief revision, when expressed in 4QL, can be performed in polynomial time. We distinguish two types of conflicts:

- *strong disagreements*: when agents have contradictory opinions;
- *mere discrepancies*: when one agent believes that an opinion is inconsistent, and the other believes it is true or false.

Although a broader treatment of general belief revision properties in 4QL is needed, here we intend to address the particular topics raised in this article. From the speech acts treated here, only two may cause agents to make some changes in their beliefs: assertions and concessions. The first one triggers a change in agent's individual beliefs, while the second in agent's group beliefs (see Section 3.3).

In case of assertions, the receiver agent might need to change it's valuation of a literal according to the rules outlined in Section 4.1. The valuation of the literal in question is obtained from the set of consequents, here implemented via the well-supported model of the 4QL program. Belief revision strategies, as expressed in 4QL, may vary from conservative to more drastic ones. In all cases, belief revision amounts to computing a new well-supported model on the basis of a refreshed set of facts and rules.

Assume that we want literal ℓ to become true (recall that ℓ may be positive as well as negative). The *conservative* approach depends on adding ℓ as a fact and temporarily blocking rules contradicting ℓ. These changes affect all constituents and the consequent of the revised belief structure. In short, the procedure looks as follows, where we assume that all rules containing variables are replaced by their ground instances (but see Remark 1):

1. adding ℓ to the set of facts (if not already there);
2. removing the fact contradicting the new valuation of ℓ from the set of facts (if any);
3. blocking rules contradicting the new valuation of ℓ (if such rules exist);
4. recomputing the well-supported model.

The details for the steps 1, 2 and 3 are presented in the Table 3. The original valuations of literal ℓ are in the first column, in the first row are the new valuations for the literal ℓ. Notice, that in our framework we do not allow for known literals to become unknown, therefore there are only three columns for the new valuations.

Table 3: Revising beliefs (G represents constituents and consequent of a revised belief structure)

ℓ	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{i}$
$\mathfrak{t}$	-	$G' \leftarrow G \cup \{\neg\ell\}$ $G' \leftarrow G - \{\ell\}$ block rules with conclusion ℓ	$G' \leftarrow G \cup \{\neg\ell\}$
$\mathfrak{f}$	$G' \leftarrow G \cup \{\ell\}$ $G' \leftarrow G - \{\neg\ell\}$ block rules with conclusion $\neg\ell$	-	$G' \leftarrow G \cup \{\ell\}$
$\mathfrak{i}$	$G' \leftarrow (G - \{\neg\ell\}) \cup \{\ell\}$ block rules with conclusion $\neg\ell$	$G' \leftarrow (G - \{\ell\}) \cup \{\neg\ell\}$ block rules with conclusion ℓ	-
$\mathfrak{u}$	$G' \leftarrow G \cup \{\ell\}$ block rules with conclusion $\neg\ell$	$G' \leftarrow G \cup \{\neg\ell\}$ block rules with conclusion ℓ	$G' \leftarrow G \cup \{\neg\ell, \ell\}$

The *drastic* strategy depends on removing the contradicting rule rather than blocking it. Note that removing the rule with, for example, $\neg\ell$ as its conclusion is harmless for possible further belief revisions making $\neg\ell$ true. In such cases one adds $\neg\ell$ to the set of facts (step 1. above), so the rule with the conclusion $\neg\ell$ becomes redundant. Of course, the original knowledge expressed by rules is partly lost.

Thanks to the flexibility of 4QL, one can combine conservative and drastic strategies. For example, in particular circumstances one can disallow removing

```
module r:
  relations: is(literal).
             do(literal).
  rules:
    is(risky) :- is(poison) | is(heat).
    -is(risky) :- -is(poison), -is(heat).
    is(heat) :- is(fire).
    -is(poison) :- is(safe).
    is(busy) :- do(detox) | do(extinguish).
    -do(detox) :- is(heat).
    -is(fire) :- is(lowtemp).
    is(X) :- perceived.is(X).
    -is(X) :- -perceived.is(X).
    do(extinguish) :- is(heat) | is(fire).
    -do(extinguish) :- -is(heat), -is(fire).
end.
module perceived:
  relations:   is(literal).
  facts:
    is(heat).
    is(fire).
end.
```

Figure 2: The 4QL program shown in Figure 1 after belief revision.

certain rules, for example those representing “hard” beliefs, and allow for removing other ones having a weaker status, for example because they have been machine learned and temporarily assumed to be valid.

To illustrate the idea consider again rules in Figure 1. Assume that we want to revise beliefs so that $is(heat)$ becomes t. For simplicity we use the drastic approach and obtain the module as illustrated in Figure 2 where, comparing to Figure 1, we removed:

`-is(heat) :- is(lowtemp).` (from module *r*)

`-is(heat).` (form module *perceived*)

and added:

`is(heat).` (to module *perceived*).

Note that the addition of $is(heat)$ is superfluous, since $is(heat)$ follows from $is(fire)$ using one of rules. The obtained 4QL program determines the following well-supported model for *perceived*:

$$\{is(heat),\ is(fire)\} \tag{5}$$

and the following well-supported model for r:

$$\{is(heat), is(fire), is(busy), is(risky), \neg do(detox), do(extinguish)\}. \tag{6}$$

```
module blocked:
  relations: item(literal).
  facts:
    item(heat).
    item(fire).
end.
```

Figure 3: A 4QL module for blocking conclusions as to heat and fire.

Remark 1. In the description of belief revision techniques we dealt with ground instances of rules. Of course, this is only a technical assumption. In actual implementations one can still have rules with variables. A simple way of blocking unwanted conclusions is to indicate items to be blocked and exclude them in premisses of rules. For example, if we have the rule:

`is(X) :- perceived.is(X).` (7)

and we want to exclude some of its conclusions, then we can replace (7) with the following rule:

```
is(X) :- perceived.is(X), -(blocked.item(X) IN {true}).
```

where `blocked` is a separate module containing blocked items.

Now blocking certain conclusions can be done by adding to module `blocked` facts about items to be blocked. An example module `blocked` is illustrated in Figure 3. $\triangleleft$

3.3. Shared Beliefs

When two agents first engage in communication they form a virtual group, for which we can define epistemic profiles as well. A broader treatment of beliefs shared by a group of agents can be found in [14], from where the following definition originates.

Definition 7. Let agents $Ag_1, \ldots, Ag_m$ be equipped with belief structures $\mathscr{B}^{\mathscr{E}_1} = \langle \mathscr{C}_1, F_1 \rangle, \ldots, \mathscr{B}^{\mathscr{E}_m} = \langle \mathscr{C}_m, F_m \rangle$, respectively. A *belief structure of the group of agents* $G = \{Ag_1, \ldots, Ag_m\}$ is a belief structure $\mathscr{B}^{\mathscr{E}}_G = \langle \mathscr{C}_G, F_G \rangle$, where constituents of G are $\mathscr{C}_G = \{F_1, \ldots, F_m\}$ and $\mathscr{E}_G$ is an epistemic profile transforming $\{F_1, \ldots, F_m\}$ into F_G. $\triangleleft$

Consequents of group members (F_{i_k}) become constituents of a group ($\mathscr{C}_G$). The group then builds group beliefs via its epistemic profile $\mathscr{E}_G$ and reaches its consequent F_G. For complexity reasons, it is reasonable to assume that only formed groups are equipped with belief structures and epistemic profiles. When a group is not formed, we assume that its belief structure $\mathscr{B}_G^{\mathscr{E}}$ is “empty”, i.e., $\mathscr{C}_G = \{\emptyset\}$ and $F_G = \emptyset$.

Information acquired via speech acts is then stored in the set of group level constituents, which are naturally transformed into consequents according to the epistemic profile of the group (see Figure 4). They represent the **beliefs shared** by the agents via communication.

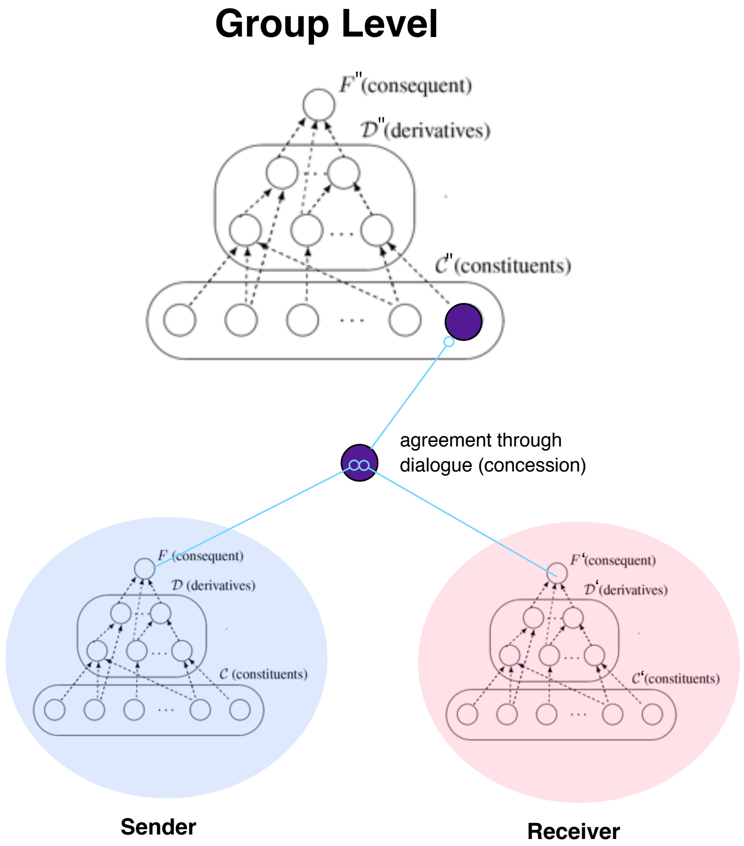

Figure 4: Belief formation at a (virtual) group level, incorporating interlocutors’ concessions as the group constituents. Updated after [17] with permission.

Whenever a concession is sent by the receiver, it means that an agreement with the sender about the valuation of the literal has been achieved. In the Figure 4, an agreement between the *Sender* and *Receiver* about a common valuation of the literal originating from the set of consequents (F in case of the *Sender* and F' in case of the *Receiver*) has been obtained. In such cases, we may consider those two agents form a virtual group, which establishes some common valuations of literals. The conceded literal is then automatically added to the set of consitituents on the group level (C'' in the Figure 4). Technically for every such group a separate 4QL program must be kept, which defines the way the group performs. Importantly, each agent composing the group should have a same copy of the group program. The belief revision on the group level is the same as on the individual level. Here, instead of starting from the individual agent's constituents, we start from their consequents and make changes (updates, block or remove rules) to the group epistemic profile and group consequents (see Section 5.1).

4. A Four-valued Characterization of Speech Acts

We define speech acts by specifying their *preconditions* and *complex post-actions*. By:

$$\{\textit{precondition}\}\langle \textit{speech act}\rangle[\textit{complex post-action}] \tag{8}$$

we mean that performing a *speech act* in the presence of the *precondition* triggers the *complex post-action*. We detail their semantics in the form of tables, in which, depending on the type of communicative relation and the preconditions, the shape of the speech act and the resulting complex post-action is determined. For simplicity, we number the respective communicative relation types, where: A signifies communication with an authority, P signifies peer to peer communication, and S communication with subordinate.

The precondition is in this case the receiver's valuation of the proposition in question. For clarity, throughout this section we employ the following notation, where $\mathscr{B}^{\mathscr{E}} = \langle \mathscr{C}, F\rangle$ is agent R's belief structure and $\mathscr{E}$ is its epistemic profile and primed expressions refer to its belief structure and epistemic profile after performing belief revision:

$$v_R(\alpha) \overset{\text{def}}{=} \alpha(F, v);\;\; v'_R(\alpha) \overset{\text{def}}{=} \alpha(F', v). \tag{9}$$

It is important to note that this way the considered formulas reflect beliefs. In fact, formula α in (9) is evaluated in the consequent F (respectively, F'). Therefore formula α is evaluated as $Bel(\alpha)$ (see Definition 6).

4.1. Assertions

$\texttt{assert}_{S,R}(\alpha, x)$ stands for agent S (sender) telling agent R (receiver) that its valuation of α is x. Let us focus on different configurations, taking into account both communicative relations between the agents and their valuations of α:

1. *Perceiving inconsistent information*, where agents engaged in peer-to-peer communication or communication with authority relations, adopt the interlocutors' theses about inconsistent information. Authorities, however, adopt such a thesis only if it was already inconsistent or unknown. In both cases a `concede` is sent as an acknowledgement, but only in the latter case, belief revision takes place.
2. *Perceiving previously unknown information*, which leads to adopting it by the receiver regardless the communicative relation. Unless the new information is also unknown (see below), belief revision takes place and a `concede` is sent.
3. *Perceiving information that is unknown*, which is ignored regardless the communicative relation, because by default all information is unknown. Therefore, taking action under such circumstances would be redundant.
4. *Perceiving previously inconsistent information*, which depends on the communicative relation. In communication with authority, the sender's belief (unless unknown) overrides the receiver's. A belief revision takes place if necessary, that is, unless their beliefs are equal, and a `concede` is sent. In the two remaining cases the message is ignored, unless it is also inconsistent. If so, a `concede` is sent.
5. *Perceiving compatible information*, where "compatible" means that both agents have exactly the same valuation of the proposition. This yields no belief revision, but in all cases but one, the assertion is acknowledged by sending a `concede`. In the one case of unknown information such a perception is ignored.
6. *Perceiving contradictory information*, where, regardless communicative relation, whenever one agent believes a proposition is true and the other believes the contrary, they must come to an agreement using a `challenge` speech act. This may succeed, leading to adopting the sender's thesis, or fail, with no direct effect on the interlocutors. Notice, that upon receiving $\texttt{assert}_{S,R}(\alpha,x)$, the speech act $\texttt{challenge}_{R,S}(\alpha,x)$ stands for agent R asking agent S: "why does your valuation of α equal x?". For more details on the semantics of challenges see Subsubection 4.4 at the end of this subsection.

In Table 4, all complex post-actions of assertions discussed in cases 1-6 are summarized.

Table 4: Perceiving information.

Type	Precondition	Speech Act	Complex Post-Action[5]
A, P	$v_R(\alpha)=\mathfrak{f}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$v'_R(\alpha)=\mathfrak{i};\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
S	$v_R(\alpha)=\mathfrak{f}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	-
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$v'_R(\alpha)=\mathfrak{i};\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
A, P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
A, P	$v_R(\alpha)=\mathfrak{t}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$v'_R(\alpha)=\mathfrak{i};\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
S	$v_R(\alpha)=\mathfrak{t}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	-
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{f})$	$v'_R(\alpha)=\mathfrak{f};\mathtt{concede}_{R,S}(\alpha,\mathfrak{f})$
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$v'_R(\alpha)=\mathfrak{i};\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{t})$	$v'_R(\alpha)=\mathfrak{t};\mathtt{concede}_{R,S}(\alpha,\mathfrak{t})$
A, P, S	$v_R(\alpha)=\mathfrak{f}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{t}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{f})$	$v'_R(\alpha)=\mathfrak{f};\mathtt{concede}_{R,S}(\alpha,\mathfrak{f})$
P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{f})$	—
A, P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
A	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{t})$	$v'_R(\alpha)=\mathfrak{t};\mathtt{concede}_{R,S}(\alpha,\mathfrak{t})$
P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{t})$	—
A, P, S	$v_R(\alpha)=\mathfrak{f}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{f})$	$\mathtt{concede}_{R,S}(\alpha,\mathfrak{f})$
A, P, S	$v_R(\alpha)=\mathfrak{u}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{u})$	-
A, P, S	$v_R(\alpha)=\mathfrak{i}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{i})$	$\mathtt{concede}_{R,S}(\alpha,\mathfrak{i})$
A, P, S	$v_R(\alpha)=\mathfrak{t}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{t})$	$\mathtt{concede}_{R,S}(\alpha,\mathfrak{t})$
A, P, S	$v_R(\alpha)=\mathfrak{t}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{f})$	$\mathtt{challenge}_{R,S}(\alpha,\mathfrak{f})$
A, P, S	$v_R(\alpha)=\mathfrak{f}$	$\mathtt{assert}_{S,R}(\alpha,\mathfrak{t})$	$\mathtt{challenge}_{R,S}(\alpha,\mathfrak{t})$

4.2. Requests

$\mathtt{request}_{S,R}(\alpha)$ stands for agent *S* requesting agent *R* to provide information about α. After such a request, the sender must wait for a reply, while the receiver should reply with what it knows about α:

$$\{v_R(\alpha)=x\}\ \langle\mathtt{request}_{S,R}(\alpha)\rangle\ [\mathtt{assert}_{R,S}(\alpha,x)].$$

The sender, after receiving the response, behaves according to the rules for assertions.

Table 5: Requests, concessions and challenges. Cases not included in the table are ignored.

Type	Precondition	Speech Act	Complex Post-Action
A, P, S	$v_R(\alpha) = \mathfrak{f}$	$\texttt{request}_{S,R}(\alpha)$	$\texttt{assert}_{R,S}(\alpha, \mathfrak{f})$
A, P, S	$v_R(\alpha) = \mathfrak{u}$	$\texttt{request}_{S,R}(\alpha)$	$\texttt{assert}_{R,S}(\alpha, \mathfrak{u})$
A, P, S	$v_R(\alpha) = \mathfrak{i}$	$\texttt{request}_{S,R}(\alpha)$	$\texttt{assert}_{R,S}(\alpha, \mathfrak{i})$
A, P, S	$v_R(\alpha) = \mathfrak{t}$	$\texttt{request}_{S,R}(\alpha)$	$\texttt{assert}_{R,S}(\alpha, \mathfrak{t})$
A, P, S	$v_R(\alpha) = \mathfrak{f}$	$\texttt{concede}_{S,R}(\alpha, \mathfrak{f})$	add α to group[a] constituents[b]
A, P, S	$v_R(\alpha) = \mathfrak{u}$	$\texttt{concede}_{S,R}(\alpha, \mathfrak{u})$	add α to group constituents
A, P, S	$v_R(\alpha) = \mathfrak{i}$	$\texttt{concede}_{S,R}(\alpha, \mathfrak{i})$	add α to group constituents
A, P, S	$v_R(\alpha) = \mathfrak{t}$	$\texttt{concede}_{S,R}(\alpha, \mathfrak{t})$	add α to group constituents
A, P, S	$v_R(\alpha) = \mathfrak{f}$	$\texttt{challenge}_{S,R}(\alpha, \mathfrak{t})$	$\texttt{assert}_{R,S}(PROOF(\alpha, \mathfrak{t}))$
A, P, S	$v_R(\alpha) = \mathfrak{t}$	$\texttt{challenge}_{S,R}(\alpha, \mathfrak{f})$	$\texttt{assert}_{R,S}(PROOF(\alpha, \mathfrak{f}))$

---

[a] The virtual group consisting of R and S.

[b] Adding facts to the set of constituents may trigger belief revision on the virtual group level.

4.3. Concessions

$\texttt{concede}_{S,R}(\alpha, x)$ stands for agent S's communicating its agreement about the valuation of α. Importantly, only concessions about compatible valuations are considered, others are ignored, as indicated in Table 5. Concession is more of an acknowledgement, as no belief revision on the individual level takes place here. Instead, α with a valuation x is added to the agents' virtual group's set of constituents (see Subsection 3.1) and (possibly) a belief revision on the virtual group's level occurs.

4.4. Challenges

$\texttt{challenge}_{S,R}(\alpha, x)$ stands for S's communicating its contradictory stance with respect to R's opinions regarding α (x is either $\mathfrak{t}$ or $\mathfrak{f}$ here). Inspired by [35], a challenge is in fact a request to provide a proof of the receiver's stance towards α together with an implicit assertion of the contradictory stance towards α. To restrict emitting redundant information we make this assertion implicit.

$$\texttt{challenge}_{S,R}(\alpha, \mathfrak{t}) \equiv \texttt{request}_{S,R}(\texttt{assert}_{R,S}(PROOF(\alpha, \mathfrak{t})))$$
$$\texttt{challenge}_{S,R}(\alpha, \mathfrak{f}) \equiv \texttt{request}_{S,R}(\texttt{assert}_{R,S}(PROOF(\alpha, \mathfrak{f})))$$

The proof in question depends on the structure of α and might represent just the last rule used to derive α (or a choice of rules if there were several ways to achieve α). If, for an atomic α, its negation $\neg\alpha$ is a fact, there is no way to prove that α is true, and the challenged agent who received $\texttt{challenge}_{S,R}(\alpha,\mathbf{t})$ should reply with a special symbol $\texttt{assert}_{R,S}(\alpha,\bot)$ for "I give up" (cf. [35]). This counts as agent's R failure to prove α. An agent who receives a `challenge` should react according to the rules for requests. Challenges per se do not yield belief revision, the true impact on agents' beliefs is achieved by the assertions initiated by them. However, when a (possibly deeply nested) challenge folds, if it was successful, an acknowledgement must be sent. Therefore whenever an agent sends $\texttt{challenge}_{S,R}(\alpha,x)$, if at some point α becomes x it triggers sending a `concede` by S. This is a sign of a challenge ending successfully for R. Otherwise a challenge fails. Notice that even a failed challenge might have caused some belief revision if some of the assertions have been acknowledged.

5. An Example

To demonstrate perceiving speech acts in the four-valued framework, consider a rescue-agent (receiver) capable of putting out fire and detoxicating an area. Suppose at the moment it is aware of the fire but due to faulty sensors cannot sense the heat. The situation is reflected by the 4QL module illustrated in Figure 1 (where *lowtemp* is an output of the sensor in the vicinity of the suspect fire). Let us recall that for simplicity, we focus on the case when the sender S is an authority and the receiver R a subordinate. Let B_{R_1} be the well-supported model (4) determined by this program.

On the basis of B_{R_1} we know that agent R is inconsistent about $is(heat)$ and $do(detox)$. Literals $is(fire)$, $is(risk)$, $is(busy)$ and $do(extinguish)$ are true. Literals absent in the model are unknown.

The dialogue starts when agent S, an authority, asserts to R that *heat* is true (see (a) in Figure 5). In R's belief base B_{R_1}, $is(heat)$ is inconsistent. Therefore, after perceiving this assertion, according to the belief revision rules (see Table 4), agent R adopts S's belief. In the course of belief revision, the program shown in Figure 1 is changed into the one shown in Figure 2 so the new well-supported model B_{R_2} is the one given by (6).

Finally, agent R answers with a concession (see (b) in Figure 5), acknowledging that it shares a belief about $is(heat)$ being true. Notice, that if R's valuation of $is(heat)$ would have been false in the first place (in B_{R_1}), it would have to challenge the sender's assertion. Then, S would have to provide a proof.

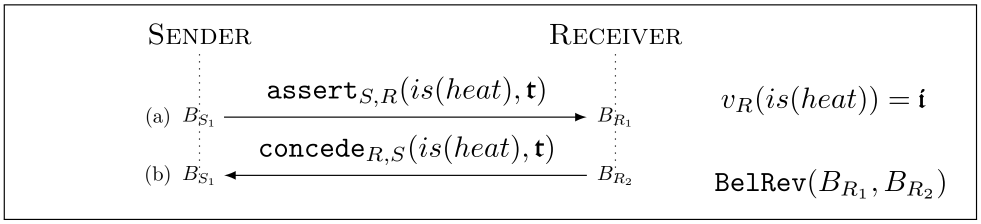

Figure 5: Perceiving compatible information.

Let us continue with the example from above but now consider a case of conflict resolution by means of a challenge. Now the sender makes an assertion that $do(extinguish)$ is false (see (a) in Figure 6), which is contradictory with the receiver's beliefs. In our scenario, regardless the communicative relation, *R* must challenge the sender, as it should abandon $do(extinguish)$ only if entirely convinced. Therefore, *R* sends a challenge (see (b) in Figure 6), to which *S* must answer with a proof. As indicated in the description of the `challenge` speech act, we focus on the request part. The additional assertion is omitted to keep the picture clear. Assume that the relevant rules that the sender agent will be using to prove it's statements are the following:

```
-do(extinguish) :- -is(fire).
-is(fire) :- is(lowtemp), -is(heat).
```

(10)

As explained in Subsection 4 in the part about challenges, agent *S* must provide a proof for the assertion about $\neg do(extinguish)$ it has made before. In our example, the last rule in the derivation of $\neg do(extinguish)$ determines the proof (see (10)). In order for $\neg do(extinguish)$ to hold, $is(fire)$ must be false. Therefore, *S* answers with an assertion (see (c) in Figure 6) about $is(fire)$. Now we analyze what impact this assertion has on agent *R*. The assertion about $is(fire)$ being false, made by agent *S*, causes another challenge (see (d) in Figure 6), because $is(fire)$ was true in B_{R_1}. Luckily, again *S* knows how to prove this (see (10)) and answers with a sequence of assertions (see (e) and (g) in Figure 6): one about $is(lowtemp)$ and one about $\neg is(heat)$.

The first assertion causes *R* to revise its beliefs (because $is(lowtemp)$ is absent (unknown) from the model) and acknowledge (see (f) in Figure 6). There are no rules with conclusion $\neg is(lowtemp)$, so belief revision amount here to adding the

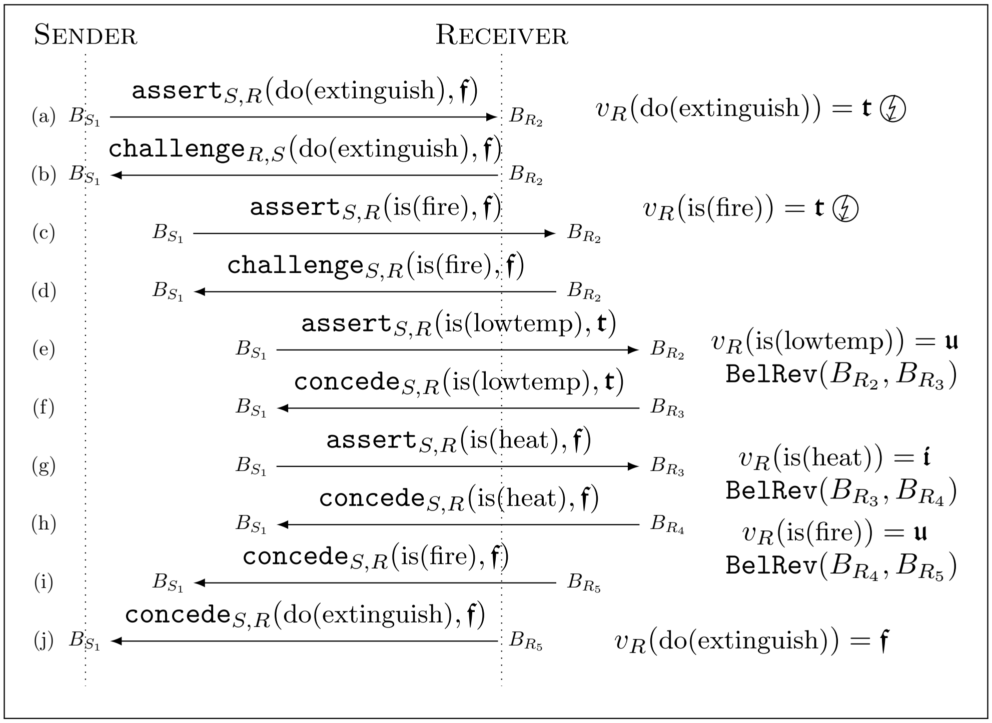

Figure 6: Perceiving contradictory information.

new fact $\neg is(lowtemp)$ to the *perceived* module and computing the new well-supported model. This results in its subsequent belief base B_{R_3}:

$$\begin{aligned}&\{is(heat), \neg is(heat), is(fire), \neg is(fire), is(busy), \neg is(busy), is(risky),\\ &is(lowtemp), do(extinguish), \neg do(extinguish), do(detox), \neg do(detox)\}.\end{aligned} \tag{11}$$

The second assertion also causes belief revision (S is an authority and $is(heat)$ is inconsistent in B_{R_3}). This time, according to the belief revision rules in Table 3, the receiver should remove:

`is(heat) :- is(fire).` (from module *r*)

`is(heat).` (form module *perceived*)

and add:

`-is(heat).` (to module *perceived*).

The resulting new belief base B_{R_4} is:

$$\begin{gathered}\{\neg is(heat), is(fire), \neg is(fire), is(lowtemp),\\ do(extinguish), \neg do(extinguish)\}.\end{gathered} \tag{12}$$

In the changed belief base, the receiver agent can now reconsider $\neg is(fire)$. As $is(fire)$ is inconsistent in B_{R_4}, agent R performs belief revision anew and obtains belief base B_{R_5}, which is the final set of R's consequents:

$$\{is(lowtemp), \neg is(heat), \neg is(fire), \neg do(extinguish)\}. \tag{13}$$

```
module r:
  relations: is(literal).
             do(literal).
  rules:
    is(risky) :- is(poison) | is(heat).
    -is(risky) :- -is(poison), -is(heat).
    -is(poison) :- is(safe).
    is(busy) :- do(detox) | do(extinguish).
    -do(detox) :- is(heat).
    -is(fire) :- is(lowtemp).
    is(X) :- perceived.is(X).
    -is(X) :- -perceived.is(X).
    do(extinguish) :- is(heat) | is(fire).
    -do(extinguish) :- -is(heat), -is(fire).
end.
module perceived:
  relations:   is(literal).
  facts:
    -is(heat).
    -is(fire).
    is(lowtemp).
end.
```

Figure 7: The 4QL program shown in Figure 2 after communication.

Agent R concedes that $is(fire)$ is false (see (i) in Figure 6), at which point the second challenge folds. Since agent R has been convinced that $do(extinguish)$ is false as well (13), it replies with another concession (see (j) in Figure 6). That causes the first challenge to fold and we may conclude that agent S has successfully convinced R that $do(extinguish)$ no longer holds. The final module representing agent's R epistemic profile and constituents after this communication is presented in Figure 7.

Notice, that although the above scenario partially resembles a persuasion dialogue, it should not be considered as an example of such. As mentioned before, in this work we provide a basis for building more complex dialogues by analyzing what happens when agents perceive speech acts, the reactional layer of communication. In true persuasion dialogues, various interesting aspects come into play, among others the motivation of agents and their pro-activeness when choosing to utter a speech act, which clearly is outside of the scope of this paper (but see [35] and [18, Chapter 8]).

5.1. Building a Shared Belief Base

In the course of the dialogue above, the agents exchange their beliefs and acknowledge the common ones. Agents which engage in communication for the first time, form a virtual group, which by definition is empty: the set of group constituents contains only an empty set and the are no group consequents. In addition, the epistemic profile in such cases is an identity function. This situation and can be represented in 4QL trivially, by one module with no rules, only facts obtained by communication. In this case, this singular module represents both constituents and consequents.

Whenever a concession is sent, it means an agreement has been made, therefore the conceded literals should be added to the agents' shared knowledge base, at the level of constituents.

Adding new facts may, similarly as in the case of individual agents, lead to belief revision. In our example this happened, e.g., after the first assertion (see (a) in Figure 5), agent R conceded that $is(heat)$ was true. Then, in the course of the challenge, S managed to convince him that *is(heat)* was not the case anymore (see (h) in Figure 6). Therefore, at first $is(heat)$ was in the shared belief base but ultimately $\neg is(heat)$ took its place.

Because the group's epistemic profile is empty (this virtual group has just emerged through communication), the set of constituents is transformed into the set of consequents by an identity transformation (in our forthcoming paper, adding also rules to the group's epistemic profile will become possible).

In summary, after the process of exchanging information, the belief base of the virtual group consisting of S and R includes the following facts:

$$\{\, is(lowtemp), \neg is(fire), \neg is(heat), \neg do(extinguish)\}. \tag{14}$$

6. Conclusions and Discussion

Agents need to act in uncertain and dynamic environments, where they receive information from multiple sources. A new four-valued paraconsistent logic [23, 24] appears to be cut out for this situation. Nonmonotonic logic, in turn, allows for drawing conclusions that typically hold, but not necessarily always. Their combination, implemented in 4QL, has already been shown to model agents' individual and group beliefs. Agents' reasoning schemas are formalized in terms of rules in the chosen four-valued knowledge-based framework, belief structures and epistemic profiles [14]. A great bonus of 4QL is that queries can be computed in polynomial time. This tractability stands in stark contrast to the usual two-valued approaches to group interactions, where EXPTIME completeness of satisfiability problems is a common hindrance [19].

Notice, that such features distinguish 4QL from other formalisms, e.g., Answer Set Programming (ASP)[20], where computing a so-called "answer set" (stable model) is NP-complete. Further, ASP is based on the trivalent semantics (*true*, *false*, *unknown*), and does not admit inconsistency. The answer sets may contain conclusions that are not grounded in facts, which may be suitable for ASP primary applications (specification and computation of problems from the NP class), however is not appropriate in our case.

Dealing with paraconsistency in dialogue is not a new subject (see Chapter 6 in [33] for general nonmonotonic/defeasible reasoning techniques and e.g., [5, 6, 9] for paraconsistent reasoning techniques). For example, in classical logic, one of the mechanisms preventing the logic from trivialization is employing consistent subsets of the knowledge [22]. However, instead of restricting oneself to such methods, a full-fledged paraconsistent logic can be exploited. In [32], the authors proposed a logic of multiple-valued argumentation (LMA), in which agents can argue using multi-valued knowledge base in the extended annotated logic programming (EALP). Such an approach was next applied in [26]. Although this work shows the most congenial approach, our solution uses the 4QL logic while EALP provides a framework for arbitrary many-valued logics. When tailored to four-valued paraconsistent logics, ELAP ususally uses a Belnap-like approach. As discussed in Example 1, in our application areas the 4QL logic delivers more intuitive results.

This paper is a first step in a research program that combines dialogue theory and argumentation theory with the new four-valued approach to modeling multi-agent interactions. We consider three types of communication: from an authority, from a peer, and from a subordinate. In each type of communication, the speech acts are considered from the mentalistic perspective, as expressed in the triple: precondition, speech act, complex post-action. Along with defining rules for perceiving speech acts, we indicated how communication influences agents' beliefs.

Our example of a simple dialogue between a rescue-agent and its boss shows how speech acts and agents' reasoning rules naturally combine in the framework of 4QL, leading to intuitive conclusions while maintaining tractability. Thus, a foundation has been laid for extending the four-valued approach to modeling more complex dialogues and argumentations between agents reasoning in uncertain and dynamic environments. This will be the subject of the forthcoming paper.

7. Acknowledgements

This work has been supported by the Polish National Science Centre grant 2011/01/B/ST6/02769. The authors would like to thank anonymous reviewers for their comments which greatly improved this paper.

References

[1] K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152:157–206, 2005.

[2] J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, second edition, 1975. Edited by J. O. Urmson and M. Sbisa.

[3] D. Barbucha, M.T. Le, R.J. Howlett, and L.C. Jain, editors. *Advanced Methods and Technologies for Agent and Multi-Agent Systems, Proceedings of the 7th KES Conference on Agent and Multi-Agent Systems - Technologies and Applications (KES-AMSTA 2013)*, volume 252 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2013.

[4] N.D. Belnap. A useful four-valued logic. In G. Epstein and J.M. Dunn, editors, *Modern Uses of Many Valued Logic*, pages 8–37. Reidel, 1977.

[5] J.-Y. Beziau, W. Carnielli, and D.M. Gabbay. *Handbook of Paraconsistency*. College Publications, 2007.

[6] H. A. Blair and V. S. Subrahmanian. Paraconsistent logic programming. *Theor. Comput. Sci.*, 68(2):135–154, October 1989.

[7] K. Budzynska, M. Kacprzak, and P. Rembelski. Perseus. software for analyzing persuasion process. *Fundamenta Informaticae*, 93(1-3):65–79, 2009.

[8] C. Castelfranchi. The social nature of information and the role of trust. *International Journal of Cooperative Information Systems*, 11(3):381–403, 2002.

[9] Carlos Viegas Damsio and Lus Moniz Pereira. A survey of paraconsistent semantics for logic programs. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 241–320. Kluwer Academic Publishers, 1998.

[10] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9:145–158, 2001.

[11] D. Dubois. On ignorance and contradiction considered as truth-values. *Logic Journal of the IGPL*, 16(2):195–216, 2008.

[12] B. Dunin-Kęplicz, A. Strachocka, A. Szałas, and R. Verbrugge. Perceiving speech acts under incomplete and inconsistent information. In Barbucha et al. [3], pages 255–264.

[13] B. Dunin-Kęplicz, A. Strachocka, and R. Verbrugge. Deliberation dialogues during multi-agent planning. In M. Kryszkiewicz, H. Rybinski, A. Skowron, and Z. W. Ras, editors, *ISMIS*, volume 6804 of *LNCS*, pages 170–181. Springer, 2011.

[14] B. Dunin-Kęplicz and A. Szałas. Epistemic profiles and belief structures. In G. Jezic, M. Kusek, N.T. Nguyen, R.J. Howlett, and L.C. Jain, editors, *Proc. 6th KES AMSTA International Conference on Agent and Multi-Agent Systems. Technologies and Applications*, volume 7327 of *LNCS*, pages 360–369. Springer-Verlag, 2012.

[15] B. Dunin-Kęplicz and A. Szałas. Taming complex beliefs. *Transactions on Computational Collective Intelligence XI*, LNCS 8065:1–21, 2013.

[16] B. Dunin-Kęplicz and A. Szałas. Indeterministic belief structures. In *Proc. KES-AMSTA 2014: Agents and Multi-agent Systems: Technologies and Applications*, volume 296 of *Advances in Intelligent and Soft Computing*, pages 57–66. Springer, 2014.

[17] B. Dunin-Kęplicz, A. Szałas, and R. Verbrugge. Tractable reasoning about group beliefs. In *2nd international Workshop on Engineering Multi-Agent Systems (EMAS 2014)*, Lecture Notes in Artificial Intelligence. Springer Berlin Heidelberg, 2014.

[18] B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach.* Wiley, Chichester, 2010.

[19] M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz. Complexity issues in multiagent logics. *Fundamenta Informaticae*, 75(1-4):239–262, 2007.

[20] M. Gelfond and Y. Kahl. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents - The Answer-Set Programming Approach.* Cambridge University Press, 2014.

[21] H. Lebbink, C. Witteman, and J.-J. Meyer. Dialogue games for inconsistent and biased information. *Electronic Notes in Theoretical Computer Science*, 85(2):134–151, 2004.

[22] R. Malouf. Maximal consistent subsets. *Comput. Linguist.*, 33(2):153–160, 2007.

[23] J. Małuszyński and A. Szałas. Living with inconsistency and taming nonmonotonicity. In O. de Moor, G. Gottlob, T. Furche, and A. Sellers, editors, *Datalog Reloaded*, volume 6702 of *LNCS*, pages 384–398. Springer-Verlag, 2011.

[24] J. Małuszyński and A. Szałas. Partiality and inconsistency in agents' belief bases. In Barbucha et al. [3], pages 3–17.

[25] V. Mascardi, D. Demergasso, and D. Ancona. Languages for programming BDI-style agents: an overview. In F. Corradini, F. De Paoli, E. Merelli, and A. Omicini, editors, *WOA 2005 - Workshop From Objects to Agents*, pages 9–15, 2005.

[26] K. Matsunaga and H. Sawamura. Aaslma: An automated argument system based on logic of multiple-valued argumentation. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3684 of *Lecture Notes in Computer Science*, pages 830–838. Springer, 2005.

[27] Simon Parsons and Peter McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12:415–439, 2003.

[28] H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.

[29] C. Reed and D. Walton. Towards a formal and implemented model of argumentation schemes in agent communication. *Autonomous Agents and Multi-Agent Systems*, 11(2):173–188, 2005.

[30] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, 1969.

[31] A. Szałas. How an agent might think. *Logic J. IGPL*, 21(3):515–535, 2013.

[32] T. Takahashi and H. Sawamura. A logic of multiple-valued argumentation. In *AAMAS*, pages 800–807. IEEE Computer Society, 2004.

[33] F. van Harmelen, V. Lifschitz, and B. Porter. *Handbook of Knowledge Representation*. Elsevier Science, San Diego, USA, 2007.

[34] A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.

[35] D.N. Walton and E.C.W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.

Perceiving Rules under Incomplete and Inconsistent Information[⋆]

Barbara Dunin-Kęplicz, Alina Strachocka

Institute of Informatics, University of Warsaw, Poland

Abstract. The overall goal of this research program is a construction of a paraconsistent model of agents' communication, comprising two building blocks: speaking about facts and speaking about reasoning rules. To construct complex dialogues, such as persuasion, deliberation, information seeking, negotiation or inquiry, the speech acts theory provides the necessary building material. This paper extends the implementation of the speech act *assert* in the paraconsistent framework, presented in our previous paper, by providing means for agents to perceive and learn not only facts, but also rules. To this end the *admissibility criterion* for a rule to be accepted has been defined and the Algorithm for Perceiving Assertions About Rules has been proposed. A natural four-valued model of interaction yields multiple new cognitive situations. *Epistemic profiles* encode the way agents reason, and therefore also deal with inconsistent or lacking information. *Communicative relations* in turn comprise various aspects of communication and allow for the fine-tuning of applications.

The particular choice of a rule-based, DATALOG$^{\neg\neg}$-like query language 4QL as a four-valued implementation framework ensures that, in contrast to the standard two-valued approaches, tractability of the model is maintained.

1 Communication under Uncertain and Inconsistent Information

The traditional approaches to modeling Agent Communication Languages settled for the two-valued logics despite their natural modeling limitations: inability to properly deal with lacking and inconsistent information. This work continues the subject-matter of the paraconsistent approach to formalizing dialogues in multiagent systems in a more realistic way [5]. The underpinning principle of this research is the adequate logical modeling of the dynamic environments in which artifacts like agents are situated. Agents, viewed as heterogenous and autonomous information sources, may perceive the surrounding reality differently while building their informational stance. Even though consistency of their belief structures is a very desirable property, in practice it is hard to achieve: inevitably, all these differences result in the lack of consistency of their beliefs. However, instead of making a reasoning process trivial, we view inconsistency as a first-class citizen trying to efficiently deal with it.

There is a vast literature on logical systems designed to cope with inconsistency (see for example [28, 33]). However none of them turned out to be suitable in all cases. As

---

[⋆] Supported by the Polish National Science Centre grants 2011/01/B/ST6/02769 and CORE 6505/B/T02/2011/40

inconsistency is an immanent property of realistic domains, we lean towards a more pragmatic and flexible solution. Assuming that we have various disambiguation methods at hand, the flexible approach allows for an application-, situation- or context-specific choice that does not have to be made a priori. Furthermore, there might be a benefit from postponing the related decision as long as possible, as the new information may come up or the agent being the cause of the conflict may change its mind.

We base our solution on a four-valued logic [12] and the ideas underlying the 4QL query language [11, 12] which major win is that queries can be computed in polynomial time. Tractability of 4QL stands in stark contrast to the usual two-valued approaches to group interactions, where EXPTIME completeness of satisfiability problems is a common hindrance [7, 8]. This way an important shift in perspective takes place: rather than drawing conclusions from logical theories we reason from paraconsistent knowledge bases. As a great benefit, the belief revision methods turned out to be dramatically simplified. 4QL was designed in such a way that the inconsistency is tamed and treated naturally in the language. The application developer has a selection of uniform tools to adequately deal with inconsistencies in their problem domain.

Building upon the 4-valued logic of 4QL, we deal with four types of situations:

- fact a holds,
- fact a does not hold,
- it is not known whether a holds,
- information about a is inconsistent.

They are confined in the four logical values: $\mathfrak{t}$, $\mathfrak{f}$, $\mathfrak{u}$ and $\mathfrak{i}$, respectively (Sec. 3). In such settings, maintaining truth or falsity of a proposition, in the presence of multiple information sources, is challenging. Furthermore, the two additional logical values allow to model complex interactions between agents in a more intuitive and subtle manner.

The way the individual agents deal with conflicting or lacking information is encoded in their *epistemic profile* (Sec. 4) which embodies their reasoning capabilities, embracing the diversity of agents and providing a complete agent's characteristics. Moreover, epistemic profiles specify agents' communicative strategies realized with regard to the three communicative relations between the agents involved: *communication with authority*, *peer to peer communication* and *communication with subordinate* as proposed in [5]. These in turn influence the agent's reasoning processes and finally affect the agents' *belief structures*, i.e., their informational stance [6] (Sec. 4). In principle, various agents may reason in completely different ways, as well as apply diverse methods of information disambiguation.

The ultimate aim of our research program is a paraconsistent model of agents' communication. To construct complex dialogues, such as persuasion, deliberation, information seeking, negotiation or inquiry (see [18]), the speech acts theory provides the necessary building material. We initiated our research program [5], by proposing a paraconsistent framework for perceiving new facts via four different speech acts: *assert*, *concede*, *request* and *challenge*. They enable the agents to discuss their informational stance, i.e.,:

- inform one another about their valuations of different propositions via $assertions$,
- ask for other agents' valuations via $requests$,
- acknowledge the common valuations via $concessions$ and
- question the contradictory valuations via $challenges$.

In the current paper the next step is taken. We allow the agents to perceive not only new facts but also reasoning rules, which make up the epistemic profiles. To our best knowledge, approaches to modeling communication in MAS, as a legacy of Austin and Searle, settled for frameworks where propositions were the only valid content of speech acts. On the other hand, argumentation about reasoning rules has been well studied in the legal reasoning domain (see for example [26, 27, 34]). Here we intend to bring together these two worlds by leveraging the legal argumentation theory in our paraconsistent communication framework and therefore by allowing the agents to discuss their reasoning rules. We attack this complex problem from analyzing how agents react to perceiving assertions about reasoning rules: should they adopt, reject, ignore or maybe challenge the new rule? Consequently, the paramount issue here is the formulation of the admissibility criterion of the incoming rule (Sec. 5) as a basis to formulate the Algorithm for Perceiving Assertions about Rules.

As we view complex dialogues as communicative games between two or more agents, the dialogue participants, being independent information sources, try to expand, contract, update, and revise their beliefs through communication [25]. The great advantage of our approach is the possibility to revise the belief structures in a very straightforward way, what will be presented in the sequel.

The paper is structured as follows. First, in Section 2, we introduce the building blocks of our approach. Section 3 is devoted to a four-valued logic which is used throughout the paper and to basic information on 4QL. Section 4 introduces epistemic profiles and belief structures, whereas Section 5 outlines the communicative relations and rule admissibility conditions. Section 6 discusses the main technical contribution of the paper, followed by an example in Section 7. Finally, Section 8 concludes the paper.

2 Perceiving Rules

Our goal is to allow agents to communicate flexibly in the paraconsistent world. We will equip agents with various dialogical tools for conversing about rules: from informing or requesting information about a rule head or body, through challenging legitimacy of a rule, to rejecting or conceding acceptance of a new rule. These all can be performed with the use of dedicated speech acts: *assert*, *request*, *challenge*, *reject* and *concede* respectively and later will be used to construct complex dialogues.

In this paper, we take the first basic step, namely, how should agents react upon perceiving assertions $(\mathtt{assert}_{S,R})$ regarding rules $(l :\!- b)$ of inference. As these are ”actions that make you change your mind” [25], we explain the process of adopting the new rules and specifically put a spotlight on the easiness of the belief revision phase in our approach. Therefore we ask:

- **In what cases can the rules be added to the agent's epistemic profile without harming the existing structures?**
- **How does the agent's belief structure change in response?**

The merit of the rule base update in traditional approaches lies in solving inconsistency that the new rule might introduce to the logical program. When creating 4QL, the biggest effort was to ease the way we deal with inconsistency. We will exploit this when defining the *admissibility criterion* for a rule to be accepted. Informally, it is meant to express compatibility of the rule conclusions with the current belief structure. This compatibility is founded on the special ordering of truth values, by which we try to achieve two goals:

- protect true and false propositions from being flooded by inconsistency and
- protect already possessed knowledge from unknown.

The execution of the admissibility criterion is the heart of the Algorithm for Perceiving Assertions About Rules, a generalized 4-step procedure, realized via: *filtering*, *parsing*, *evaluation* and *belief revision*. In a perfect case, agents communicate successfully, extending and enriching their knowledge. In more realistic scenarios, some communicative actions fail, calling for a system consistency ensuring mechanism. Also, at each stage of the algorithm, agents must know how to proceed in the lack of response.

3 A Paraconsistent Implementation Environment

In order to deal with perceiving rules, we need to introduce several definitions (in Sections 3, 4 and 5):

- the 4-valued logic we build upon,
- the implementation tool: a rule-based query language 4QL,
- the notions of *epistemic profiles* and *belief structures*, which embody the agents' informational stands and reasoning capabilities,
- the preserving knowledge truth ordering,
- the rule admissibility criterion.

In what follows all sets are finite except for sets of formulas. We deal with the classical first-order language over a given vocabulary without function symbols. We assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols. A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with $\bar{\tau} \in (Const \cup Var)^k$, where k is the arity of R. *Ground literals over Const*, denoted by $\mathcal{G}(Const)$, are literals without variables, with all constants in *Const*. If $\ell = \neg R(\bar{\tau})$ then $\neg\ell \overset{\text{def}}{=} R(\bar{\tau})$. Let $v : Var \longrightarrow Const$ be a *valuation of variables*.

For a literal ℓ, by $\ell(v)$ we mean the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$. The semantics of propositional connectives is summarized in Table 1.

Table 1. Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$ (see [11, 12, 17]).

$\wedge$	f	u	i	t
f	f	f	f	f
u	f	u	u	u
i	f	u	i	i
t	f	u	i	t

$\vee$	f	u	i	t
f	f	u	i	t
u	u	u	i	t
i	i	i	i	t
t	t	t	t	t

$\rightarrow$	f	u	i	t
f	t	t	t	t
u	t	t	t	t
i	f	f	t	f
t	f	f	t	t

	$\neg$
f	t
u	u
i	i
t	f

Definition 3.1. The *truth value* of a literal ℓ w.r.t. a set of ground literals L and valuation v, denoted by $\ell(L, v)$, is defined as follows:

$$\ell(L,v) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$

◁

For a formula $\alpha(x)$ with a free variable x and $c \in$ *Const*, by $\alpha(x)^x_c$ we understand the formula obtained from α by substituting all free occurrences of x by c. Definition 3.1 is extended to all formulas in Table 2, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 1. Observe that the definitions of $\wedge$ and $\vee$ reflect minimum and maximum w.r.t. the ordering

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}. \tag{1}$$

Table 2. Semantics of first-order formulas.

- if α is a literal then $\alpha(L, v)$ is defined in Definition 3.1;
- $(\neg\alpha)(L, v) \stackrel{\text{def}}{=} \neg(\alpha(L, v))$;
- $(\alpha \circ \beta)(L, v) \stackrel{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x\alpha(x))(L, v) = \min_{a \in \textit{Const}} (\alpha^x_a)(L, v)$,
  where min is the minimum w.r.t. ordering (1);
- $(\exists x\alpha(x))(L, v) = \max_{a \in \textit{Const}} (\alpha^x_a)(L, v)$,
  where max is the maximum w.r.t. ordering (1).

From several languages designed for programming BDI agents (for a survey see, e.g., [13]), none directly addresses belief formation, in particular nonmonotonic or defeasible reasoning techniques. 4QL enjoys tractable query computation and captures all tractable queries. It supports a modular and layered architecture, providing simple, yet powerful constructs for expressing nonmonotonic rules reflecting default reasoning, autoepistemic reasoning, defeasible reasoning, the local closed world assumption, etc. [11]. The openness of the world is assumed, which may lead to lack of knowledge. Negation in rule heads may lead to inconsistencies.

Definition 3.2. By a *rule* we mean any expression of the form:

$$\ell :\!- \, b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}. \tag{2}$$

where $\ell, b_{11}, \ldots, b_{1i_1}, \ldots, b_{m1}, \ldots, b_{mi_m}$ are (negative or positive) literals and ‘,’ and ‘|’ abbreviate conjunction and disjunction, respectively. Literal ℓ is called the *head* of the rule and the expression at the righthand side of :– in (2) is called the *body* of the rule. By a *fact* we mean a rule with an empty body. Facts ‘ℓ :– .’ are abbreviated to ‘ℓ.’. A finite set of rules is called a *program*. ◁

Definition 3.3. Let a set of constants, $Const$, be given. A set of ground literals L with constants in $Const$ is a *model of a set of rules S* iff for each rule (2) and any valuation v mapping variables into constants in $Const$, we have that:

$$\big(((b_{11} \wedge \ldots \wedge b_{1i_1}) \vee \ldots \vee (b_{m1} \wedge \ldots \wedge b_{mi_m})) \rightarrow \ell\big)(L, v) = \mathfrak{t},$$

where it is assumed that the empty body takes the value $\mathfrak{t}$ in any interpretation. $\triangleleft$

To express nonmonotonic/defeasible rules we need the concept of modules and external literals. In the sequel, *Mod* denotes the set of *module names*.

Definition 3.4. An *external literal* is an expression of one of the forms:

$$M.R, -M.R, M.R \text{ IN } T, -M.R \text{ IN } T, \tag{3}$$

where $M \in$*Mod* is a module name, R is a positive literal, '$-$' stands for negation and $T \subseteq \{\mathfrak{f}, \mathfrak{u}, \mathfrak{i}, \mathfrak{t}\}$. For literals of the form (3), module M is called the *reference module*. $\triangleleft$

The intended meaning of "$M.R$ IN T" is that the truth value of $M.R$ is in the set T. External literals allow one to access values of literals in other modules. If R is not defined in the module M then the value of $M.R$ is assumed to be $\mathfrak{u}$.

Assume a strict tree-like order $\prec$ on *Mod* dividing modules into layers. An external literal with reference module M_1 may appear in rule bodies of a module M_2, provided that $M_1 \prec M_2$.

The semantics of 4QL is defined by well-supported models generalizing the idea of [9]. Intuitively, a model is *well-supported* if all derived literals are supported by a reasoning that is grounded in facts. It appears that for any set of rules there is a unique well-supported model and this can be computed in polynomial time.

4 Epistemic Profiles and Belief Structures

An essential question is how to realize heterogeneity of agents in multiagent systems. Clearly, being different, when seeing the same thing, agents may perceive it differently and then may draw different conclusions. In order to define the way an agent reasons (e.g., by the use of rules) and to express the granularity of their reasoning (e.g., by varying the level of certain attributes or accuracy of rules expressing the modeled phenomena) we introduce a notion of *epistemic profile*. Epistemic profiles also characterize the manner of dealing with conflicting or lacking information by combining various forms of reasoning (also "light" forms of nonomonotonic reasoning), including belief fusion, disambiguation of conflicting beliefs or completion of lacking information. Especially dealing with inconsistency is important for us. Particular agents may adopt different general methods of the disambiguation (like minimal change strategy) or just implement their own local, application-specific methods via rules encoding knowledge on an expert in the field. This way the flexibility of dealing with inconsistency is formally implemented.

As inconsistency is one of the four logical values, it naturally appears on different reasoning levels. It may be finally disambiguated when the necessary information is in place. This is an intrinsic property of 4QL supported by its modular architecture. As an example, consider a rescue agent trying to save people from the disaster region. However it cannot work in high temperatures. Suppose it has inconsistent information about the temperature there. In the classical approach it would stop him from acting immediately, while in our approach, it may proceed till the moment the situation is clarified.

Tough decisions about conflicting or missing information may be solved by the system designer (application developer) based on their expert knowledge. For instance a rule might say that if some external literal is inconsistent or unknown ($M.l \in \{\mathfrak{u}, \mathfrak{i}\}$) a specific authority source should be consulted (alternatively, the rule cannot be applied).

The following definitions are adapted from [6], where more intuition, explanation and examples can be found. If S is a set then by $\textsc{Fin}(S)$ we understand the set of all finite subsets of S.

Definition 4.1. Let $\mathbb{C} \stackrel{\text{def}}{=} \textsc{Fin}(\mathcal{G}(\mathit{Const}))$ be the set of all finite sets of ground literals over the set of constants *Const*. Then:

- by a *constituent* we understand any set $C \in \mathbb{C}$;
- by an *epistemic profile* we understand any function $\mathcal{E} : \textsc{Fin}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over an epistemic profile* $\mathcal{E}$ we mean $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$, where:
  - $\mathcal{C} \subseteq \mathbb{C}$ is a nonempty set of constituents;
  - $F \stackrel{\text{def}}{=} \mathcal{E}(\mathcal{C})$ is the *consequent* of $\mathcal{B}^{\mathcal{E}}$. $\lhd$

We alternate between the notions of the set of consequents and well-supported models. Epistemic profile is realized via 4QL program, which may consist of several modules.

Definition 4.2. Let $\mathcal{E}$ be an epistemic profile. The *truth value of formula* α w.r.t. belief structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$ and valuation v, denoted by $\alpha(\mathcal{B}^{\mathcal{E}}, v)$, is defined by:[1]

$$\alpha(\mathcal{B}^{\mathcal{E}}, v) \stackrel{\text{def}}{=} \alpha(\bigcup_{C \in \mathcal{C}} C, v). \qquad \lhd$$

5 Communicative Relations and Rule Admissibility Conditions

In multiagent domains many different aspects of inter-agent relations have been studied, e.g., trust, reputation, norms, commitments. They all have a greater scope of influence than just communication. The communicative relations we propose below, can be viewed as selective lens, through which we can see only these parts of the relations involved, which affect communication. They were introduced in [5] for guarding agents' informational stance. Now we extend our perspective to cover also reasoning rules:

1. *communication with authority*: an agent (receiver) is willing to evaluate the interlocutor's (sender, authority) rules even if they contain unknown premises or unknown conclusions,
2. *peer to peer communication*: both parties are viewed as equally credible and important information sources, therefore nobody's opinion prevails a priori. Unknown premises should be resolved before checking the admissibility of the rule. Whereas to recognize unknown conclusions, different application-specific solutions might be applied (see Algorithm 1).

---

[1] Since $\bigcup_{C \in \mathcal{C}} C$ is a set of ground literals, $\alpha(\mathcal{S}, v)$ is well-defined by Table 2.

3. *communication with subordinate*: when dealing with a less reliable source of information, the receiver with an authority would not be willing to risk his beliefs' and epistemic profile consistency. He would evaluate the new rule only when the conclusions are known (i.e. he would not learn new concepts from the subordinates).

In all cases, whenever the rule makes through to Evaluation and the admissibility criterion holds, the agents accept the new rule regardless the communicative relation. Otherwise, when the rule is not admissible, the interested agents engage in conflict resolution via challenge. Recall, that during the complex communication processes, we intend to protect the already possessed knowledge from unknown and ensure that true or false propositions are abandoned for good reasons solely. This is reflected in the knowledge preserving ordering $\leq_k$ on the truth values (Fig. 1).

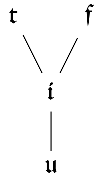

Fig. 1. Knowledge ordering $\leq_k$

Dealing with unknown information is a delicate matter. Indeed, accepting rules with unknown literals is risky for the receiver. If the valuation of the unknown literal is finally established as the sender intended, the receiver's resulting belief structure might no longer be compatible. We solve this problem on a meta-level utilizing the communication relations: rules containing unknown premises are evaluated only when the sender is an authority. Otherwise, the unknown premises need to be resolved first.

As epistemic profiles are 4QL programs, adding a rule to an epistemic profile amounts to adding that rule to the specific module in the program.

Definition 5.1. We define an operation of *adding a rule* $M_i.\ell \mathrel{:\!-} b$ to an epistemic profile $\mathcal{E} = \{M_1, ..., M_n\}$ as follows:

$$\mathcal{E}' = \mathcal{E} \cup \{M_i.\ell \mathrel{:\!-} b\} = \{M_1, ..., M_{i-1}, M_i \cup \ell \mathrel{:\!-} b, M_{i+1}, ..., M_n\}$$

Definition 5.2. Let v be a valuation, l a literal, $\mathcal{C}_i$ the set of constituents, F_i the set of consequents, $\mathcal{E}_i$ the epistemic profile and $\mathcal{B}_i$ the belief structure for $i \in \{a, b\}$. Belief structure $\mathcal{B}_b^{\mathcal{E}_b} = \langle \mathcal{C}_b, F_b \rangle$ is *compatible* with belief structure $\mathcal{B}_a^{\mathcal{E}_a} = \langle \mathcal{C}_a, F_a \rangle$ iff.

$$\forall \ell \in F_a \cap F_b \;\; \ell(\mathcal{B}_a^{\mathcal{E}_a}, v) \;\leq_k\; \ell(\mathcal{B}_b^{\mathcal{E}_b}, v).$$

Definition 5.3. Let $\mathcal{C}$ be a set of constituents, F, F' sets of consequents, $\mathcal{E}, \mathcal{E}'$ the epistemic profiles. Rule $\ell \mathrel{:\!-} b$ is $compatible$ with belief structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$, where $F = \mathcal{E}(\mathcal{C})$ iff. belief structure $\mathcal{B}^{\mathcal{E}}$ is compatible with belief structure $\mathcal{B}^{\mathcal{E}'} = \langle \mathcal{C}, F' \rangle$, such that: $\mathcal{E}' = \mathcal{E} \cup \{\ell \mathrel{:\!-} b\},\ F' = \mathcal{E}'(\mathcal{C})$.

We will allow for a rule to be added into agent's epistemic profile only if it is compatible with the agent's current belief structure.

6 Perceiving Assertions About Rules: the Algorithm

In our framework we deal with five different speech acts: *assert*, *concede*, *request*, *reject* and *challenge* (see Table 3), which allow us to characterize the way the 4QL agents communicate. Below, we present the Algorithm of Perceiving Assertions About Rules. The algorithm, viewed as a complex action, determines what move should an agent make after perceiving an assertion about a reasoning rule. It comprises four phases: *filtering* (Subsection 6.1), *parsing* (Subsection 6.2), *evaluation* (Subsection 6.3) and *belief revision* (Subsection 6.4). Filtering restricts the amount of incoming information, Parsing, in addition, provides means for investigating the message's content. In Evaluation the new rule is examined against the admissibility criterion and in Belief Revision, the resulting belief structure is computed on the basis of the new set of rules.

Filtering and Parsing are more tied to a specific application. In the case of Filtering, the implementations may vary from no filtering at all, to advanced solutions where both properties of the message and the current beliefs of the agent are considered. In the Parsing phase we intended to accent the general concepts, like the importance of the proper treatment of the unknown literals, and leave some space to application dependent decisions. In this spirit we have investigated rules in four conceptual groups depending on the location of the unknown literals in the rule head or body, and proposed a specific solution for dealing with unknown with the use of communicative relations.

In the case of Evaluation and Belief Revision, the solution has a general flavor. As explained in Section 5, the special truth ordering serves as a means to adequately identify possible conflicts or threats to the system, which the new rule might introduce. Thanks to the properties of 4QL, the evaluation of the admissibility criterion is straightforward and the conflicting region can be easily determined by comparing the original and the resulting belief structures. Then, the agent knows if it can harmlessly add the new rule or whether it should engage in a conflict resolution dialogue. Finally, the Belief Revision, as advocated before, is also a general procedure that, based on the Evaluation result, should generate a new belief structure, compatible with the previous one.

Table 3. Speech acts and their intended meaning.

$\texttt{assert}_{S,R}(l :- b)$	Sender S tells the Receiver R the rule $l :- b$
$\texttt{concede}_{S,R}(l :- b)$	Sender S tells the Receiver R that it agrees with the rule $l :- b$
$\texttt{reject}_{S,R}(l :- b)$	Sender S tells the Receiver R that it could not accept the rule $l :- b$
$\texttt{challenge}_{S,R}(l :- b)$	Sender S tells the Receiver R that it disagrees with the rule $l :- b$ and asks for its justification
$\texttt{request}_{S,R}(l)$	Sender S asks the Receiver R information about l

6.1 Filtering

The aim of the filtering phase is to restrict the amount of incoming information and to guard its significance. During this step, the agent filters out noise, unimportant, resource-consuming, or harmful messages. To this end, different properties of the perceived message play a role: the sender, the type of the speech act, the context of the message, etc. Accordingly, different filtering mechanisms can be implemented in 4QL as separate modules, e.g., a module for communicative-relations-based filtering.

If a message makes through Filtering barrier to the Parsing phase, that means it is relevant and significant enough for the agent to consume its resources for handling it.

6.2 Parsing

The goal of parsing is to dissolve a rule into literals and to identify the unknown literals. Then, the receiver's reaction depends both on the communicative relation with the sender and on the rule itself, distinguishing the cases presented below.

Rule head is unknown, rule body is known This means, that the agent recognizes all the premises separately: all the literals in the rule body are either true, false or inconsistent. The novel assembly of literals leads to a new, unknown beforehand conclusion and may be viewed as learning the new concept.

Example 6.1. Let module Tom contain only the following facts: `use(hammer,nail)`, `nail`, `hammer`, `painting`, and a rule: `hanger :– nail, hammer, use(hammer, nail)`. In other words, Tom has a nail, a hammer and a painting, and he can use the hammer and the nail. The rule signifies that Tom can make a hanger if he has a nail and a hammer and he can use them. Suppose Bob has uttered a new rule:

`hangingPainting :– hanger, painting.`

The rule states that that one can achieve a hanging painting if he has a painting and a hanger. For Tom, the rule body is known (literals `painting` and `hanger` are true in Tom's belief structure), but the rule head is unknown. If Tom accepts the new rule he would learn how to hang a painting.

Rule head is known, rule body is unknown This situation relates to the case when some of the premises are unknown, but the conclusion is known. That may be described as widening the knowledge, or making it more detail. Depending on the communication relation, the unknown literals in the rule body can be treated as a possible threat to the consistency of the agent's beliefs (if the sender is a peer or a subordinate) and therefore need further investigation. Alternatively, in case of communication with an authority, the unknowns need not to be resolved a priori (the sender might for example want to communicate some regulations regarding upcoming events, for which some literals' valuations cannot be known beforehand). Here we follow the philosophy of exploiting communicative relations as explained in Section 5.

Example 6.2. Continuing the example from above, the module now contains the following two rules (one known before, one just learnt):

`hanger :– nail, hammer, use(hammer, nail).`

`hangingPainting :– hanger, painting.`

Suppose Bob has uttered another rule:

`hanger :– nail, hammer | borrow(hammer), use(hammer, nail).`

The rule states, that in order to build a hanger one must have a nail, must know how to use the hammer and the nail, as well as must have a hammer or borrow one. In this case, the rule head is known (`hanger` is true), but the rule body is not known (`borrow(hammer)`). If Tom accepts this rule, he would learn another way to build a hanger.

Rule head is known, rule body is known Philosophically, such situation pertains to two different cases: the incoming rule is known already, or the incoming rule combines previously known literals as premises (*Eureka!*). That may be described as knowledge discovery.

Example 6.3. If Bob says: `hammer` :– `hanger`, `nail`, `use`(`hammer`, `nail`), both the head and the body of the rule are known to Tom, which of course does not mean Tom should adopt this rule immediately.

Rule head is unknown, rule body is unknown In that case, the agent is overburdened with new information and, when communicating with a peer or subordinate, should start from resolving the unknown premises first. However, if the sender was an authority, such a rule may get through Parsing to Evaluation.

Example 6.4. If Bob says: `pancake` :– `flour`, `egg`, `milk`, `pan`, `stove`, Tom does not know any of the literals.

Searching for the meaning of the unknown premises requires a sort of information seeking phase (dialogue). This in turn may fail, leading to the rejection of the rule in question. In the course of dialogue the belief structures could evolve, calling for a repetition of the whole procedure, for example, when the sender turned out to be unreliable it is important to perform filtering anew.

If a message makes through Parsing to Evaluation, that means, the agent has all the means to properly evaluate the rule in its belief structure.

6.3 Evaluation

The evaluation stage is the one when the decision about adopting the new rule is made. The agent needs to verify if it can harmlessly add the rule in question to its epistemic profile. The outcome of this process can be twofold:

- if the rule provides conclusions compatible with current beliefs: admit it,
- if the rule provides conclusions incompatible with current beliefs: if possible, try to resolve the contradictions and otherwise reject the rule.

The rule is compatible with the current beliefs, if when added to agent's current epistemic profile, makes the resulting belief structure compatible with the current structure (see Definition5.3). Thus, all literals that were true or false, remain true or false, respectively. Literals that were inconsistent may become true, false, or remain inconsistent. Literals that were unknown may become true, false, inconsistent or remain unknown.

Similarly to the Filtering phase, the possibility of challenging the sender about the rule in question opens the doors for failures. In case of communication problems, or system-specific parameters such as timeouts, the challenge might fail forcing the agent to reject the rule in question. However, a successful completion of a challenge is always a one-side victory:

- either the challenging agent won (the receiver of the rule), and therefore the rule was not legitimate to accept,

- or the opponent won (the sender of the rule) and the receiver has been convinced to accept the rule.

The messages exchanged in this process might have changed the belief structures of communicating agents. In case the challenging agent won, it may terminate the process, even without explicitly rejecting the rule, as the opponent is perfectly clear of the defeat. In case the challenging agent lost, it means that for its new belief structure the rule in question is no longer incompatible. It may proceed to the Belief Revision phase. Challenges about the rules are subject of the upcoming article, but see [5].

If a message makes through Evaluation to Belief Revision, it means the admissibility criterion is met.

6.4 Belief Revision

The aim of belief revision stage is to update the belief structure according to the rule and type of speech act. In case of assertions, agent's individual beliefs as well as shared beliefs must be refreshed. For concessions, only the shared belief base gets updated.

We do not present a new semantics for belief revision[2]. It is rather a technical means to verify to what extent do the new rules interfere with the previously obtained belief structures. When computing the new belief structures, still the information might be lacking and the inconsistencies may occur. In fact this is the merit of our approach. Later on the modular architecture of 4QL allows for dealing with inconsistencies differently on various layers. Afterwards the update of the rule base is almost trivial: if suffices to compute the new well-supported model, which is in P-Time. Of course, there is space for improvement, for instance by examining only the fragments of the previous well-supported model, which would provide better results. However in the worst case still no better than P-Time can be achieved.

In the case of a successful belief revision, an acknowledgement in form of the concession speech act must be sent, in order to notify the sender about the agreement about the rule. A failure at this stage is a very rare incident, however, might happen (if for example the program running the agent is manually killed) and would cause a fatal error, for which to recover from, special means are needed.

If a message makes through Belief Revision, that means, that the rule has been successfully integrated with the current knowledge and the appropriate acknowledgement has been sent to whom it may concern.

6.5 The Algorithm

The Perceiving Assertions About Rules Algorithm takes the following input parameters:

- $\ell :\!- b$. A rule with a body $b = b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$ and a head ℓ, wrapped up in a speech act `assert`.
- S. The sender of the message.
- R. The receiver of the message.
- $\mathcal{E}$. Agent's R epistemic profile.
- $\mathcal{B}_R^{\mathcal{E}} = \langle \mathcal{C}_R, F_R \rangle$. Agent's R belief structure.
- `applicationType`. Application type.

---

[2] For literature see [29–32]

Algorithm 1 Perceiving Assertions About Rules Algorithm

```
procedure PERCEIVE($S$, $R$, $\ell$, $b$, $\mathcal{E}$, $\mathcal{B}^{\mathcal{E}}_{R}$, applicationType)
[Filtering]
    if FilteringModule.allow(speechAct=SA, sender=$S$, ...) IN{$\mathfrak{f}$} then
        go to [End]
    end if
[Parsing]
[Case 1]                                                                ▷ rule head and body recognized
    if $l \in F_R \wedge \forall_{j \in 1..m,\, k \in 1..i_m} b_{j_k} \in F_R$ then      ▷ $l, b_{j_k} \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}$
        go to [Evaluation]
[Case 2]                                                                ▷ only rule body recognized
    else if $\forall_{j \in 1..m,\, k \in 1..i_m} b_{j_k} \in F_R$ then             ▷ $l = \mathfrak{u}, b_{j_k} \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}$
        switch applicationType do
            case "exploratory":
                $\langle \mathcal{B}^{\mathcal{E}}_{R_1}, result \rangle \leftarrow InformationSeekingAbout(\ l\ )$
                if $result$ ==success then restart($\mathcal{B}^{\mathcal{E}}_{R_1}$)   ▷ possibly new belief structure
                else plug-in custom solutions here
                end if
            case "real time": go to [Evaluation]
            case "other": $send$(reject$_{R,S}$($\ell$ :– $b$)
[Case 3]                                                                ▷ only rule head recognized
    else if $l \in F_R$ then                                            ▷ $l \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}, b_{j_k} = \mathfrak{u}$
        if $communicativeRelation(S)$ == "authority" then
            go to [Evaluation]
        else
            for all $j, k : b_{j_k} = \mathfrak{u}$ do                 ▷ execute in parallel
                $\langle \mathcal{B}^{\mathcal{E}}_{R_2}, result \rangle \leftarrow send$(request$_{R,S}(b_{j_k})$)
                if $result$ ==success then restart($\mathcal{B}^{\mathcal{E}}_{R_2}$)    ▷ new belief structure
                else $send$(reject$_{R,S}$($\ell$ :– $b$))
                end if
            end for
        end if
[Case 4]                                                                ▷ rule head and rule body unknown
    else                                                                ▷ $l, b = \mathfrak{u}$
        go to [Case 3]                                                  ▷ resolve the body first
    end if
[Evaluation]
    if $l \in F_R$ then                          ▷ rule head known: check if the rule is admissible
        $\mathcal{E}_{TEST} \leftarrow \mathcal{E} \cup \{\ell :\!- b\}$          ▷ add the rule to a candidate epistemic profile
        $F_{R_{TEST}} \leftarrow \mathcal{E}_{TEST}(\mathcal{C})$
        $\mathcal{B}^{\mathcal{E}}_{R_{TEST}} \leftarrow \langle \mathcal{C}_R, F_{R_{TEST}} \rangle$     ▷ compute the candidate belief structure
        if $incompatible(\mathcal{B}^{\mathcal{E}}_{R_{TEST}}, \mathcal{B}^{\mathcal{E}}_{R})$ then      ▷ try to resolve the problem
            $\langle \mathcal{B}^{\mathcal{E}}_{R_3}, result, winner \rangle \leftarrow send$(challenge$_{R,S}$($\ell$ :– $b$))
            if $result$ ==success then
                if $winner == R$ then                                   ▷ the rule was not admissible
                    go to [End]
                else restart($\mathcal{B}^{\mathcal{E}}_{R_3}$)    ▷ the opponent won, restart with the new belief structure
                end if
            else $send$(reject$_{R,S}$($\ell$ :– $b$)
            end if
        else go to [BeliefRevision]                                     ▷ belief structures compatible
        end if
    else                                                                ▷ rule head unknown
        switch $communicativeRelation(S)$ do
            case "authority": go to [BeliefRevision]
            case "peer": plug-in custom solutions here
            case "subordinate": go to [End]
    end if
```

Algorithm 2 Perceiving Rules Algorithm (continued)

```
[BeliefRevision]
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\ell :\!- b\}$                ▷ add the rule to the epistemic profile
    $F_R \leftarrow \mathcal{E}_{(}\mathcal{C})$
    $\mathcal{B}_R^{\mathcal{E}} \leftarrow \langle \mathcal{C}_R, F_R \rangle$                ▷ compute the new belief structure
    $send(\texttt{concede}_{R,S}(\ell :\!- b))$
[End]
end procedure
```

7 Example

Let us present a more thorough example demonstrating some of the cases described above. Recall, that Tom is an agent realized[3] via 4QL program outlined in Figure 2.

```
module tom:
  relations:   a(literal), use(literal, literal), borrow(literal).
  rules:
    a(hanger) :- a(nail), a(hammer), use(hammer, nail).
    a(X) :- borrow(X).
  facts:
    a(nail).
    a(hammer).
    a(painting).
    use(hammer, nail).
end.
```

Fig. 2. Example of a 4QL program realizing agent Tom.

Tom's epistemic profile consist of four facts ($\texttt{hammer}, \texttt{painting}, \texttt{use}(\texttt{hammer}, \texttt{nail})$, $\texttt{nail}$), and two rules: one, describing his ability to borrow things and the other, depicting how to make a hanger. Tom's belief structure (the well-supported model) is:

$$B_T = \{\texttt{nail}, \texttt{hammer}, \texttt{painting}, \texttt{use}(\texttt{hammer}, \texttt{nail}), \texttt{hanger}\}$$

Suppose Bob has uttered the following rule (see Section 2):

$$\texttt{assert}_{B,T}(\texttt{hangingPainting} :\!- \texttt{hanger}, \texttt{painting}.)$$

The rule head is unknown to Tom (it is absent from his belief structure: B_T) but the rule body is recognized: both literals are in the belief structure (but notice that `hanger` is not a fact from the epistemic profile). According to the algorithm, Tom needs to exercise the admissibility criterion for the new rule. He adds the rule to his candidate epistemic profile and computes the new belief structure:

$$B'_T = \{\texttt{nail}, \texttt{hammer}, \texttt{painting}, \texttt{use}(\texttt{hammer}, \texttt{nail}), \texttt{hanger}, \texttt{hangingPainting}\}$$

Now, B'_T is compatible with B_T, because all literals that were true remained true and one literal which was unknown is now true. Tom concludes that he can add the rule

---

[3] For modeling and for computing well-supported models we use the 4QL interpreter, developed by P. Spanily. It can be downloaded from http://www.4ql.org/ .

to his epistemic profile permanently. In this way, Tom learnt how to achieve something from already available means.

Another interesting case concerns agents' ability to learn alternative ways of achieving goals. In Figure 3 the new module Tom, equipped with the newly learnt rule is presented. Consider now the case that Tom does not have the hammer at hand (fact `hammer` is false). Tom's new belief structure is the following:

$$B''_T = \{\mathtt{nail}, \neg\mathtt{hammer}, \mathtt{painting}, \mathtt{use}(\mathtt{hammer}, \mathtt{nail})\}$$

```
module tom:
  relations:   a(literal), use(literal, literal), borrow(literal).
  rules:
    a(hangingPainting) :- a(hanger), a(painting).
    a(hanger) :- a(nail), a(hammer), use(hammer, nail).
    a(X) :- borrow(X).
  facts:
    a(nail).
    -a(hammer).
    a(painting).
    use(hammer, nail).
end.
```

Fig. 3. Tom with a new rule added, but without the hammer.

Suppose Bob has uttered the following rule, providing another way to achieve a hanger:

$$\mathtt{hanger} :\!- \mathtt{nail},\ \mathtt{hammer} \mid \mathtt{borrow}(\mathtt{hammer}),\ \mathtt{use}(\mathtt{hammer},\ \mathtt{nail}).$$

All literals are known to Bob, so the candidate belief structure B'''_T can be computed:

$$B'''_T = \{\mathtt{nail}, \neg\mathtt{hammer}, \mathtt{painting}, \mathtt{use}(\mathtt{hammer}, \mathtt{nail})\}$$

The new rule can be safely added to Tom's epistemic profile. Notice that if Tom borrowed the hammer (a fact `borrow(hammer)` was added to Tom's epistemic profile), he would achieve `hangingPainting` now (see $B_{T'''_{borrowed}}$). It would have been impossible without the new rule from Bob (compare with $B_{T''_{borrowed}}$):

$$\begin{aligned} B_{T'''_{borrowed}} = \{&\mathtt{nail}, \mathtt{hammer}, \neg\mathtt{hammer}, \mathtt{borrow}(\mathtt{hammer}), \mathtt{painting}, \mathtt{use}(\mathtt{hammer}, \mathtt{nail}), \\ &\mathtt{hanger}, \mathtt{hangingPainting}\} \end{aligned}$$

$$B_{T''_{borrowed}} = \{\mathtt{nail}, \mathtt{hammer}, \neg\mathtt{hammer}, \mathtt{borrow}(\mathtt{hammer}), \mathtt{painting}, \mathtt{use}(\mathtt{hammer}, \mathtt{nail})\}$$

8 Discussion and Conclusions

This paper aligns with our ultimate research goal, namely, a paraconsistent model of agents' communication. In order to construct complex dialogues, the speech acts theory provides the necessary building material. We initiated our research program by proposing a paraconsistent framework for perceiving new facts via four different speech acts: assert, concede, request and challenge [5]. In this work we make a second step by allowing the agents to perceive assertions about reasoning rules as well.

The application of Speech Acts theory to communication in MAS dates back to late 20th century [19]. Since then it proved to be a practical tool for creating various agent communication languages such as KQML and FIPA ACL [10] as well as formal models of dialogues [1, 14, 15, 24].

Perceiving new information, whether it is some previously unknown fact, a new valuation of a proposition, or a reasoning rule, typically requires belief revision [21]. Our implementation tool of choice, the rule-base query language 4QL was designed in such a way that the inconsistency is tamed and treated naturally in the language. As a great benefit, belief revision turned out to be dramatically simplified and obtained in P-Time.

In this paper we focus on the case, when the information in question reflects the procedural component on the agents' epistemic profile, namely: the rules. This subject has hitherto received little attention. Even though in [22], a cooperative rule learning approach for exchanging sets of rules among agents has been presented and in [23], a formalism has been proposed that allows for discussing inference rules acceptability by agents, none of the approaches deals explicitly with unknown and possibly inconsistent information. Trying to fill this gap in [5] and our recent paper, the next step will concern challenging rules. In this context the aspect of validity and sensibility of the rules themselves, which wasn't treated here, will be vital.

References

1. K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152:157–206, 2005.
2. J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, second edition, 1975. Edited by J. O. Urmson and M. Sbisa.
3. S. de Amo and M. Pais. A paraconsistent logic approach for querying inconsistent databases. *International Journal of Approximate Reasoning*, 46:366–386, 2007.
4. F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9:145–158, 2001.
5. B. Dunin-Kęplicz, A. Strachocka, A. Szałas and R. Verbrugge. Perceiving Speech Acts under Incomplete and Inconsistent Information. *KES AMSTA, Frontiers of Artificial Intelligence and Applications* 252, s. 255-264, IOS Press, 2013.
6. B. Dunin-Kęplicz and A. Szałas. Epistemic profiles and belief structures. *KES-AMSTA, Lecture Notes in Computer Science* 7327, s. 360-369, Springer-Verlag, 2012.
7. D.Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed EXPTIME-complete. *Journal of Logic and Computation*, 16(6):765–787, 2006.
8. M. Dziubiński, R. Verbrugge, and B. Dunin-Kęplicz. Complexity issues in multiagent logics. *Fundamenta Informaticae*, 75(1-4):239–262, 2007.
9. F. Fages. Consistency of Clark's completion and existence of stable models. *Methods of Logic in Computer Science*, 1:51–60, 1994.

10. FIPA. 2002. `http://www.fipa.org/`.
11. J. Małuszyński and A. Szałas. Living with inconsistency and taming nonmonotonicity. In O. de Moor, G. Gottlob, T. Furche, and A. Sellers, editors, *Datalog 2.0*, volume 6702 of *LNCS*, pages 384–398. Springer-Verlag, 2011.
12. J. Małuszyński and A. Szałas. Partiality and Inconsistency in Agents' Belief Bases. *KES-AMSTA, Frontiers of Artificial Intelligence and Applications* 252, s. 3-17, IOS Press, 2013.
13. V. Mascardi, D. Demergasso, and D. Ancona. Languages for programming BDI-style agents: an overview. In F. Corradini, F. De Paoli, E. Merelli, and A. Omicini, editors, *WOA 2005 - Workshop From Objects to Agents*, pages 9–15, 2005.
14. S. Parsons and P. McBurney. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation*, 12:415–439, 2003.
15. H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.
16. J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, 1969.
17. A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.
18. D. Walton and E. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.
19. Cohen, Philip R. and Levesque, Hector J. *Rational interaction as the basis for communication.* Technical Report 433, SRI International, Menlo Park, (CA), 1988
20. M. Kaiser and R. Dillmann and O. Rogalla. *Communication as the basis for learning in multi-agent systems* ECAI 96 Workshop on Learning in Distributed AI Systems, 1996
21. Fabio Paglieri and Cristiano Castelfranchi. *Revising Beliefs Through Arguments: Bridging the Gap between Argumentation and Belief Revision in MAS* Proceedings of the 1 st workshop on Argumentation in MAS (ArgMAS), 2004, 78–94, Springer
22. S. Costantini. *Learning by knowledge exchange in logical agents* WOA 2005: Dagli, 2005
23. Peter Mcburney and Simon Parsons. *Tenacious Tortoises: A formalism for argument over rules of inference* Computational Dialectics (ECAI 2000 Workshop), 2000
24. Munindar Singh. *A semantics for speech acts* Annals of Mathematics and Artificial Intelligence, Springer Netherlands, 1012-2443, 47–71, 1993
25. Linder, B. and Hoek, W. and Meyer, J.-J.Ch. Actions that make you change your mind. *KI-95: Advances in Artificial Intelligence*, 98:185–196, 1995.
26. Henry Prakken. Modelling Reasoning about Evidence in Legal Procedure. *Proceedings of the Eighth International Conference on Artificial Intelligence and Law*, 119–128, 2001.
27. Bench-Capon, T.J.M. and Prakken, H. Using Argument Schemes for Hypothetical Reasoning in Law. *Artificial Intelligence and Law*, 18(2), 153-174, 2010.
28. D. Gabbay and A. Hunter, Making Inconsistency Respectable: A Logical Framework for Inconsistency in Reasoning, Part I - A Position Paper. *Fundamentals of Artificial Intelligence Research: International Workshop FAIR'91 Proc.*, 19-32, 1991.
29. M. Winslett, Updating logical databases. Cambridge University Press, 1990.
30. van Harmelen, Frank, Lifschitz, Vladimir, Porter and Bruce, Handbook of Knowledge Representation, Elsevier Science, 2007.
31. J. J. Alferes, J. A. Leite , L. M. Pereira, H. Przymusinska and T. C. Przymusinski, Dynamic Logic Programming, Procs. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning, Trento, Italy, pp. 98-109, 1998.
32. J. J. Alferes, A. Brogi, J. A. Leite and L. M. Pereira, Evolving Logic Programs, 8th European Conference on Logics in Artificial Intelligence (JELIA'02), pp. 50-61, LNCS 2424, 2002.
33. Béziau, Jean-Yves and Carnielli, Walter Alexandre and Gabbay, Dov M, Handbook of paraconsistency, College publications, 2007.
34. D. Walton, C. Reed and F. Macagno, Argumentation Schemes, Cambridge University Press, 2008.

Web Intelligence and Agent Systems: An International Journal 5 (2015) 1–5
IOS Press

Paraconsistent Argumentation Schemes [1]

Barbara Dunin-Kęplicz [a] and Alina Strachocka [a,*]
[a] *Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland*
E-mail: {keplicz,astrachocka}@mimuw.edu.pl

Abstract

Various types of everyday arguments are represented as *argumentation schemes*, originating from the Legal Argumentation literature. The recent achievements in this domain can be applied to multi-agent settings to enrich the paradigmatic aspects of communication and reasoning. Agents typically populate complex environments where incompleteness and inconsistency of information is rather a rule than exception. Although the problem how to tackle inconsistencies is already present in argumentation, a *paraconsistent* (that is, tolerating inconsistency) approach is still missing from the literature. The contribution of this research is a *computationally-friendly framework* for formalizing *paraconsistent argumentation schemes*. This is achieved by extending agent's reasoning capabilities with non-deductive methods rooted in argumentation. To this end we provide a generic paraconsistent program template for implementation of various argumentation schemes.

Our methodology is strongly influenced by ideas underlying 4QL: a four-valued, rule-based, DATALOG$^{\neg\neg}$-like query language. Thanks to its properties, the tractability of the solution (so hardly obtainable in logical modeling) has been reached. The paper concludes with examples of several paraconsistent argumentation schemes implemented in 4QL.

Keywords: multi-agent systems, communication, nonmonotonic reasoning, paraconsistent argumentation schemes

1. Modeling Assumptions

The logical modeling of complex phenomena appearing in the context of intelligent distributed systems sometimes lacks realism. What emerges, is often both computationally complex and idealized theories not fitting well the modeled reality. A great deal of this mismatch lies in the quality of information available in dynamic and unpredictable environments. Intelligent agents, viewed as autonomous information sources, may perceive the surrounding reality differently while building their informational stance. The information they need to handle comes from multiple sources of diverse credibility, quality or significance. Even though consistency of their beliefs is a desirable property, in practice it is hard to achieve. Thus, when modeling real-world situations, ignorance and inconsistency of information occurs naturally. However, instead of making a reasoning process trivial, what is a hindrance in classical logical systems, we view inconsistency as a first-class citizen and try to efficiently deal with it. This leads to creating logical systems which tolerate inconsistencies, i.e., to *paraconsistent* systems. On the other hand, as missing knowledge can be completed and inconsistent information can be disambiguted, the realistic modeling of agency requires *nonmonotonic* reasoning mechanisms, where new information may invalidate previously obtained conclusions. Such approach to lacking and inconsistent information is the basis of realistic models of agency. Despite the rich field of non-monotonic reasoning and paraconsistent formalisms (see Section 1.1 and 1.2 for a survey), in general a *tractable* approach that combines both aspects was missing from the multi-agent systems (MAS) literature.

The approaches to modeling agency typically feature the *informational* and *motivational* stance of an agent, comprising *beliefs*, *intentions* and *commitments*.

---

[1] Supported by the Polish National Science Centre grant 2015/17/N/ST6/03642

[*] Corresponding author. E-mail: astrachocka@mimuw.edu.pl

From multiple available frameworks (see a discussion about realistic models of agency in Section 1.3), our solution falls into the category of *rule-based systems*: we associate individual agents with *programs* composed of a set of *facts* and *reasoning rules* organized into a layered architecture by the use of *modules*.

To reflect the diversity of possibilities we view agents as heterogeneous reasoners. Following Dunin-Kęplicz and Szałas [20], the way the individual agents deal with conflicting or lacking information is encoded in their *epistemic profile* (see Section 3). Typically, an epistemic profile embodies agents' reasoning capabilities. These in turn influence the agent's deductive processes to finally affect their *belief structures*, i.e., agents' informational stance [20]. Moreover, various agents may reason differently using diverse methods of information disambiguation.

This paper is a part of a larger research program, whose overall goal is a creation of a paraconsistent model of agents' communication, suitable for dealing with unsure and incomplete information especially in applications related to multi-agent systems. Generally, MAS are created for the synergistic effect of collaborating agents, thus their essence is complex interactions. They are vital to the paradigmatic activities like *coordination*, *collaboration* and *negotiation*, which naturally include phases of *communication* and *reasoning*. Therefore, an adequate modeling of communication is one of the challenges in building autonomous agents.

Communication has a long tradition as an important topic in computer science, specifically in (distributed) artificial intelligence and recently in MAS [45,14]. Starting from fixed communication protocols in distributed systems, we now attempt to approach flexible *dialogues* among agents (see e.g., [13]). To achieve the goal of agents communicating freely in the paraconsistent world, we build upon Austin and Searle's theory of *speech acts* [44,2] and Walton and Krabbe semi-formal theory of dialogue [55]. We view the dialogue participants as independent information sources, which try to *expand*, *contract*, *update*, and *revise* their beliefs through communication. Taking this perspective, speech acts are "actions that change your mind" [52] and provide the necessary building blocks to construct complex dialogues, such as *information seeking*, *inquiry*, *persuasion*, *negotiation* or *deliberation*.

We have initiated our research program, by proposing a paraconsistent framework for perceiving new facts [18]. That enabled the agents to discuss their informational stance, i.e.,:

- inform one another about values of different propositions via *assertions*,
- ask for other agents' values via *requests*,
- acknowledge the common values via *concessions*, and
- question the contradictory values via *challenges*.

The second step was a preliminary analysis on communication involving reasoning rules [16]. In that paper, a model for assertions and concessions regarding reasoning rules was proposed, permitting the agents to accept or reject another agent's reasoning rule according to a defined *admissibility criterion*. Recently, these building blocks were used to construct tractable, sound and complete inquiry dialogues in the paraconsistent settings [23]. In the current paper we show how to extend agent's reasoning capabilities with non-deductive methods like argumentation skills, which will later enable agents to conduct arguments over reasoning rules. To this end, we study so-called *argumentation schemes* [56] (see Section 1.3).

Indeed, the question whether to adopt, challenge or reject a reasoning rule has no single straightforward answer. This issue has been studied from multiple angles. In the Belief Revision approach the influence of reasoning rules on agents' knowledge bases was studied. Our solution proposed in [16] followed this approach. A different approach leverages the social choice theory methods where the decision about the admissibility of a rule is made on the social level (see e.g., [22]). Finally, yet another research thread concerns the methods originating form the Legal Reasoning, where argumentation schemes are proxies for an ontology of reasoning methods. In such approach a reasoning rule can be accepted if it is an instance (or a template) of an established argumentation scheme.

1.1. Paraconsistency in Multi-agent Settings

In real-world applications one should accept uncertainty and inconsistency of information, assuming that four types of situations may occur:

- fact a holds,
- fact a does not hold,
- it is not known whether a holds (no source has any information about a),
- information about a is inconsistent (some sources claim a holds, other that a does not hold).

Modeling assumptions of this research require the use of a paraconsistent and paracomplete formalism.

Put crudely, paraconsistency is a property of logics whose logical consequence relation is not explosive (i.e., where *ex contradictione quodlibet* (ECQ) does not hold). Indeed, the principle of explosion is controversial to paraconsistent logicians who argue that "the move from a contradiction to an arbitrary formula does not seem like reasoning" and provide examples of such absurd "proofs" [40].

Needless to say, paraconsistency was studied from multiple angles (see e.g., [26,7,11,40] for paraconsistent reasoning techniques):

- *discussive logic* of Jaśkowski [27] aimed at ensuring that contradictory sentences do not arise by blocking the rule of adjunction;
- *preservationist school* of Schotch and Jennings [43] for reasoning with consistent subsets of premisses;
- *relevance logics* (Orlov, Belnap, Anderson, see [1] and references therein) required that premisses are relevant to the conclusion;
- *C-System* of da Costa [10] (and its generalization: *logics of formal inconsistency*) allowed to distinguish consistent sentences from inconsistent ones and reason *about* them differently;
- Priest's *logic of paradox* [38] utilized third truth value (*both*) and identified designated values (*true*, *both*) to allow reasoning about paradoxes;
- *logic of Belnap* [3] added *unknown* and *inconsistent* truth values to reason about information from many sources.

Although to model phenomena such as lack and inconsistency of information, the Belnap's four-valued logic is commonly used, it often provides counter-intuitive results in ares we focus on (see [15,53] for details). Let us recall the well-known example to show one problematic situation.

Example 1 *A family owns two cars: c_1 and c_2. Then, $safe(c_1) \vee safe(c_2)$ expresses whether the family has a safe car. The safety of car c_1 was checked by two different mechanics m_1 and m_2. The first expert m_1 confirmed the safety of the car but the second expert m_2 denied the car is safe. Car c_2 has not gone through any safety tests yet. Thus, the truth values of $safe(c_1)$ and $safe(c_2)$ are: $\mathfrak{i}$ and $\mathfrak{u}$ respectively. When $\vee$ is defined by Belnap's truth ordering, we get $safe(c_1) \vee safe(c_2) = \mathfrak{i} \vee \mathfrak{u} = \mathfrak{t}$. However, due to contradictory results of safety tests, the safety of car c_1 is unclear. Moreover, we know nothing about safety of car b. Similar example shows that also Belnap's $\wedge$ operator yields counter-intuitive results.*

The logic underpinning our implementation tool (see Section 5) does not share such problems therefore is better suited for this research.

1.2. Nonmonotonic Reasoning in Agency

The second important aspect of realistic modeling is allowing that "additional information may invalidate conclusions" [50]. Indeed, agents are situated in environments where only incomplete information is available, so any monotonic formalism (where new information never invalidates conclusions) is inadequate to capture this phenomenon.

Currently, there is a variety of nonmonotonic techniques (see e.g. Chapter 6 in [50] for general nonmonotonic/defeasible reasoning techniques):

- Reiter's *default reasoning* [41], consisting of applying defaults, i.e., meta-rules of the form "in the absence of any information to the contrary assume ..." [50];
- *Closed World Assumption* (CWA) [42], to represent how databases handle negative information;
- Moore's *autoepistemic logic* [34] to formalize how perfectly rational agents form beliefs;
- McCarthy's *Circumscription* [33] to model "jumping to conclusions" by minimizing the extent of *abnormal* predicate;
- *Preferential models* [28] based on preference relation expressing typicality of possible worlds (e.g., Shoham's preference logic);
- Nute's *Defeasible Logic* [35] encompassing strict and defeasible rules together with a preference relation among defeasible rules.

As typically these nonmonotonic techniques were designed to formalize phenomena appearing in commonsense reasoning, they are suitable for modeling rational agents. On the other hand, *argumentation schemes* attempt to classify the various types of everyday arguments, utilizing the ideas underlying the formalisms described above. As characterized in [56], *argumentation schemes (AS)*

> "are argument forms that represent inferential structures of arguments used in everyday discourse, and in special contexts like legal argumentation, scientific argumentation and especially in AI".

Each scheme is accompanied by a set of *critical questions*, used to evaluate the argument (see e.g., Table 1). Although the various argumentation schemes may *rep-*

resent different types of reasoning (e.g., deduction, induction, abduction, presumption, see also the tentative classification given in [56], Chapter 10), in general their goal is to model plausible, thus defeasible, reasoning.

1.3. Looking for Tractability

The entire line of our research is characterized by a shift in perspective: instead of creating complex logical theories, we tailor them to their tractable versions, like rule-based systems. Then, instead of reasoning in logical systems of high complexity we query paraconsistent knowledge bases.

Many important aspects of classical agency have to be adjusted when adopting a paraconsistent semantics. First of all, the AGM postulates for Belief Revision [8] are no longer valid (but see [48,39]). On the other hand, some assumptions underlying classical formalizations of agency (Dynamic Epistemic Logic (DEL) [49], intention logic [9]; BDI [25] & KARO [51] frameworks) are unpractical: real agents do not have infinite resources (like time) available for reasoning and they are not logically perfect reasoners. Therefore a formalism which does not require such assumptions would be preferred.

Our approach is strongly influenced by ideas underlying 4QL: a four-valued paraconsistent query language introduced by Małuszyński and Szałas (see [29, 30] and Section 5 for details). We have chosen 4QL as a suitable tool for formalizing *paraconsistent argumentation schemes (PAS)*, as it provides simple, yet powerful constructs for application of a range of well-known nonmonotonic techniques (see [19] for examples of default reasoning, autoepistemic reasoning, defeasible reasoning and the (Local) Closed World Assumption expressed in 4QL).

The contribution of the paper is extending epistemic profiles with paraconsistent argumentation schemes, providing templates for their implementation and ensuring tractability of reasoning by using 4QL.

This paper is an extended version of our conference paper [17]. In comparison to [17] we:

- provide implementation and discussion of additional five argumentation schemes,
- show scheme embedding, based on the Reputation and Prudence schemes to reason about trust,
- elaborate on the role of communicative relations and their relationship to the Ethotic Argument,
- revise and extend discussion and background.

The paper is structured in the following way. First, in Section 2, we describe the computational approach to argumentation schemes. Next, in Section 3 the language and logic used throughout the paper as well as the concept of epistemic profiles and belief structures are introduced. Section 4 presents the main contribution of this paper, namely the formalization of the paraconsistent argumentation scheme as a part of epistemic profile. In Section 5 our implementation tool of choice is described. Section 6 presents the formalization of paraconsistent argumentation schemes with use of the notion of *well-supported models* as well as the implemented solution, which we illustrate on four argumentation schemes: Expert Opinion, Position to Know, Perception and Ethos. In Section 7 we show how to embed schemes, utilizing the Reputation and Prudence scheme for reasoning about trust. Section 8 concludes the paper.

2. Computational Perspective on Paraconsistent Argumentation Schemes

The aim of this work is augmentation of agents' inferential capabilities with the methods rooted in argumentation. Our primary goal is to provide means for implementing the paraconsistent argumentation schemes in a tractable way. This computational approach aims at combining techniques of classical reasoning with non-monotonic, argumentative reasoning. The conclusions obtained with the use of both methods exist on equal terms, but possibly can be used in different situations.

Following [37], we simplify the set of critical questions to those pointing to the specific undercutters, called *exceptions*. Thus, the exceptions serve as means to both undercut the argument and shift the burden of proof to the other side [56].

Observation that most common sense rules have exceptions gave birth to nonmonotonic reasoning techniques [50], however the difficulty lied in specifying all the 'abnormal' cases. Here, through modeling critical questions as exceptions, we try to minimize the set of abnormalities under which the scheme is not applicable. All in all, the opponent may attack the claim in three ways:

- by rebutting the premisses,
- by rebutting the conclusion,
- by undercutting the argument using the exceptions.

In addition, we encourage a rigorous separation of various aspects of reasoning:

- the information,
- the opinion about the information and its source,
- the disambiguation of inconsistent information.

To illustrate this, consider our running example: the argument from Expert Opinion. It is summarized in Table 1 (see also its implementation in 4QL presented in Figure 4). The first column conveys the original form of the argument, including the scheme (premises, conclusion) and critical questions (as in [56]). The second column presents the adapted, paraconsistent version of the argument. There, the set of critical questions is replaced with the set of exceptions and the original premises are encoded by four-valued literals. Importantly, the conclusion is tetravalent ($v(is(X)) = V$ means that the value of $is(X)$ is V, where V can be one of: t, f, i u). Consider an expert e states that "it is unknown whether a medicament m is safe to use during pregnancy". In this case, the conclusion of the Expert Opinion scheme, `safe(m, pregnancy)`, should take the *unknown* value, which is far more expressive and accurate than *false*, which would have been chosen in the standard 2-valued approach under the Closed World Assumption.

Table 1

Expert Opinion Argumentation Scheme

	Original Argumentation Scheme
Scheme	A is an expert in domain D A asserts that X is true X is within D
	X is true
Critical Questions	How credible is A as an expert source? Is A an expert in domain D? What did A assert that implies X? Is A personally reliable as a source? Is X consistent with what other experts assert? Is A's assertion of X based on evidence?
	Paraconsistent Argumentation Scheme
P-Scheme	isExpert(A,D) assert(A,X,V) inDomain(X,D)
	$v(is(X)) = V$
Exceptions	¬isReliable(A) ¬evidenceBased(A,X,V)

Now, the various aspects of the reasoning can be easily distinguished:

- object level, e.g., the claim $\mathit{assert}(A, X, V)$,
- meta-level:
  - ∗ the reasoning about the claim, e.g., using $\mathit{inDomain}(X, D)$, $\mathit{evidenceBased}(A, X, V)$,
  - ∗ the reasoning about the sender agent, e.g., using $\mathit{isExpert}(A, D)$, $\mathit{isReliable}(A)$.
- meta-meta-level: reasoning about the compatibility of experts' opinions (here it is excluded from the scheme, as it corresponds to disambiguation of inconsistent information arising from multiple sources of opinion).

Our formalization concerns a mechanism for specifying any argumentation scheme. In addition, when formalized in 4QL, the solution becomes tractable. Since 4QL captures all tractable queries, the expressiveness is maintained.

3. Language and Epistemic Profiles

The heterogeneity of agents' means, among others, that even when seeing the same thing, the particular individuals may draw different conclusions. The powerful notion of *epistemic profile* [20] explicitly models this problem. In general, it defines the way an agent reasons (e.g., in this paper a rule-based agent implementation is assumed), including the manner of dealing with conflicting or lacking information (e.g., by combining various forms of reasoning available to the agent, including belief fusion, disambiguation of conflicting beliefs or completion of lacking information).

The following definitions are adapted from [20], where intuition and examples can be found.

In what follows all sets are finite except for sets of formulas. We deal with the classical first-order language over a given vocabulary without function symbols presented in [29,20,46]. We assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols. We shall use this notation in the following definitions.

Definition 1 A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with $\bar{\tau}$ being a sequence of parameters, $\bar{\tau} \in (\mathit{Const} \cup Var)^k$, where k is the arity of R. *Ground literals over Const*, denoted by $\mathcal{G}(\mathit{Const})$, are literals without variables, with all constants in *Const*. If $\ell = \neg R(\bar{\tau})$ then $\neg\ell \overset{\text{def}}{=} R(\bar{\tau})$. ◁

Though we use classical first-order syntax, the semantics substantially differs from the classical one as truth values $\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}$ (true, inconsistent, unknown, false) are explicitly present; the semantics is based on sets of ground literals rather than on relational structures. This allows one to deal with lack of information as well as inconsistencies. Because 4QL is based on the same principles, it can directly be used as implementation tool.

The semantics of propositional connectives is summarized in Table 2. Observe that definitions of $\wedge$ and $\vee$ reflect minimum and maximum with respect to the ordering:

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}, \tag{1}$$

as argued in [12,29,53].

Table 2

Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$ (see [53,29,31]).

$\wedge$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\vee$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\rightarrow$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$		$\neg$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$

It is worth noting that whenever truth values are restricted to $\{\mathfrak{f}, \mathfrak{t}\}$, the semantics we consider is compatible with the semantics of classical first-order logic.

Let $v : \mathit{Var} \longrightarrow \mathit{Const}$ be a *valuation of variables*. For a literal ℓ, by $\ell(v)$ we understand the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$.

Definition 2 The *truth value* $\ell(L, v)$ of a literal ℓ w.r.t. a set of ground literals L and valuation v, is defined by:

$$\ell(L,v) \overset{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$

◁

For a formula $\alpha(x)$ with a free variable x and $c \in \mathit{Const}$, by $\alpha(x)^x_c$ we understand the formula obtained from α by substituting all free occurrences of x by c. Definition 2 is extended to all formulas in Table 3, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 2.

Table 3

Semantics of first-order formulas.

If α is a literal then $\alpha(L, v)$ is defined in Definition 2;
$(\neg\alpha)(L, v) \overset{\text{def}}{=} \neg(\alpha(L, v))$;
$(\alpha \circ \beta)(L, v) \overset{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
$(\forall x \alpha(x))(L, v) = \min\limits_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$,
where min is the minimum w.r.t. ordering (1);
$(\exists x \alpha(x))(L, v) = \max\limits_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$,
where max is the maximum w.r.t. ordering (1).

Belief structures can now be defined as in [20]. If S is a set, then $\textsc{Fin}(S)$ represents the set of all finite subsets of S.

Definition 3 Let $\mathbb{C} \overset{\text{def}}{=} \textsc{Fin}(\mathcal{G}(\mathit{Const}))$ be the set of all finite sets of ground literals over constants in *Const*. Then:

- a *constituent* is any set $C \in \mathbb{C}$;
- an *epistemic profile* is any function $\mathcal{E} : \textsc{Fin}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over epistemic profile* $\mathcal{E}$ is meant a structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$; here $\mathcal{C} \subseteq \mathbb{C}$ is a nonempty set of constituents and $F \overset{\text{def}}{=} \mathcal{E}(\mathcal{C})$ is the *consequent* of $\mathcal{B}^{\mathcal{E}}$. ◁

Final beliefs are represented as *consequents*.

Notice, that the epistemic profile, being any function, can encode any reasoning schema (especially when we disregard complexity issues). In this paper we extend agent's repertoire to include user-defined argumentation schemes, that employ incomplete and uncertain information. In the sequel, we will show the theoretical foundations, and then the implementation in 4QL.

4. Argumentation Schemes as Parts of Epistemic Profiles

Until now, the epistemic profiles, being arbitrary functions, conveyed all reasoning capabilities of agents (and groups of agents) [20], including their communicative strategies [18]. Here we distinguish yet another component, namely, argumentation schemes that extend an agent's (or a group's) epistemic profile.

4.1. Paraconsistent Argumentation Schemes: Intuition

Argumentation schemes are modeled with the use of the sets of *premisses* and *exceptions*, the latter one replacing the concept of critical questions (see Section 6 for the alternative formalization using the well-supported models of 4QL modules). In our approach, the conclusion of the paraconsistent argumentation scheme can be: $true$, $false$, $inconsistent$ or $unknown$. We assume that a **scheme has been applied** when it leads to $true$, $false$ or $inconsistent$ conclusions. Otherwise (when the conclusion's value is $unknown$) the **scheme was not applicable**. This happens in two cases:

- when the premisses are lacking, or
- when the exceptions are present.

If the conclusion may be drawn, then its value is logically equivalent to conclusion resulting from the premisses.

Let us give some intuitions first, before defining the paraconsistent argumentation scheme. Since we admit premisses, exceptions and conclusions to take any of the four logical values, what happens when inconsistent and missing information regards the premisses of the argumentation scheme? To this end, recall the argument from Expert Opinion (see Table 1). When the premise "a is an expert in domain D" is:

- $unknown$ or $false$, it cannot serve to draw conclusions,
- $true$ or $inconsistent$, it can be used. In the worst scenario, the overall outcome will become $inconsistent$.

Thus, attacking an argument on premisses can be achieved by:

- proving one of them is $false$ or $unknown$ (**rebutting**),
- proving their inconsistency (**undercutting**).

Next, let's consider the set of exceptions. It contains all the exceptions potentially applicable in the scheme. Whenever any of them becomes true, the schema is blocked and cannot be applied. For example, consider the exception regarding the expert's reliability:

- if the expert is not reliable ($\neg isReliable$ is $true$), the Expert Opinion scheme cannot be applied,
- otherwise, the exception is not triggered.

Therefore, as regards exceptions, their attacking power matters only when they are $true$.

4.2. Paraconsistent Argumentation Schemes: Definition

In our framework for paraconsistent argumentation schemes we deal with the three sets of ground literals: Premisses (P), Exceptions (E) and Conclusions (Con), and a function $\mathcal{PAS}\ (\{P, E\}) = Con$, which represents the **paraconsistent argumentation scheme**.

The high-level structure of our running example is presented in Figure 1. The ovals correspond to the sets of ground literals, arranged into above-mentioned sets of Conclusions, Premisses and Exceptions, as well as: Expert, Ontology, Evidence, Assertions and Reliability, which are specific for a particular scheme (here the Expert Opinion utilizes the set of Assertions for constructing both the Premisses and Exceptions). The arrows (from Premisses and Exceptions to Scheme) represent the function of the paraconsistent argumentation scheme ($\mathcal{PAS}$).

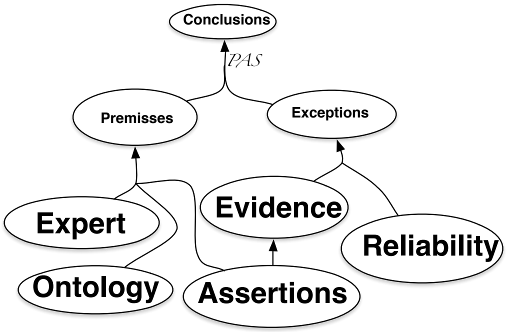

Figure 1. Modular architecture of Expert Opinion Scheme

The elements of the Conclusions set (literals) are **conclusions** of the argumentation scheme as expressed by the function $\mathcal{PAS}$. In case of single-threaded reasoning (reasoning about one subject at a time), the Conclusions set is a singleton. In case of a multi-threaded reasoning (reasoning about many subjects simultaneously), the set of Conclusions can be larger, with one literal corresponding to one conclusion on the subject.

The set of Premisses contains **candidates for conclusion** of the scheme. They are obtained by means specific to every argumentation scheme.

The elements of the set of Exceptions are **triggers** that, when present, forbid the respective candidate conclusion from being drawn. Intuitively, a conclusion c cannot be obtained when the exceptions indicate $\neg c$.

Ultimately, the conclusion of the scheme is obtained in the following way. If there exists a tetravalent candidate for a conclusion $c \in Premisses$ (value of c is **not** $unknown$), check whether there exists a trigger blocking this candidate: $\neg\ c \in\ Exceptions$ (value of $\neg c$ is $true$).

- If there is no such a trigger, the candidate conclusion becomes the ultimate conclusion of the scheme.
- Otherwise, the scheme cannot be applied causing that the value of $c\ \in\ Conclusions$ is $unknown$.

To sum up (adopting the notation from Definition 4), a conclusion c is established based on the supporting arguments given by the set of Premisses (i.e. $c(P, v) = \mathfrak{t}$) and (lack of) rebutting triggers provided by the set of Exceptions (i.e. $\neg c(E, v) \neq \mathfrak{t}$).

Definition 4 Recall that

- $\mathbb{C} = \textsc{Fin}(\mathcal{G}(\mathit{Const}))$ stands for the set of all finite sets of ground literals over the finite set of constants Const,
- $v : Var \longrightarrow \mathit{Const}$ is a valuation of variables.
- by a constituent we understand any set $C \in \mathbb{C}$.

Let:

- P and E be two constituents, representing the set of premisses and exceptions, respectively,
- $\mathcal{S} = \{P, E\}$ be a nonempty set of constituents ($\mathcal{S} \subseteq \mathbb{C}$),
- $Con \in \mathbb{C}$ be a finite set of ground literals, representing the conclusions.

Then, paraconsistent argumentation scheme is a partial function $\mathcal{PAS} : \textsc{Fin}(\mathbb{C}) \longrightarrow \mathbb{C}$, $\mathcal{PAS}\ (\{P, E\})$ $= Con$, such that:

$$c(Con, v) \overset{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ iff } & c(P, v) = \mathfrak{t} \text{ and } \neg c(E, v) \neq \mathfrak{t}; \\ \mathfrak{i} \text{ iff } & c(P, v) = \mathfrak{i} \text{ and } \neg c(E, v) \neq \mathfrak{t}; \\ \mathfrak{u} \text{ iff } & c(P, v) = \mathfrak{u} \text{ or } \neg c(E, v) = \mathfrak{t}; \\ \mathfrak{f} \text{ iff } & c(P, v) = \mathfrak{f} \text{ and } \neg c(E, v) \neq \mathfrak{t}. \end{cases}$$

By belief structure over $\mathcal{PAS}$ we mean $\mathcal{B}^{\mathcal{PAS}} = \langle \mathcal{S}, Con \rangle$, where:

- $\mathcal{S} = \{P, E\}, \mathcal{S} \subseteq \mathbb{C}$ is a nonempty set of constituents;
- $Con \overset{\text{def}}{=} \mathcal{PAS}(\mathcal{S})$ is the *consequent* of $\mathcal{B}^{\mathcal{PAS}}$.

We identify the belief structure over $\mathcal{PAS}$ with the instance of a paraconsistent argumentation scheme. $\triangleleft$

The above definition presents the paraconsistent argumentation scheme as a partial function: a fragment of agent's epistemic profile that expresses agent's argumentative skills. The implementation of PAS in 4QL is presented in Definition 7, where also the analogy between both definitions is explained.

5. 4QL: an Implementation Tool

The rule language 4QL has been introduced in [29] and further developed in [31,46]. In 4QL, beliefs are distributed among *modules*. The 4QL language allows for negation in premisses and conclusions of rules. In particular, negation in rule heads may lead to inconsistencies. 4QL is based on the four-valued logic described in Section 3. The semantics of 4QL is defined by *well-supported models* [29,30,31,46], i.e., models consisting of (positive or negative) ground literals, where each literal is a conclusion of a derivation starting from facts. For any set of rules, such a model is uniquely determined:

> "Each module can be treated as a finite set of literals and this set can be computed in deterministic polynomial time" [29,31].

Thanks to this correspondence and the fact that 4QL captures PTIME, the constituents and consequents of Definition 3, can be directly implemented as 4QL modules [21].

Typically, in multi-agent architectures, 4QL would be situated in the layer that processes qualitative information, as opposed to the lower-level quantitative information processing layer, for which various techniques, including fuzzy, rough set and probabilistic approaches, can be used (e.g., for image, voice and other sensor data processing).

For specifying rules and querying modules, we adapt the language of [46]. To this end we need the notion of multisource formulas defined as follows.

Definition 5 A *multisource formula* is an expression of the form: $m.A$ or $m.A \in T$, where:

- m is a module name;
- A is a first-order or a multisource formula;
- $T \subseteq \{\mathbf{t}, \mathbf{i}, \mathbf{u}, \mathbf{f}\}$.

We write $m.A = v$ (respectively, $m.A \neq v$) to stand for $m.A \in \{v\}$ (respectively, $m.A \notin \{v\}$). ◁

The intuitive meaning of a multisource formula $m.A$ is:

> "return the answer to query expressed by formula A, computed within the context of module m".

The value of '$m.A \in T$ ' is:

$$\begin{cases} \mathbf{t} \text{ when the truth value of } A \text{ in } m \text{ is in the set } T; \\ \mathbf{f}\ otherwise. \end{cases}$$

Let $A(X_1, \ldots, X_k)$ be a multisource formula with $X_1, \ldots, X_k$ being its all free variables and D be a finite set of literals (a belief base). Then A, understood as a query, returns tuples $\langle d_1, \ldots, d_k, tv \rangle$, where $d_1, \ldots, d_k$ are database domain elements and the value of $A(d_1, \ldots, d_k)$ in D is tv.

Definition 6

- *Rules* are expressions of the form:

  $$\textit{conclusion} :\!-\ \textit{premisses}. \tag{2}$$

  where *conclusion* is a positive or negative literal and *premisses* are expressed by a multisource formula.
- A *fact* is a rule with empty premisses (such premisses are evaluated to $\mathbf{t}$).
- A *module* is a syntactic entity encapsulating a finite number of facts and rules.
- A 4QL *program* is a set of modules, where it is assumed that there are no cyclic references to modules involving multisource formulas of the form $m.A \in T$. ◁

```
MODULE r :
RELATIONS: is(literal).
           do(literal).
RULES:
  is(heat):- is(fire).
 -is(fire):- is(lowtemp).
  do(extinguish):- is(heat)|is(fire).
 -do(extinguish):- -is(heat),-is(fire).
  is(X):-   perceived.is(X).
 -is(X):-  -perceived.is(X).
END.

MODULE perceived :
RELATIONS: is(literal).
FACTS:
   is(fire).
  -is(heat).
END.
```

Figure 2. Example of a 4QL program.

Openness of the world is assumed, but rules can be used to close it locally or globally.

Let us illustrate 4QL, consider Figure 2, where we use syntax of the 4QL interpreter inter4QL.[1] The program shown in Figure 2 consists of two modules: r and $perceived$. The program uniquely determines the following well-supported model for module $perceived$:

$$\{is(fire), \neg is(heat)\} \tag{3}$$

and the following well-supported model for module r:

$$\{is(heat), \neg is(heat), is(fire), do(extinguish)\}. \tag{4}$$

From the epistemic profile perspective, the two modules: r and $perceived$ define sets of ground literals (3) and (4) and can be seen as a belief structure with one constituent $perceived$ and a consequent r. The epistemic profile is defined by rules of module r. As well-supported models are sets of ground literals (with deterministic polynomial time data complexity), we alternate between the notions of the set of consequents and well-supported models (as mentioned in Sections 3 and 6).

---

[1] The interpreter, developed by P. Spanily and revised by Ł. Białek, can be downloaded from http://www.4ql.org/ .

6. Paraconsistent Argumentation Schemes in 4QL

Each argumentation scheme is a standalone entity, with a dedicated 4QL module, containing two specific rules:

```
is(X):- Premisses.is(X),
        -Exceptions.is(X) in {false,unknown,incons}.
```

and

```
-is(X):- -Premisses.is(X),
         -Exceptions.is(X) in {false,unknown,incons}.
```

The multisource formulas in the bodies of the rules pertain to two other specific sub-modules: one corresponding to the premisses and one to the exceptions. Intuitively, these rules express the mechanism of drawing the scheme conclusions using the premisses and exceptions in the way described in Section 4. Altogether, the 3-modular 4QL architecture reflects the structure of the argumentation scheme:

- the set of $premisses$ is translated to the module `SN_Premisses`,
- the set of $exceptions$ is captured in the module `SN_Exceptions`,
- the $conclusion$ (`is(X)`) is evaluated in the `SN` (scheme name) module.

These three modules constitute the standard approach to argumentation schemes in 4QL. They are captured in a generic template of a 4QL program consisting of the three modules: `SN`, `SN_Exceptions` and `SN_Premisses` (see Figure 3). Evaluation of an instance of an argumentation scheme amounts to computing the well-supported model for the module `SN`, implementing this scheme (see e.g., Figure 4 for an implementation of the Expert Opinion scheme).

The two rules encoded in module `SN` allow for drawing *true*, *false* or *inconsistent* conclusions. Recall that a conclusion c can be obtained when the exceptions do not conclusively indicate $\neg c$. Again, let's first investigate, when the scheme cannot be applied?

1. **The conclusion cannot be drawn due to the exceptions.** This situation is encoded using the literal `-SN_Exceptions.is(X)` in both rules.
   - When it is $true$, both rules in the `SN` module cannot be executed (as the rule premisses evaluate to *false*, see Table 2).
2. **The conclusion cannot be drawn due to the lack of premisses**. Consider valuations of the first literal: `(-)SN_Premisses.is(X)`.
   - The only situation where both rules cannot be executed due to it, is when the value of `(-)SN_Premisses.is(X)` is *unknown*.
   - Indeed, otherwise:
     - ∗ if it's *true*, the first rule would be executed,
     - ∗ if it's *false*, the second rule would be executed,
     - ∗ if it's *inconsistent*, both rules would be executed.

   Consider a situation, where there are both exceptions from the argumentation scheme and also not all premisses are present. This would correspond to the following case for both rules:

   - the first literal is $unknown$,
   - the second literal is $true$.

   Then, the conjunction would evaluate to $unknown$ (see Table 2) and none of the rules applies.

If there are no exceptions prohibiting the scheme from being applied, the conclusion `is(X)` in the `SN` module evaluates to the same value as `is(X)` in the `SN_Premisses` module (as explained in Section 4).

As mentioned before, the burden of proving the premisses lies on the proponent. The opponent can fight a successful argument is two ways. Either by *rebutting* it (by proving the conclusion is *unknown*) or by *undercutting* it (by proving the conclusion is *inconsistent*). The first goal can be achieved in two ways:

- by showing that in the `SN_Premisses` module at least one of the premisses is *false* or *unknown*, or
- by showing that `-SN_Exceptions.is(X)` is true, i.e, at least one of the exceptions is *true*.

To undercut a successful argument the opponent may attack the true premisses by proving their inconsistency.

In what follows we present the definition of a paraconsistent argumentation scheme, which is, in fact, an implementation of Definition 4 in 4QL. The analogy between both definitions is founded on the following observations:

- the functions are represented as the sets of rules (4QL modules),
- the sets of ground literals correspond to well-supported models (see Section 5).

```
MODULE SN :
RELATIONS: is(literal).
RULES:
      is(X) :-  SN_Premisses.is(X),-SN_Exceptions.is(X) in {false,unknown,incons}.
     -is(X) :- -SN_Premisses.is(X),-SN_Exceptions.is(X) in {false,unknown,incons}.
END.


MODULE SN_Exceptions :
RELATIONS:
      ex_1(literal,...).
      ex_n(literal,...).
      is(literal).
RULES:
      -is(X) :- ex_1(X,...) |...| ex_n(X,...).
END.


MODULE SN_Premisses :
RELATIONS:
      p_1(literal,...).
      p_m(literal,...).
      assert(literal,literal,literal).
      is(literal).
RULES:
      is(X) :- p_1(...), ... , p_m(...), assert(A,X,true).
     -is(X) :- p_1(...), ... , p_m(...), assert(A,X,false).
      is(X) :- p_1(...), ... , p_m(...), assert(A,X,incons).
     -is(X) :- p_1(...), ... , p_m(...), assert(A,X,incons).
END.
```

Figure 3. General 4ql Template for Argumentation Schemes

Definition 7 *Recall that L stands for the set of ground literals with constants in Const. Let $Exceptions$, $Premisses$ and $Scheme$ be three sets of rules. A* Paraconsistent Argumentation Scheme *is a tuple $PAS = \langle Exceptions, Premisses, Scheme\rangle$, such that if:*

- $\mathcal{M}_{\mathcal{S}}$ *is a* well supported model *of Scheme,*
- $\mathcal{M}_{\mathcal{P}}$ *is a* well supported model *of Premisses,*
- $\mathcal{M}_{\mathcal{E}}$ *is a* well supported model *of Exceptions,*

then $\exists c \in L$, such that

$$\mathcal{M}_{\mathcal{S}}(c) \overset{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ iff } \mathcal{M}_{\mathcal{P}}(c) = \mathfrak{t} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg c) \neq \mathfrak{t}; \\ \mathfrak{i} \text{ iff } \mathcal{M}_{\mathcal{P}}(c) = \mathfrak{i} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg c) \neq \mathfrak{t}; \\ \mathfrak{u} \text{ iff } \mathcal{M}_{\mathcal{P}}(c) = \mathfrak{u} \text{ or } \mathcal{M}_{\mathcal{E}}(\neg c) = \mathfrak{t}; \\ \mathfrak{f} \text{ iff } \mathcal{M}_{\mathcal{P}}(c) = \mathfrak{f} \text{ and } \mathcal{M}_{\mathcal{E}}(\neg c) \neq \mathfrak{t}. \end{cases}$$

Is such case we define c as the conclusion *of the paraconsistent argumentation scheme.* ◁

The above definition presents PAS as a tuple of specific 4QL modules (sets of rules). Figure 5 illustrates it with the rectangles depicting the 4QL modules, as in the Figures 4, 7 and 6. On the other hand, Definition 4 expresses PAS as a partial function: a fragment of agent's epistemic profile. Here, the $\mathcal{PAS}$ function (see Definition 4) is represented by the $Scheme$ module (Scheme rectangle in the Figure 5). The set of conclusions of the scheme (Con) corresponds to the well-supported model of the $Scheme$ module ($\mathcal{M}_{\mathcal{S}}$). In general, the ovals in Figure 1 correspond to the well-supported models of the modules depicted by the rectangles in the Figure 5.

Any argumentation scheme that can be represented as PAS can be implemented in 4QL. Obtaining conclusion c of the scheme is in polynomial time, as it amounts to computing the well-supported model of the module $Scheme$. This complexity result allows to reason and compare conclusions obtained with the use of multiple different argumentation schemes at once.

6.1. Example: Argument from Expert Opinion

As a showcase for our solution, consider the Argument from Expert Opinion. The modules implementing the scheme (`eo`, `eoExceptions`, `eoPremisses`) and the agent (`agent`) are shown in Figure 4. Module `agent`, represents the reasoning capabilities of the $agent$. $Agent$'s epistemic profile is composed of two rules:

```
 is(X) :-  eo.is(X,A).
-is(X) :- -eo.is(X,A).
```

It utilizes the Expert Opinion scheme to adjudicate about the final conclusion $is(X)$ on the matter X. In this simple strategy of dealing with inconsistencies, the $agent$ acknowledges all opinions of experts A regarding X, that were admitted by the Expert Opinion module. This way the object, meta- and meta-meta levels (see Section 2) are separated. For a more restrictive strategy for dealing with inconsistency, a different solution can be applied.

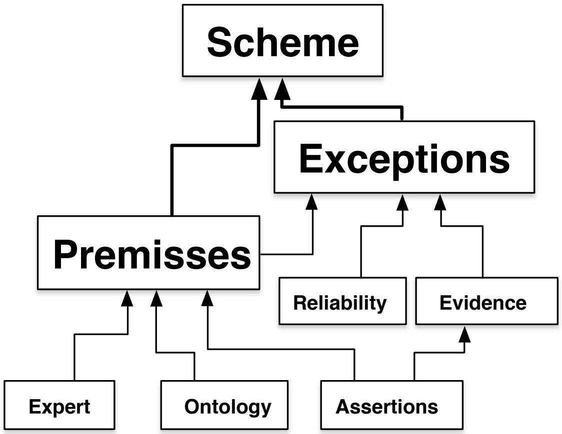

Figure 5. Modular architecture of Expert Opinion Scheme

The implementation of the Expert Opinion scheme deals with two explicit exceptions (as argued in Section 2):

- evidence-based claim: `-evidence.claim(A,X)` and
- reliability: `-reliability.isReliable(A)`.

The sub-modules implementing particular exceptions are presented in Figure 6. The `reliability` and `evidence` modules include the starting sets of facts, stating that:

- bob is not trusted: `isTrusted(bob)` and
- evidently, it rained: `sensorsLog(bob,rain)`.

The sub-modules (`expert`,`ontology`, `assertions`) implementing particular premisses are presented in Figure 7. Here, and for the remaining schemes we equip the modules with some basic facts that can be treated as test cases for a given argumentation scheme.

Module `expert` decides that an expert in field F is an agent that does a job J which is related to that field, e.g., a meteorologist is an expert in weather. The `assertions` module translates the messages perceived from agents (A) about their opinions (values $\mathfrak{t}$, $\mathfrak{f}$, $\mathfrak{i}$) on various subjects (X) into assertions in 4QL. The starting sets of facts include the information about agents meteorologists: bob, $ag1$, $ag2$, $ag3$.

This example allows us to test various scenarios of missing and inconsistent information appearing in the Expert Opinion scheme. If we load the program to the interpreter and ask: `agent.is(X)`, we would get information that there are no known facts (the well-supported model of the `agent` module is empty). Indeed, there are no perceived messages in the module `assertions` to start from. Now, if we add the following facts to the module `assertions`:

```
* perceived(bob,rain,true),
* perceived(ag1,rain,true),
* perceived(ag2,rain,false),
```

and reload the module, the same question would yield the following results: `agent.is(rain):inconsistent`.

If we intend to learn which experts' opinions were essential to that verdict, we should ask: `eo.is(X,Y)`. This query leads to the generation of the well-supported model of the `eo` module. It contains the following facts: $\{eo.is(rain, ag1), -eo.is(rain, ag2)\}$. Clearly, bob's opinion didn't count (as absent from the model). On the other hand, both $ag1$'s and $ag2$'s opinions were admitted. Indeed, the only triggered exception, was the one concerning bob (the well-supported model of the `eoExceptions` module contains only $\{-is(rain, bob)\}$). Although bob's claim was based on evidence ($evidence.claim(bob, rain) = \mathfrak{t}$) he was not reliable ($reliability.isReliable(bob) = \mathfrak{f}$).

```
MODULE agent :
RELATIONS: is(literal).
RULES:
  is(X) :-  eo.is(X,A).
 -is(X) :- -eo.is(X,A).
END.


MODULE eo :
RELATIONS: is(literal, literal).
RULES:
  is(X,A):-  eoPremisses.is(X,A), -eoExceptions.is(X,A) in {false,unknown,incons}.
 -is(X,A):- -eoPremisses.is(X,A), -eoExceptions.is(X,A) in {false,unknown,incons}.
END.


MODULE eoExceptions :
RELATIONS: is(literal, literal).
RULES:
 -is(X,A):-  eoPremisses.is(X,A),-reliability.isReliable(A)|-evidence.claim(A,X).
 -is(X,A):- -eoPremisses.is(X,A),-reliability.isReliable(A)|-evidence.claim(A,X).
END.


MODULE eoPremisses :
RELATIONS: is(literal, literal).
RULES:
  is(X,A):-expert.isExpert(A,D),ontology.inDomain(X,D),assertions.asserted(A,X).
 -is(X,A):-expert.isExpert(A,D),ontology.inDomain(X,D),-assertions.asserted(A,X).
END.
```

Figure 4. Argumentation Scheme from Expert Opinion

Now, what should happen, if in the `reliability` module's fact: `-isTrusted(bob)` we replace *bob* with *ag2* ? Clearly, the two experts whose testimony is admitted by the Expert Opinion scheme are now: $ag1$ and *bob*: $eo.is(rain, bob) = \mathbf{t} \wedge eo.is(rain, ag1) = \mathbf{t}$. The overall conclusion becomes: $agent.is(rain) = \mathbf{t}$.

```
MODULE reliability :
RELATIONS:
   isReliable(literal).
   isTrusted(literal).
RULES:
   isReliable(A) :-  isTrusted(A).
  -isReliable(A) :- -isTrusted(A).
 FACTS:
  -isTrusted(bob).
END.


MODULE evidence :
RELATIONS:
   claim(literal,literal).
   sensorsLog(literal,literal).
RULES:
   claim(A,X) :- assertions.asserted(A,X), sensorsLog(A,X).
  -claim(A,X) :- assertions.asserted(A,X), -sensorsLog(A,X).
 FACTS:
   sensorsLog(bob,rain).
END.
```

Figure 6. Modules for Exceptions

```
MODULE expert :
RELATIONS:
 isExpert(literal,literal).
 pair(literal,literal).
 is(literal,literal).
RULES:
 isExpert(A,F) :- is(J,A), pair(F, J).
FACTS:
 pair(weather, meteorologist).
 is(meteorologist, bob).
 is(meteorologist, ag1).
 is(meteorologist, ag2).
 is(meteorologist, ag3).
END.
```

```
MODULE ontology :
RELATIONS: inDomain(literal,literal).
FACTS: inDomain(rain,weather).
END.

MODULE assertions :
RELATIONS:
  perceived(literal, literal, literal).
  asserted(literal, literal).
RULES:
  asserted(A,X):- perceived(A,X,true).
 -asserted(A,X):- perceived(A,X,false).
  asserted(A,X):- perceived(A,X,incons).
 -asserted(A,X):- perceived(A,X,incons).
END.
```

Figure 7. Modules for Premisses

6.2. Example: Argument from Perception

Consider the following argumentation scheme, which allows agents to reason about percepts. The classical scheme has two premisses and one critical question (see Table 4). Since the second premise expresses the link between the first premise and the conclusion, it is redundant. The critical question works as an undercutter for that link and as such remains in our framework. The Argument from Perception scheme, viewed solely as an argumentation or reasoning pattern, is quite simple. All the difficulty connected with assessing the reliability of perception is extracted to the relation `isPerceptionReliable()`. In practical applications, this relation would be implemented with use of a dedicated module responsible for adjudicating about the reliability of perception. In our example, the final conclusion about $is(rain)$ will be that of $alice$, since *bob*'s perception is not reliable.

Table 4

Perception: from Original to Paraconsistent

	Original Argumentation Scheme as in [56]
Scheme	a has a φ image (an image of a perceptible property). To have a φ image (an image of a perceptible property) is a prima facie reason to believe that the circumstances exemplify φ.
	It is reasonable to believe that φ is the case.
*	Are the circumstances such that having a φ image is not a reliable indicator of φ?
	Adapted to Communicative Settings
Scheme	a has a φ percept
	φ is the case
**	The circumstances are such that having a φ percept is a reliable indicator of φ.
	Paraconsistent Argumentation Scheme
Scheme	percept(Agent,X)
	is(X)
**	isPerceptionReliable(Agent,X)

* - Critical Questions, ** - Assumptions

6.3. Example: Argument from Position to Know

Let's consider another example of an argumentation scheme: argument from Position to Know. The original form of this scheme is presented in Table 5. It consist of just two premisses and three critical questions. Notice that two critical questions are redundant. After eliminating them, the only relevant critical question that remains concerns the reliability of the source. In our framework it is expressed as an assumption "a is a reliable source" implemented using relation `isReliable()` in `pkExceptions` module in Figure 9.

Table 5

Position to Know: from Original to Paraconsistent.

	Original Argumentation Scheme as in [56]
Scheme	a is in the position to know whether A is true a asserts that A is true
	A is true
*	Is a in the position to know whether A is true? Is a a honest (trustworthy, reliable) source? Did a assert that A is true?
	Adapted to Communicative Settings
Scheme	a is in the position to know whether A is true a asserts that A is true
	A is true
**	a is a reliable source
	Paraconsistent Argumentation Scheme
Scheme	posKnow(Agent,X) assert(Agent,X,Value)
	is(X)
**	isReliable(Agent)

* - Critical Questions, ** - Assumptions

For simplicity, instead of expanding the relation `isReliable()`, in this example we provide the relevant information as a fact (the information that *bob* is not reliable encodes `-isReliable(bob)`). However, notice that it could have been given differently, e.g., with use of the following rule:

```
isReliable(A):- reliability.high(A).
```

Such approach would be suitable to reflect that the information about reliability is obtained via an advanced argumentation or reasoning process encapsulated in module `reliability`, rather than by a simple hardcoded fact. By adding a relevant rule with an external literal referring a sub-module responsible for, in this case, reasoning about reliability, we can embed various argumentation schemes. We show an example

Figure 8. Perception Argumentation Scheme in 4QL

```
MODULE tom :
RELATIONS: is(literal).
RULES:
 is(X):-  p.is(X).
-is(X):- -p.is(X).
END.


MODULE p :
RELATIONS: is(literal).
RULES:
  is(X):-  pPremisses.is(X,A), -pExceptions.is(X,A) in {false,unknown,incons}.
 -is(X):- -pPremisses.is(X,A), -pExceptions.is(X,A) in {false,unknown,incons}.
END.


MODULE pExceptions :
RELATIONS:
 is(literal, literal).
 isPerceptionReliable(literal, literal).
RULES:
 -is(X,Agent):- -isPerceptionReliable(Agent,X).
FACTS:
 -isPerceptionReliable(bob,rain).
END.


MODULE pPremisses :
RELATIONS:
 percept(literal, literal, literal).
 is(literal, literal).
RULES:
 is(X,A):- percept(A,X,true).
-is(X,A):- percept(A,X,false).
 is(X,A):- percept(A,X,incons).
-is(X,A):- percept(A,X,incons).
FACTS:
 percept(bob, rain, true).
 percept(alice, rain, false).
END.
```

of such embedding for the Argument from Ethos (see Section 7).

Recall, that the original form of the Position to Know scheme does not contain facts. Here, and for the remaining schemes, we equip the modules with some basic facts so that the programs shown in figures can be executed with use of 4QL interpreter. In this example, $is(fire)$ is absent (*unknown*) from the well-supported model of module `tom`. Although all the premises are in place, the source of the information (*bob*) is not reliable. Therefore, the conclusion could not be drawn due to the exceptions: $-pkExceptions.is(fire)$ is $true$.

Figure 9. Position to Know Argumentation Scheme in 4QL

```
MODULE tom :
RELATIONS: is(literal).
RULES:
 is(X):-  pk.is(X).
-is(X):- -pk.is(X).
END.

MODULE pk :
RELATIONS: is(literal).
RULES:
 is(X):-  pkPremisses.is(X), -pkExceptions.is(X) in {false,unknown,incons}.
-is(X):- -pkPremisses.is(X), -pkExceptions.is(X) in {false,unknown,incons}.
END.

MODULE pkExceptions :
RELATIONS:
 isReliable(literal).
 is(literal).
RULES:
 -is(X):- pkPremisses.assert(A,X,V), -isReliable(A).
FACTS:
 -isReliable(bob).
END.

MODULE pkPremisses :
RELATIONS:
 posKnow(literal, literal).
 assert(literal, literal, literal).
 is(literal).
RULES:
 is(X):- posKnow(A, X), assert(A, X, trueL).
-is(X):- posKnow(A, X), assert(A, X, falseL).
 is(X):- posKnow(A, X), assert(A, X, inconsL).
-is(X):- posKnow(A, X), assert(A, X, inconsL).
FACTS:
 posKnow(bob, fire).
 assert(bob, fire, trueL).
END.
```

Figure 10. Ethos Argumentation Scheme in 4QL

```
MODULE tom :
RELATIONS: is(literal).
RULES:
 is(X):-  e.is(X,Y).
-is(X):- -e.is(X,Y).
END.

MODULE e :
RELATIONS: is(literal, literal).
RULES:
 is(X,A):-  ePremisses.is(X,A), -eExceptions.is(X,A) in {false,unknown,incons}.
-is(X,A):- -ePremisses.is(X,A), -eExceptions.is(X,A) in {false,unknown,incons}.
END.

MODULE eExceptions :
RELATIONS:
 is(literal,literal).
 qualitiesRelevant(literal).
 evidenceBased(literal,literal,literal).
RULES:
 -is(X,A):- ePremisses.goodQualities(A), -qualitiesRelevant(X) |
            ePremisses.assert(A,X,Value), -evidenceBased(A,X,Value).
FACTS:
  qualitiesRelevant(rain).
 -evidenceBased(bob,rain,trueL).
END.

MODULE ePremisses :
RELATIONS:
 goodQualities(literal).
 assert(literal, literal, literal).
 is(literal, literal).
RULES:
 is(X,A):- goodQualities(A), assert(A,X,true).
-is(X,A):- goodQualities(A), assert(A,X,false).
 is(X,A):- goodQualities(A), assert(A,X,incons).
-is(X,A):- goodQualities(A), assert(A,X,incons).
FACTS:
 goodQualities(bob).
 goodQualities(alice).
 assert(bob, rain, true).
 assert(alice, rain, false).
END.
```

6.4. Example: Ethotic Argument

The Argument from Ethos is commonly used to adjudicate about truth or falsity of a proposition on the grounds of the qualities of the information source. The original form of the scheme is given in Table 6. The two non-redundant critical questions are: the one about relevancy of the character qualities and the one about the claim being evidence-based.

In the example in Figure 10, the main premise (`goodQualities` of an agent) is given with the use of simple facts (see module `ePremisses`). The two implemented critical questions are: the one about (ir)relevancy of the qualities of character and the one about the statement being evidence-based. With use of this argumentation scheme as an example, we intend to shed more light on the way scheme embedding is done in 4QL. In Section 7 we show how other argumentation schemes can be embedded into this one, to draw conclusions about the `goodQualities` of an agent in a more sophisticated way.

Table 6

Ethos: from Original to Paraconsistent

	Original Argumentation Scheme as in [56]
Scheme	If a is a person of good (bad) moral character, then what a says should be accepted as more plausible (rejected as less plausible). a is a person of good (bad) moral character.
	What a says should be accepted as more plausible (rejected as less plausible).
*	Is a a person of good (bad) moral character? Is character relevant in the dialogue? Is the weight of presumption claimed strongly enough warranted by the evidence given?
	Adapted to Communicative Settings
Scheme	a is an agent of good combination of qualities a asserts that A is the case
	A is the case
**	Agent's qualities are relevant in the dialogue. The weight of A is strongly enough warranted by the evidence given.
	Paraconsistent Argumentation Scheme
Scheme	goodQualities(Agent) assert(Agent, X, Value)
	is(X)
**	qualititesRelevant(X) evidenceBased(Agent,X,Value)

* - Critical Questions, ** - Assumptions

7. Embedding Argumentation Schemes

Let's recall the original Ethotic Argument and our paraconsistent version of it (Table 6). As we do not intend to over-antropomorphize our artificial agents, the classical notion of good (bad) moral character should be treated in a way that is more relevant to MAS. One approach can be to focus on qualities such as: veracity, prudence, perception and cognitive skills as proposed in [54]. However, in [54], the problem whether agent's statements should be considered or disregarded was adjudicated with use of a higher-level concept: the credibility of an agent, which was expressed with the use of the *credibility function*. The character traits recalled above play a role in judging agent's credibility and therefore plausability of his arguments. In addition, "when one of these traits is a relevant basis for an adjustment in a credibility function, there is a shift to a subdialogue in which the argumentation in the case is re-evaluated" [54].

A slightly different approach to the same problem is presented in [36], where the role of credibility, plays *trust*, which is considered to be "a mechanism for managing the uncertainty about autonomous entities and the information they deal with". Furthermore "trust should be reason-based, which suggests argumentation as a mechanism for constructing arguments (reasons) for and against adopting beliefs and pursuing actions, and explicitly recording the agents that need to be trusted."

Finally, in [18], yet another solution to the same problem has been proposed, with the use of *communicative relations*, which comprise various aspects of communication and "can be viewed as selective lens, through which we can see only these parts of the relations involved, which affect communication".

The common part of these approaches is the need for assessing plausibility of arguments put forward by agents. In this paper we lean towards the solution that uses the notion of *communicative relation*, which covers a wide range of relations or concepts that could be relevant for assessing the plausability of agent's arguments from the ethotic standpoint.

Figure 11 represents an example of a 4QL module (`communicationX`) of some application, that encodes the communicative relations (for brevity we write `cr` for "communicative relation") for some agent X. By `cr(A,high)` we mean that agent X has a high communicative relation with agent A. There are three levels of communicative relations required in the application: *high*, *medium* and *low*. Further, the example presents also a module that encodes one of the components of the communicative relation: trust (module `trust`). As we can see, the agent adjudicates about trust using two argumentation schemes: *reputation* (module `rep`) and *prudence* (module `pru`).

One advantage of this approach is an easy fine-tuning of applications. It is fairly straightforward to change the `communicationX` module presented in Figure 11 such that for different applications, particular concepts (trust, power, veracity, etc.) are taken into consideration with different weights.[2] In addition, if the application developers want to add yet another factor that should influence communicative relations between the agents, they need to implement a relevant submodule (like the module `trust`) and update the body of the rule that computes the level for the communicative relation (in the module `communicationX`).

7.1. *Reputation and Prudence Argumentation Schemes*

As argued above, the factors which may influence the communicative relation are diverse. Here we intend to present how the trust component can be realized in the paraconsistent setting and embedded in the Ethotic Argumentation scheme. For this purpose, we utilize two argumentation schemes for reasoning about trust, as proposed in [36]: Reputation and Prudence.

Table 7 summarizes both schemes, giving their original form in the first column, the adapted version with reduced critical questions in the second column, and the paraconsistent version in the third column.

Next, the 4QL modules that encode these schemes are presented in Figure 12 and Figure 13 respectively (notice we omit the standard top-level module `rep` and `pru` and present only the modules responsible for computing premisses and exceptions).

Let's consider the Reputation scheme first. The only critical question that is not redundant is the one regarding the manipulation of reputation score. We make it explicit in the `repExceptions` module. In case of the Prudence scheme, we need to assess the accuracy of both risk estimations, as well as the assumption that there is no other agent for which the risk of trust is lower. This all is done in the `pruExceptions` module (see Figure 13).

7.2. *Embedding Reputation and Prudence in Argument from Ethos*

Now that we have shown the paraconsistent implementation of the Reputation and Prudence Argumentation Schemes for reasoning about trust, let's look at their embedding in the Argument from Ethos. The embedding of the schemes is realized in the following way. Instead of (or in addition to) drawing conclusions about agent's qualities from the facts base only, we equip the module `e` (see Subsection 6.1), conveying the Ethotic Argument scheme, with the ability to draw conclusion based on yet another argumentation scheme. This can be done by adding a new rule:

```
goodQualities(X):-communicationX.cr(X,high)
```

to the module `ePremises` and removing (or not if we intend to keep this information) all the facts `goodQualities(X)`. In this way, we combine the Ethotic Argument scheme with the whole set of schemes for reasoning about communicative relations (encoded in the `communicationX` module, see Figure 11), in particular, for reasoning about trust with the use of the Reputation and Prudence schemes.

---

[2] Then, through simulations, one could obtain an optimal division of weights for which the system achieves its goal in the best way, e.g., the fastest.

Figure 11. Communicative Relations and Reasoning about Trust.

```
MODULE communicationX :
RELATIONS: cr(literal,literal).
RULES:
 cr(A,high)   :- trust.v(A), perception.v(A),cognitive.v(A),veracity.v(A).
 cr(A,medium):- trust.v(A),perception.v(A)in {true,unknown},
               cognitive.v(A) in {true,unknown,incons}, veracity.v(A).
 cr(A,low):- trust.v(A), perception.v(A)in {true,incons,unknown},
             cognitive.v(A) in {true,incons,unknown},
             veracity.v(A) in {true,incons,unknown}.
END.

MODULE trust :
RELATIONS: v(literal).
RULES:
 v(A):- rep.trust(A), pru.trust(A) in {true,unknown}.
END.
```

Table 7

Selected Argumentation Schemes for Reasoning about Trust.

		Original Scheme	**Adapted**	**Paraconsistent**
Reputation	Scheme	If B has a reputation for being trustworthy, then A may choose to trust B.	B has a reputation for being trustworthy	reputation(Agent, Value) & math.gt(Value,50)
		trust B	trust B	trust(Agent)
	Critical Questions (Assumptions)	Does B have a good reputation? Are we sure that B's reputation has not been manipulated to make it more positive?	B's reputation has not been manipulated	¬manipulated(Value)

		Original Scheme	**Adapted**	**Paraconsistent**
Prudence	Scheme	A may decide to trust B because it is less risky than not trusting B.	Trusting B is is less risky than not trusting B	riskOfTrust(Agent, Val1) & riskOfDistrust(Agent, Val2) & math.lt(Val1,Val2)
		trust B	trust B	trust(Agent)
	Critical Questions (Assumptions)	Is it riskier to not trust B than it is to trust B? Is it possible to accurately estimate the risk in trusting and not trusting B? Is there another individual we could trust where the risk would be lower than trusting B?	The risk in trusting and not trusting B was accurately estimated. There is no other individual we could trust where the risk would be lower than trusting B.	accurate(Val1) & accurate(Val2) -other(X,Agent) & riskOfTrust(Agent, Val1) & riskOfTrust(X,Val2) & math.lt(Val2,Val1)

Figure 12. Argumentation Scheme for Reasoning about Trust from Reputation

```
MODULE repExceptions :
RELATIONS:
 trust(literal).
 manipulated(literal).
RULES:
 -trust(A) :- repPremisses.reputation(A,V), manipulated(V).
  manipulated(V):- math.gt(V, 100).
END.

MODULE repPremisses :
RELATIONS:
 trust(literal).
 reputation(literal, literal).
RULES:
 trust(A) :- reputation(A, V), math.gt(V,50).
FACTS:
 reputation(bob, 51).
 reputation(alice, 110).
END.
```

Figure 13. Argumentation Scheme for Reasoning about Trust from Prudence

```
MODULE pruExceptions :
RELATIONS:
 trust(literal).
 accurate(literal).
RULES:
-trust(A):- pruPremises.riskOfTrust(A,V1),-accurate(V1) |
            pruPremises.riskOfDistrust(A,V2),-accurate(V2) |
            pruPremises.riskOfTrust(X,Va1),pruPremises.riskOfTrust(A,Va2),
            math.gt(Va1,Va2),base.isOther(X,A).
END.

MODULE pruPremises :
RELATIONS:
 trust(literal).
 riskOfTrust(literal, literal).
 riskOfDistrust(literal, literal).
 isTerrorist(literal).
RULES:
 trust(A):- riskOfTrust(A, V1), riskOfDistrust(A, V2), -math.gt(V1,V2).
 riskOfDistrust(A, 1000):- isTerrorist(A).
FACTS:
 isTerrorist(bob).
 riskOfTrust(bob, 50).
 riskOfDistrust(alice, 1200).
END.
```

8. Discussion and Conclusions

The contribution of this paper is a computationally-friendly framework for paraconsistent argumentation schemes, extending agent's reasoning capabilities. The tetravalent model of argumentation schemes can be used in information-rich environments, naturally obeying incomplete or uncertain information. To this end, we provided templates for implementing an arbitrary argumentation scheme in 4QL, like Position to Know, Ethos, Perception or Expert Opinion [56]. Further, we showed a way of embedding argumentation schemes, on the example of Argument from Reputation and Argument from Prudence to reason about trust. Our solution is both expressive (as 4QL captures all tractable queries) and feasible (as 4QL enjoys polynomial computational complexity of computing queries). Such a choice allows for:

- an efficient belief revision upon filling the gaps in knowledge (see also [4]), using light-weight forms of nonmonotonic reasoning [19]),
- exploring different disambiguation strategies for dealing with inconsistencies [21]. This aspect will be investigated in detail in the upcoming paper.

Such features distinguish 4QL from other formalisms, e.g., Answer Set Programming (ASP) [24]. ASP is based on the trivalent semantics ($true$, $false$, $unknown$), and does not admit inconsistency. Computing a so-called "answer set" (stable model) is NP-complete. The answer sets may contain conclusions that are not grounded in facts, which may be suitable for ASP primary applications (specification and computation of problems from the NP class), however is not appropriate in our case.

Dealing with missing or ambiguous information in argumentation is not a new subject [5]. For example, in [6] the authors propose a formal bi-party inquiry dialogue system where DeLP is used to deal with ignorance and inconsistency. In [47], the authors proposed a logic of multiple-valued argumentation (LMA), in which agents can argue using multi-valued knowledge base in the extended annotated logic programming (EALP) (such an approach was next applied in [32]). However, unlike our approach, the solution was based on Belnap's logic.

9. Acknowledgements

The authors would like to thank Professor Andrzej Szałas for his comments which greatly improved this paper.

References

[1] A. R. Anderson, N. D. Belnap, and J. M. Dunn. *Entailment: The Logic of Relevance and Necessity*, volume 2. Princeton University Press, 1992.

[2] J. L. Austin. *How to Do Things with Words*. Clarendon Press, Oxford, second edition, 1975.

[3] N. Belnap. A useful four-valued logic. In *Modern Uses of Many Valued Logic*, pages 8–37. Reidel, 1977.

[4] L. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[5] P. Besnard and A. Hunter. *Elements of argumentation*, volume 47. MIT press Cambridge, 2008.

[6] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[7] H. A. Blair and V. S. Subrahmanian. Paraconsistent logic programming. *Theoretical Computer Science*, 68(2):135–154, 1989.

[8] P. G. C. Alchourrón and D. Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50:510–530, 1985.

[9] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[10] N. C. A. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 10 1974.

[11] C. V. Damásio and L. M. Pereira. A survey of paraconsistent semantics for logic programs. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 241–320. Kluwer Academic Publishers, 1998.

[12] S. de Amo and M. Pais. A paraconsistent logic approach for querying inconsistent databases. *International Journal of Approximate Reasoning*, 46:366–386, 2007.

[13] F. Dignum, B. Dunin-Kęplicz, and R. Verbrugge. Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9:145–158, 2001.

[14] F. Dignum and M. Greaves, editors. *Issues in Agent Communication*, volume 1916 of *LNCS*. Springer, 2000.

[15] D. Dubois. On ignorance and contradiction considered as truth-values. *Logic Journal of the IGPL*, 16(2):195–216, 2008.

[16] B. Dunin-Kęplicz and A. Strachocka. Perceiving rules under incomplete and inconsistent information. In *Computational Logic in Multi-Agent Systems*, volume LNCS 8143 of *LNCS*, pages 256–272. Springer, 2013.

[17] B. Dunin-Kęplicz and A. Strachocka. Computationally-friendly argumentation schemes. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 167–174, August 2014.

[18] B. Dunin-Kęplicz, A. Strachocka, A. Szałas, and R. Verbrugge. Paraconsistent semantics of speech acts. *Neurocomputing*, 151:943–952, 2015.

[19] B. Dunin-Kęplicz and A. Szałas. Distributed paraconsistent belief fusion. In *Intelligent Distributed Computing VI*, volume 446 of *Studies in Computational Intelligence*, pages 59–69. Springer, 2013.

[20] B. Dunin-Kęplicz and A. Szałas. Taming complex beliefs. *Transactions on Computational Collective Intelligence XI*, LNCS 8065:1–21, 2013.

[21] B. Dunin-Kęplicz, A. Szałas, and R. Verbrugge. Tractable reasoning about group beliefs. In *Engineering Multi-Agent Systems*, volume 8758 of *LNCS*, pages 328–350. Springer International Publishing, 2014.

[22] B. Dunin-Keplicz and A. Strachocka. Paraconsistent multi-party persuasion in talklog. In Q. Chen, P. Torroni, S. Villata, J. Y. Hsu, and A. Omicini, editors, *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, volume 9387 of *Lecture Notes in Computer Science*, pages 265–283. Springer, 2015.

[23] B. Dunin-Keplicz and A. Strachocka. Tractable inquiry in information-rich environments. In Q. Yang and M. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 53–60. AAAI Press, 2015.

[24] M. Gelfond and Y. Kahl. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents - The Answer-Set Programming Approach*. Cambridge University Press, 2014.

[25] M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. In *Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, ATAL '98, pages 1–10, London, UK, 1999. Springer-Verlag.

[26] W. C. J.-Y. Beziau and D. Gabbay. *Handbook of Paraconsistency*. College Publications, 2007.

[27] S. Jaśkowski. Propositional calculus for contradictory deductive systems. *Studia Logica*, 24(1):143–157, 1969.

[28] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, July 1990.

[29] J. Małuszyński and A. Szałas. Living with inconsistency and taming nonmonotonicity. In *Datalog Reloaded*, volume 6702 of *LNCS*, pages 384–398. Springer-Verlag, 2011.

[30] J. Małuszyński and A. Szałas. Logical foundations and complexity of 4QL, a query language with unrestricted negation. *Journal of Applied Non-Classical Logics*, 21(2):211–232, 2011.

[31] J. Małuszyński and A. Szałas. Partiality and inconsistency in agents' belief bases. In *KES-AMSTA*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 3–17. IOS Press, 2013.

[32] K. Matsunaga and H. Sawamura. Aaslma: An automated argument system based on logic of multiple-valued argumentation. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3684 of *LNCS*, pages 830–838. Springer, 2005.

[33] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[34] R. Moore. Possible-world semantics for autoepistemic logic. In *Readings in Nonmonotonic Reasoning*, pages 137–142. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[35] D. Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 3)*, pages 353–395. Oxford University Press, Inc., New York, NY, USA, 1994.

[36] S. Parsons, K. Atkinson, K. Z. Haigh, K. N. Levitt, P. McBurney, J. Rowe, M. P. Singh, and E. Sklar. Argument schemes for reasoning about trust. In B. Verheij, S. Szeider, and S. Woltran, editors, *COMMA*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 430–441. IOS Press, 2012.

[37] H. Prakken. On the nature of argument schemes. In *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning*, pages 167–185. College Publications, 2010.

[38] G. Priest. The logic of paradox. *Journal of Philosophical Logic*, 8(1):219–241, 1979.

[39] G. Priest. Paraconsistent belief revision. *Theoria*, 67(3):214–228, 2001.

[40] G. Priest, K. Tanaka, and Z. Weber. Paraconsistent logic. In *The Stanford Encyclopedia of Philosophy*. Spring 2015 edition, 2015.

[41] R. Reiter. A logic for default reasoning. In *Readings in Nonmonotonic Reasoning*, pages 68–93. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[42] R. Reiter. On closed world data bases. In *Readings in Nonmonotonic Reasoning*, pages 300–310. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[43] P. K. Schotch and R. E. Jennings. Inference and necessity. *Journal of Philosophical Logic*, 9(3):327–340, 1980.

[44] J. Searle and D. Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, 1985.

[45] M. P. Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, Dec. 1998.

[46] A. Szałas. How an agent might think. *Logic Journal of the IGPL*, 21(3):515–535, 2013.

[47] T. Takahashi and H. Sawamura. A logic of multiple-valued argumentation. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 800–807. IEEE Computer Society, 2004.

[48] K. Tanaka. The AGM theory and inconsistent belief change. *Logique Et Analyse*, 48(189-192):113–150, 2005.

[49] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[50] F. van Harmelen, V. Lifschitz, and B. W. Porter, editors. *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*. Elsevier, 2008.

[51] B. van Linder, W. van der Hoek, and J.-C. Meyer. Formalizing abilities and opportunities of agents. *Fundamenta Informaticae*, 34(1-2):53–101, 1998.

[52] B. van Linder, W. van der Hoek, and J.-J. Meyer. Actions that make you change your mind. In I. Wachsmuth, C.-R. Rollinger, and W. Brauer, editors, *KI-95: Advances in Artificial Intelligence*, volume 981 of *Lecture Notes in Computer Science*, pages 185–196. Springer Berlin Heidelberg, 1995.

[53] A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.

[54] D. Walton. Ethotic arguments and fallacies: The credibility function in multi-agent dialogue systems. *Pragmatics and Cognition*, 7(1):177–203, 1999.

[55] D. Walton and E. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.

[56] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

Tractable Inquiry in Information-Rich Environments*

Barbara Dunin-Kęplicz, Alina Strachocka
Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
keplicz,astrachocka@mimuw.edu.pl

Abstract

In the contemporary autonomous systems the role of complex interactions such as (possibly relaxed) dialogues is increasing significantly. In this paper we provide a paraconsistent and paracomplete implementation of **inquiry dialogue** under realistic assumptions regarding availability and quality of information. Various strategies for dealing with unsure and inconsistent information are analyzed. The corresponding dialogue outcomes are further evaluated against the (paraconsistent and paracomplete) distributed beliefs of the group.

A specific 4-valued logic underpins the presented framework. Thanks to the qualities of the implementation tool: a rule-based query language 4QL, our solution is both expressive and tractable.

1 Paraconsistent Nonmonotonic Dialogues

The synergistic effect of collaborating agents is achieved by their proper communication. However, in dynamic and unpredictable environments up-to-date, sure and complete information is hardly obtainable. This leads to conflicts, uncertainty and paracompleteness, particularly when handling information originating from multiple sources of diverse credibility, quality or significance. In this paper we introduce a new approach to logical modeling of conversing agents, which are prepared to handle inconsistency and ignorance.

To this end, a paracomplete and paraconsistent (i.e., tolerating inconsistencies) logic is necessary, supported by two new truth values: *unknown* ($\mathfrak{u}$) and *inconsistent* ($\mathfrak{i}$). In line with other paraconsistent approaches to modeling dialogues [Takahashi and Sawamura, 2004; Prakken, 2010; Black and Hunter, 2009], inconsistency does not trivialize reasoning but is treated as first-class citizen alike *true* ($\mathfrak{t}$) and *false* ($\mathfrak{f}$). In our system the following choices have been made.

- The four-valued logic of [Vitória *et al.*, 2009] underpins the solution.
- Unknown or inconsistent conclusions do not enforce termination of the reasoning process.
- Such conclusions can be handled via lightweight forms of nonmonotonic reasoning.

Entailment in logic amounts to deriving conclusions from theories that can be seen as complex knowledge bases. However, instead of querying arbitrary theories, in order to reduce complexity we tailor them to their tractable versions like specific rule-based systems. Thus, instead of reasoning in logical systems of high complexity, we query paraconsistent knowledge bases. Only recently has a sufficiently expressive tool existed for creating and querying them in polynomial time: 4QL - a DATALOG¬¬-like four-valued rule-based query language. Following this shift in methodology, the contribution of this paper is an implementation of a tractable, paraconsistent and paracomplete multi-party **inquiry dialogue** suitable for agents situated in information-rich environments. The overall goal of inquiry is to collectively solve a theoretic problem, resulting in the common knowledge about the solution.

Consider a multi-agent system where each *swarm agent* is specialized in gathering different type of information via a polling system, and a *supervisor agent* which verifies certain information for the human user. Suppose the human user asks if it is safe to travel to place X ($safe(X)$?) and none of the agents knows the answer. Engaging in inquiry on the topic $safe(X)$ allows agents to share only the relevant pieces of their (possibly vast) knowledge and collectively arrive at a final recommendation for the human user. Although conflicts may naturally appear on many different levels of such group activity [Dunin-Kęplicz *et al.*, 2014], it is not the goal of inquiry but rather persuasion to resolve them.

Unlike the classical case [Walton and Krabbe, 1995], our approach to inquiry permits 4-valued statements. It turns out that the initial valuation of the topic separates *Inquiry-What* from *Inquiry-That*. In both cases, several strategies to handle missing and inconsistent information are presented and formally investigated. The final outcomes of such dialogues are compared against the (possibly inconsistent and incomplete) distributed knowledge of the conversing group [Fagin *et al.*, 1995]. In this regard the *soundness* of a strategy means that whenever a dialogue terminates with a given conclusion, the same result would be obtained by an individual reasoning from the union of all the agents' belief bases. Accordingly, if a solution is obtainable from the union of agents beliefs, an inquiry under a *complete* strategy will reach it. The main result of this research concerns soundness and completeness

---

*This research is partially supported by Warsaw Center of Mathematics and Computer Science.

of the *open-minded* inquiry strategy (Theorem 4).

Enriching the modeling perspective allows us to contemplate several new cognitive situations in communication (see e.g., [Dunin-Keplicz *et al.*, 2015]), occurring also in inquiry. Arguably, other normative models of dialogues would benefit from the 4-valued approach.

The paper is structured as follows. First, in Section 2, the notions underpinning our solution are recalled. Section 3 concerns the formalization of inquiry, its strategies and properties. Finally, Section 4 concludes the paper.

2 Language and Implementation Tool

Our formal inquiry dialogue system uses the logical language of [Małuszyński and Szałas, 2013; Szałas, 2013; Małuszyński and Szałas, 2011]. Agents' informational stance is encoded in the rule-based query language 4QL[1] defined in [Małuszyński and Szałas, 2013], further developed in [Szałas, 2013] and based on a 4-valued logic of [Vitória *et al.*, 2009]. 4QL allows for negation both in premisses and conclusions of rules. Importantly, negation in the conclusions may lead to inconsistencies. Even though openness of the world is assumed, rules can be used to close the world locally or globally. Below, the formal language underlying 4QL will be briefly introduced.

In what follows all sets are finite except for sets of formulas. We deal with the classical first-order language over a given vocabulary without function symbols. We assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols.

Definition 1 A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, $\bar{\tau}$ being a sequence of parameters, $\bar{\tau} \in (\mathit{Const} \cup Var)^k$, where k is the arity of $R \in \mathit{Rel}$. *Ground literals over Const*, denoted by $\mathcal{G}(\mathit{Const})$, are literals without variables, with all constants in *Const*. If $\ell = \neg R(\bar{\tau})$ then $\neg \ell \overset{\text{def}}{=} R(\bar{\tau})$.$\lhd$

Though we use classical first-order syntax, the semantics substantially differs from the classical one as truth values $\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}$ (true, inconsistent, unknown, false) are explicitly present; the semantics is based on sets of ground literals rather than on relational structures. Intuitively:

- a is $\mathfrak{t}$ if all sources claim a,
- a is $\mathfrak{f}$ if all sources claim $\neg a$,
- a is $\mathfrak{u}$ if no sources claim a nor $\neg a$,
- a is $\mathfrak{i}$ if some sources claim a, other claim $\neg a$.

For semantics of propositional connectives see Table 1. The definitions of $\wedge$ and $\vee$ reflect minimum and maximum with respect to the truth ordering

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}. \tag{1}$$

Whenever truth values are restricted to $\{\mathfrak{f}, \mathfrak{t}\}$, the semantics is compatible with the semantics of classical first-order logic.

Let $v : \mathit{Var} \rightarrow \mathit{Const}$ be a *valuation of variables*. For a literal ℓ, by $\ell(v)$ we mean the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$.

Table 1: Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$.

$\wedge$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\vee$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\rightarrow$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$		$\neg$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{f}$

Definition 2 The *truth value* $\ell(L, v)$ of a literal ℓ w.r.t. a set of ground literals L and valuation v, is defined by:

$$\ell(L,v) \overset{\text{def}}{=} \begin{cases} \mathfrak{t} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} & \text{if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} & \text{if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$

$\lhd$

For a formula $\alpha(x)$ with a free variable x and $c \in \mathit{Const}$, by $\alpha(x)^x_c$ we understand the formula obtained from α by substituting all free occurrences of x by c. Definition 2 is extended to all formulas in Table 2, where α denotes a first-order formula, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 1.

Table 2: Semantics of first-order formulas.

- if α is a literal then $\alpha(L, v)$ is defined in Definition 2;
- $(\neg\alpha)(L, v) \overset{\text{def}}{=} \neg(\alpha(L, v))$;
- $(\alpha \circ \beta)(L, v) \overset{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x \alpha(x))(L, v) = \min\limits_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$, where min is the minimum w.r.t. ordering (1);
- $(\exists x \alpha(x))(L, v) = \max\limits_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$, where max is the maximum w.r.t. ordering (1).

In 4QL beliefs are distributed among *modules*. Each module can be treated as a finite set of literals. For specifying rules, multisource formulas and querying modules, we apply the language of [Szałas, 2013].

Definition 3 A *multisource formula* is an expression of the form: $m.A$ or $m.A \in T$, where:

- m is a module name;
- A is a first-order or a multisource formula;
- v is a valuation;
- $T \subseteq \{\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}\}$.

We write $m.A = v$ (respectively, $m.A \neq v$) to stand for $m.A \in \{v\}$ (respectively, $m.A \notin \{v\}$). $\lhd$

---

[1] Open-source implementation of 4QL is available at `4ql.org`.

The intuitive meaning of a multisource formula $m.A$ is:

> "return the answer to query expressed by formula A, computed within the context of module m".

The value of '$m.A \in T$ ' is:

$$\begin{cases} \mathsf{t} & \text{when the truth value of } A \text{ in } m \text{ is in the set } T; \\ \mathsf{f} & otherwise. \end{cases}$$

Let $A(X_1, \ldots, X_k)$ be a multisource formula with $X_1, \ldots, X_k$ being its all free variables and D be a finite set of literals (a belief base). Then A, understood as a query, returns tuples $\langle d_1, \ldots, d_k, tv \rangle$, where $d_1, \ldots, d_k$ are database domain elements and the value of $A(d_1, \ldots, d_k)$ in D is tv.

From now on we assume that the domain and language are fixed and the programs and rules are ground. If S is a set, then $\text{FIN}(S)$ represents the set of all finite subsets of S. In what follows let $\mathbb{C} \stackrel{\text{def}}{=} \text{FIN}(\mathcal{G}(\mathit{Const}))$ be the set of all finite sets of ground literals over constants in *Const*.

Definition 4

- *Rules* are expressions of the form:

  $$\ell :\!-\, b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}. \tag{2}$$

  where the *conclusion* ℓ is a positive or negative literal and the *premisses* $b_{11}, \ldots, b_{1i_1}, \ldots, b_{m1}, \ldots, b_{mi_m}$ are multisource formulas and ',' and '|' abbreviate conjunction and disjunction, respectively.
- A *fact* is a rule with empty premisses (evaluated to t).
- A *module* is a syntactic entity encapsulating a finite number of facts and rules.
- A 4QL *program* is a set of modules, without cyclic references to modules involving multisource formulas of the form $m.A \in T$. $\lhd$

If δ is a rule, by $head(\delta)$ we mean the rule conclusion. If δ is a fact, $head(\delta) = \delta$.

The key concepts of modules and multisource formulas allow us to deal with unknown or inconsistent conclusions without enforcing termination of the reasoning process. Technically, any literal l corresponds to a multisource formula $M.l$, thus $l \in M$.

The semantics of 4QL is defined by *well-supported models* [Małuszyński and Szałas, 2013; Szałas, 2013], i.e., models consisting of (positive or negative) ground literals, where each literal is a conclusion of a derivation starting from facts. For any set of rules, such a model is uniquely determined and computable in deterministic polynomial time $O(N^k)$ where N is the size of domain and $k = max(s, t)$ where s is the maximal arity of relations and t is the maximum number of free variables. As we deal with ground programs, $t = 0$. When s is a bound constant, which is the case in practical applications of 4QL (qualitative not quantitative reasoning), we achieve tractability. Notice that it is the same complexity as SQL with recursion.

Definition 5 Let P be a 4QL program, A a formula, and $\mathcal{M}_P$ the well-supported (unique) model of P. Then: $P \models A$ iff for any valuation v we have $\mathcal{M}_P \models v(A)$.

As an example, consider program $P = \{top, su\}$ consisting of two modules top and su (for surveillance).

$$\begin{array}{rrcl}
top = \{ & enter(b) & :\!- & isAt(s,b), \neg has(s,h).,\\
 & isAt(s,b) & :\!- & isArmed(s), hearShotsAt(b).,\\
 & isAt(s,b) & :\!- & su.isAt(s,b) \in \{\mathsf{u}, \mathsf{i}, \mathsf{t}\}.,\\
 & & & \neg has(s,h),\\
 & & & has(s,h),\\
 & & & isArmed(s)\}\\
su = \{ & isAt(s,b) & :\!- & see(s,b), \neg conditions(fog).,\\
 & & & see(s,b),\\
 & & & \neg conditions(fog)\}
\end{array} \tag{3}$$

The literals s, b, h represent $suspect, building$ and $hostage$, respectively. The program uniquely determines the following well-supported model for module su:

$$\mathcal{M}_{su} = \{\neg conditions(fog), see(s,b), isAt(s,b)\} \tag{4}$$

and the following well-supported model for module *top*:

$$\begin{array}{l} \mathcal{M}_{top} = \{enter(b), \neg enter(b), isAt(s,b), \\ isArmed(s), has(s,h), \neg has(s,h)\}. \end{array} \tag{5}$$

Definition 6 Let ℓ be a literal and P a 4QL program. A *derivation* of ℓ from P is the well-supported model $\mathcal{M}_P$.

A dependence set of a literal ℓ from a program P consists of literals reachable via backward chaining on P from ℓ.

Definition 7 Let ℓ be a literal and P a 4QL program. A *dependence set* of ℓ from P, denoted $\mathcal{D}_{P,\ell}$ is a set of literals such that:

- $\neg\ell, \ell \in \mathcal{D}_{P,\ell}$,
- if there is a rule $\ell' :\!-\, b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$ in P, such that $\ell' \in \mathcal{D}_{P,\ell}$ then $\forall_{j \in 1..m} \forall_{k \in 1..i_j}\ b_{jk}, \neg b_{jk} \in \mathcal{D}_{P,\ell}$. $\lhd$

A proof of a literal ℓ from a program P is a subprogram S of P generated from the dependence set $\mathcal{D}_{P,\ell}$ by taking all rules and facts of P whose conclusions are in $\mathcal{D}_{P,\ell}$.

Definition 8 Let ℓ be a literal, P a 4QL program. A *proof* of l from P is a 4QL program $S \subseteq P$ such that $\delta \in S$ iff $head(\delta) \in \mathcal{D}_{P,\ell}$, where δ is a fact or a rule. The *size of the proof S* is the size of the program S. The *size of domain of the proof S* is the size of the dependence set $\mathcal{D}_{P,\ell}$. $\lhd$

In order to implement dialogues, the functionality of adding a rule to a 4QL program is required.

Definition 9 We define an operation of *adding a ground rule* $M_i.\ell :\!-\, b$ to a 4QL program $P = \{M_1, ..., M_n\}$ as follows:
$P' = P \cup \{M_i.\ell :\!-\, b\} =$
$\{M_1, ..., M_{i-1}, M_i \cup \ell :\!-\, b, M_{i+1}, ..., M_n\}$

3 Inquiry

The purpose of inquiry is to collectively solve a theoretical problem [Walton and Krabbe, 1995]. In multi-agent systems, inquiry *"starts when some agents are ignorant about the solution to some question or open problem. The main goal is the growth of knowledge, leading to agreement about the conclusive answer of the question. This goal may be attained* *in many different ways, including an incremental process of argument which builds on established facts in drawing conclusions beyond a reasonable doubt. Both information retrieval and reasoning may be intensively used in this process"* [Dunin-Kęplicz and Verbrugge, 2010]. In its classical form, inquiry seeks to prove a statement as true or false:

I1. $is(suspect, guilty) = \mathfrak{t}$: 'prove that suspect is guilty',

I2. $is(suspect, guilty) = \mathfrak{f}$: 'prove that suspect is not guilty'.

Our paraconsistent and paracomplete framework allows for contemplating other possibilities:

I3. $is(suspect, guilty) = \mathfrak{i}$: 'prove that there are inconsistent information concerning suspect's guilt',

I4. $is(suspect, guilty) = \mathfrak{u}$: 'is suspect guilty?'.

Specifically (3) exemplifies that inquiry can commence when the goal's initial valuation is $\mathfrak{i}$. Our approach allows to model such situations, common in practical applications. In contrast, until a valuation for (4) is established, no classical inquiry on this subject (finding a proof) can commence. This scenario resembles discovery dialogue, where *"we want to discover something not previously known"* [McBurney and Parsons, 2001]. In our setting, the dialogue aiming at discovering the value of a statement is just another variation of inquiry, so is structured exactly the same. Thus, two types of inquiry dialogues are distinguished:

1. **Inquiry-WHAT**, where initial valuation of s is $\mathfrak{u}$ and the goal of the dialogue is to establish the valuation v_f of s (see **I4**).
2. **Inquiry-THAT**, where initial valuation of s is $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$, and the goal of the dialogue is to confirm or refute this by providing the proof for s (see **I1**, **I2**, **I3**).

Inquiry-WHAT succeeds if the final valuation $v_f \neq \mathfrak{u}$ and Inquiry-THAT succeeds if v_f is equal to the initial valuation v_i of s. An outcome of a successful inquiry is the valuation of the goal and the proof of it (see Definiton 13).

3.1 Query and Commitment Stores

A common approach to modeling inquiry is to keep two stores (see e.g., [Black and Hunter, 2009; Singh, 1998] and references therein): a *Commitment Store* (CS), reflecting *agents' commitments*, and a *Query Store* (QS), reflecting *current open questions*. Commitment Store is thus associated with an individual agent while Query Store - with the dialogue.

We maintain both stores associated with the dialogue (CS_d, QS_d) without any assumptions about agents' individual Commitment Stores. Technically, our Commitment Store reflects the current accumulated knowledge base. It is created empty when the dialogue begins (as no locutions have been uttered yet) and updated with every assertion relevant to the dialogue. To sum up, in our methodology, the inquiry Commitment Store is just an evolving 4QL program (see also [Alferes *et al.*, 2002]).

We assume that in the course of dialogue agents assert only relevant information, that is, rules whose conclusions match the current entries in QS. For a literal $l \in QS$, relevant responses include both l :– b and $\neg l$:– b (or ℓ and $\neg\ell$). Accordingly, two locutions crucial to inquiry are:

- $assert(S_i, \delta, d)$: agent S_i asserts a rule or a fact δ in the dialogue d. If assertion is relevant, it's content is added to CS_d and its premisses are added to QS_d.
- $requestAll(S_i, d)$: agent S_i requests content of QS_d.

Backward chaining mechanism employed here is commonly used in deductive argumentation for driving the argumentation process (see e.g., [Besnard and Hunter, 2008; Black and Hunter, 2009; Prakken, 2010]).

Definition 10 Locution m^t is *relevant* to an inquiry d at time t iff $m^t = assert(S, M_i.\ell :\!-\, b, d)$ and $(\neg)M_i.\ell \in QS^t_d$, where QS^t_d is the Query Store of d at time t. We will alternate between the notions of locution, message, move and utterance. $\triangleleft$

To make the communication more flexible, the assumption about relevance of the locutions can be realized by a filtering mechanism. Then, instead of requiring that agents make specific moves, we allow them to utter any locutions, filtering out the irrelevant ones[2].

Definition 11 *Commitment Store* of a dialogue d at time t is a 4QL program denoted as $CS^t_d = \langle M^t_1, \ldots, M^t_k \rangle$:

- $CS^0_d = \emptyset$
- $CS^t_d = CS^{t-1}_d \cup \{M_i.\ell :\!-\, b\}$, such that $m^t = assert(S, M_i.\ell :\!-\, b, d)$ is relevant to d at time t,
- $CS^t_d = CS^{t-1}_d$ otherwise. $\triangleleft$

Next, the Query Store, is a repository of active, unresolved leads in the inquiry. It contains literals which compose the proof of the inquiry goal s. At the beginning the Query Store contains s as a single entry. The mechanism of updating QS is in fact a paraconsistent and paracomplete distributed version of backward chaining[3], as discussed in Section 3.3. However, in contrast to the classical backward chaining, here we have a number of additional options to investigate. Consequently, there may be various policies for adding literals to QS (selecting threads to follow) and removing them from QS (closing explored threads). Functions $open$ and $close$ (see Definition 12) correspond to such methods.

Definition 12 Let:

- CS^t_d be the Commitment Store of dialogue d at time t,
- m^t be the message received at time t,
- *close* : $\textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \rightarrow \textsc{Fin}(\mathbb{C})$ be a method for removing entries from the Query Store,
- *open* : $\textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \rightarrow \textsc{Fin}(\mathbb{C})$ be a method for adding entries to Query Store.

Then, *Query Store* of an inquiry dialogue d on subject s at time t is a finite set of literals denoted as QS^t_d such that:

- $QS^0_d = \{s\}$

---

[2] Such a filter is easy to implement: upon receiving a message, QS is inspected to verify if the rule head is in the scope of inquiry.

[3] Hybrid backward-forward chaining techniques may be used if $assert$ locution contains a set of rules, e.g., a subset of proof constructed bottom-up. This is a topic for future research.

- $QS^t_d = (QS^{t-1}_d \cup B') \setminus B''$, if $m^t = assert(S, M_i.\ell :\!- b, d)$, where
  $$B' = open(b, CS^t_d),$$
  $$B'' = close(QS^{t-1}_d \cup B', CS^t_d),$$
- $QS^t_d = QS^{t-1}_d$ otherwise. $\lhd$

3.2 Dialogue Outcome vs. Distributed Knowledge

Our setting consists of a finite set of n cooperative agents. The assumption that agents do not withhold information implicitly constraints the number of *requestAll* locutions per one assertion. Agents' belief bases are encoded as finite, ground 4QL programs $P_1, \ldots, P_n$, that share a common ontology and do not change during the course of dialogue. Agents communicate one-to-all without coordination. The well-supported models $\mathcal{M}_{P_1}, \ldots, \mathcal{M}_{P_n}$ of the programs express agents' final beliefs. The union of individual agents' belief bases (i.e., their distributed knowledge [Fagin *et al.*, 1995]) is expressed by the sum of their 4QL programs: $\bigcup_{i \in 1..n} P_i$. An agent can join and leave a dialogue at any time if in between $join$ and $leave$ locutions it utters at least one $assert$ion. Agents cannot repeat assertions. These assumptions allow us to verify quality and completeness of the obtained results.

Since 4QL programs are finite and agents cannot repeat utterances, there must be a moment t when no agent has anything more to utter because either it has run out of relevant moves or because the dialogue goal s has been achieved, whichever comes first. Thus, dialogue terminates at time t. The knowledge accumulated in the course of a dialogue d is expressed by the Commitment Store of that dialogue at termination time t: CS^t_d. The final conclusion depends on the dialogue strategy (see below) and is expressed as follows.

Definition 13 For an inquiry terminating at time t, with the goal s of initial valuation v_i, the value of the dialogue conclusion is $v_f = v(s, \mathcal{M}_{CS^t_d})$, where $\mathcal{M}_{CS^t_d}$ is the well-supported model of CS^t_d. Dialogue is:

- successful iff
  - $v_i = \mathfrak{u} \wedge v_f \neq \mathfrak{u}$ [Inquiry-WHAT], or
  - $v_i \neq \mathfrak{u} \wedge v_f = v_i$ [Inquiry-THAT],
- unsuccessful otherwise. $\lhd$

The value of the goal s obtained from the union of agents' programs is expressed as $v(s, \mathcal{M}_{\bigcup_{i \in 1..n} P_i})$.

Definition 14 Let:

- $open : \textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \rightarrow \textsc{Fin}(\mathbb{C})$,
- $close : \textsc{Fin}(\mathbb{C}) \times \textsc{Fin}(\mathbb{C}) \rightarrow \textsc{Fin}(\mathbb{C})$

be two methods for adding and removing entries to Query Store of dialogue d. Then: $ST = \langle open, close \rangle$ is a *strategy*.

Definition 15 A strategy ST is *sound* iff whenever dialogue d on subject s conducted under this strategy terminates at t with conclusion k, then if $v(s, \mathcal{M}_{CS^t_d}) = k$ then $v(s, \mathcal{M}_{\bigcup_{i \in 1..n} P_i}) = k$.

Definition 16 A strategy ST is *complete* iff whenever dialogue d on subject s conducted under this strategy terminates at t with conclusion k, then if $v(s, \mathcal{M}_{\bigcup_{i \in 1..n} P_i}) = k$ then $v(s, \mathcal{M}_{CS^t_d}) = k$.

3.3 Opening and Closing Inquiry Threads

In classical backward chaining, the inference engine selects rules whose consequents match the goal to be proved. If the antecedent of the rule is not known to be true, then it is added to the list of goals. In our paraconsistent and nonmonotonic distributed version of backward chaining, the conditions under which antecedent can be added to the list of goals differ depending on the method used. Consequently, there may be various policies for adding literals to QS (selecting threads to follow via function *open*). From a variety of possibilities, here we investigate two such methods. A literal can be added to the Query Store if:

A1. **Its valuation in the CS model is $\mathfrak{u}$**, meaning that only threads lacking any evidence whatsoever are explored.

A2. **Always**, meaning that *every* premise is investigated further, even one that is tentatively assumed to be $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$.

Definition 17 Let CS^t_d be the Commitment Store of an inquiry dialogue d at time t and $\mathcal{M}_{CS^t_d}$ be its well-supported model. Let $m^t = assert(S, M_i.\ell :\!- b, d)$ be the message received at time t, such that: $b = b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$. Then,

$$open(b, CS^t_d) \overset{\text{def}}{=} \begin{cases} \{b_{j_k} | j \in 1..m, k \in 1..i_j & \\ \text{ and } \mathcal{M}_{CS^t_d}(b_{j_k}) = \mathfrak{u}\} & [A1] \\ \{b_{j_k} | j \in 1..m, k \in 1..i_j\} & [A2] \end{cases}$$

$\lhd$

Notice that in the nonmonotonic paraconsistent backward-chaining, obtaining a truth value for p does not necessarily close the line of reasoning about p, since the evidence put forward by other agents may change the value of p in a number of ways. This is why we conduct inquiry until all relevant information is shared by the agents.

The conditions under which a goal can be abandoned also differ depending on the policy employed. We distinguish two methods for removing literals from QS (closing explored threads via function *close*):

R1. **Once its valuation in the CS model is not $\mathfrak{u}$,** meaning that a thread is terminated whenever any evidence for it is found. In some cases it may be closed prematurely, without exposing other evidence relevant to the thread.

R2. **Never**, meaning the threads are never abandoned, as the information regarding them may grow. This will not lead to infinite dialogues, since agents cannot repeat utterances and their programs do not change during dialogue.

Definition 18 Let CS^t_d be the Commitment Store of an inquiry dialogue d at time t and $\mathcal{M}_{CS^t_d}$ be its well-supported model. Let QS^{t-1}_d be the Query Store of an inquiry dialogue d at time $t-1$ and $\mathcal{M}_{QS^{t-1}_d}$ be its well-supported model. Then,

$$close(QS^{t-1}_d, CS^t_d) \overset{\text{def}}{=} \begin{cases} \{x \in \mathcal{M}_{QS^{t-1}_d} | \mathcal{M}_{CS^t_d}(x) \neq \mathfrak{u}\} & [R1] \\ \emptyset & [R2] \end{cases}$$

3.4 Inquiry Strategies

The ensuing question is which combination of methods for updating QS makes sense (see Table 3) and how do resulting inquiry strategies differ. Unlike other approaches, we do not assume that the distributed knowledge of the group is complete. If the statement s cannot be proved by agents, the conclusion would simply be $\mathfrak{u}$.

Table 3: Inquiry strategies defined as pairs of methods for updating QS.

	R1	**R2**
A1	narrow-minded	pragmatic
A2	forgetful	open-minded

Theorem 1 *Narrow-minded strategy is neither sound nor complete. Moreover, it is type 1 nondeterministic.*[4]

Proof. Due to the non-monotonicity of our inquiry, applying the narrow-minded strategy may result in overlooking some important information. As the counterexample, assume three agents A_1, A_2, A_3 are engaged in an inquiry dialogue with the goal $enter(b)$. Their programs are shown in Table 4 and the dialogue conduct is presented in Table 5.

Table 4: Programs of Agents A_1, A_2, A_3.

	A_1	A_2	A_3
1	$enter(b)$:– $isAt(s,b)$, $\neg has(s,h)$	$\neg su.isAt(s,b)$	$hearShotsAt(b)$
2	$isAt(s,b)$:– $su.isAt(s,b) \in \{\mathfrak{u}, \mathfrak{i}, \mathfrak{t}\}$	$isArmed(s)$	
3	$isAt(s,b)$:– $isArmed(s)$, $hearShotsAt(b)$		
4	$\neg has(s,h)$		

Table 5: Example of a Narrow-Minded Inquiry.

t	QS_d^t	m_t	$\mathcal{M}_{CS_d^t}$
0	$enter(b)$	$\emptyset$	$\emptyset$
1	$enter(b), isAt(s,b), has(s,h)$	$A_1(1)$	$\emptyset$
2	$enter(b), has(s,h), su.isAt(s,b)$	$A_1(2)$	$isAt(s,b)$
3	$enter(b), has(s,h)$	A_2 (1)	$\neg su.isAt(s,b)$
4	$enter(b)$	$A_1(4)$	$\neg su.isAt(s,b)$, $\neg has(s,h)$

For brevity, we denote assertions in Table 5 as $A_j(k)$, standing for the k-th rule of agent A_j. Dialogue terminates in step 4, since only agent A_1 has a rule with conclusion $enter(b)$ but it has already uttered it. Notice that at timepoint $t = 2$ we had to remove $isAt(s,b)$ from the Query Store, as it became true in $\mathcal{M}_{CS_d^2}$. Therefore, agent A_1 didn't have a chance to use rule (3) in the dialogue. Obviously, $v(enter(b), \mathcal{M}_{CS_d^4}) = \mathfrak{u}$, whereas the conclusion obtained by merging agents' programs is $v(enter(b), \mathcal{M}_{\bigcup_{i\in 1..3} P_i}) = \mathfrak{t}$. If instead in the timepoint $t = 2$ agent A_1 would have uttered rule (3), then Query Store and in consequence, the whole dialogue, would look differently, leading to a true conclusion even if agent A_1 didn't have a chance to utter rule (1). ◁

Two strategies are equal if the dialogues conducted under these strategies cannot be distinguished on the basis of the content of the stores at any time.

Definition 19 Dialogue D_1 is *equal* to dialogue D_2 iff for all finite sequences of moves $s = m^1, \ldots, m^{t_s}$, s.t. m^i is relevant at i to D_1 and to D_2, we have that $\forall i \in 1..t_s$ $CS^i_{D_1} = CS^i_{D_2}$ and $QS^i_{D_1} = QS^i_{D_2}$.

Strategies S_1 and S_2 are *equal* iff dialogues conducted under these strategies are equal. ◁

Theorem 2 *Forgetful and narrow-minded strategies are equal.*

Proof sketch. In the forgetful strategy, we add *all* literals from the rule body to QS only to remove the *known* ones afterwards. Therefore, what remains are the $unknown$ literals. Since agents cannot query QS in between adding and removing literals (in theory update of QS is atomic operation), these two strategies are indistinguishable. ◁

Theorem 3 *Pragmatic and open-minded strategies are equal in terms of dialogue conduct.*

Proof. Let's consider the pragmatic strategy and a goal s. In the first step, the rule s :– b is considered. All rule premisses (b) are either empty (when s is a fact) or unknown (since CS^0 is empty). Therefore, in the first step all premisses (b) are added to QS^0 and the initial rule s :– b (or fact s) is added to CS^0. Obviously for a literal to be $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$, it has to be a rule conclusion or a fact. Since only rules, whose conclusions are in QS are admitted to CS, there cannot be a $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$ literal which is in CS but was not in QS beforehand. ◁

Theorem 4 *Open-minded strategy is sound and complete.*

Proof sketch. Assume that $v(s, \mathcal{M}_{CS_d^t}) = k$ and $v(s, \mathcal{M}_{\bigcup_{i\in 1..n} P_i}) \neq k$. At the time of dialogue termination, CS contains all relevant messages. Each of these was uttered by at most one agent. Therefore, we can assign each message to a set CS_i where i was the sender. Obviously, $CS_i \subseteq P_i$. Therefore we have: $CS = \bigcup_{i\in 1..n} CS_i \subseteq \bigcup_{i\in 1..n} P_i$. Since $v(s, \mathcal{M}_{CS_d^t}) = k$ and $v(s, \mathcal{M}_{\bigcup_{i\in 1..n} P_i}) \neq k$, that means that there is a part of the union of programs $S \overset{\text{def}}{=} \bigcup_{i\in 1..n} P_i \setminus CS$, such that, adding S to CS would change the valuation of s. However, that would mean that there exists a rule (or a fact) in S whose conclusion is in premisses of CS. That means, that rule is a part of the proof for s but was not uttered by the agent, which contradicts our assumptions.

Proof of completeness is analogous. ◁

Notice that in open-minded inquiry on subject s, CS is the evolving proof of s from $\bigcup_{i\in 1..n} P_i$ and QS is the evolving dependence set of s from $\bigcup_{i\in 1..n} P_i$.

---

[4] Type 1 nondeterminism in logic programs means freedom to choose the rule to apply [Schöning, 2008].

3.5 Complexity

Complexity measures of proposed inquiry strategies include:

- *communication complexity*, concerning only the amount of communication among agents (who have unlimited computational power) [Kushilevitz and Nisan, 1997],
- *computational complexity* (data complexity), concerning the amount of computation (when communication is free) required to:
  - achieve dialogue termination,
  - obtain a conclusion of a terminated dialogue.

Computational complexity of both problems is expressed in terms of *data complexity* [Vardi, 1982; Papadimitriou and Mihalis, 1997], i.e., complexity of evaluating a fixed query (here: inquiry goal) on an arbitrary database (CS). Thus data complexity is given as a function of the size of CS.

In what follows we deal with terminated dialogues and thus we write CS and QS instead of CS^t_d and QS^t_d, respectively. Since open-minded strategy subsumes narrow-minded (Theorems 1 and 4), the (pessimistic) communication complexity results of open-minded strategy hold for both (see Table 6).

Theorem 5 *If the size of the domain of the proof of s is N, then the size $|QS|$ of the Query Store at the end of the open-minded inquiry is $N/2 \leq |QS| \leq N$.*

Proof. Since all literals from rule bodies are added to QS and they are never removed from QS, in fact they all take part in proving the goal s. Moreover, negative and positive literals from the proof are added to QS only once (either l or $\neg l$). $\lhd$

Theorem 4 allows us to conclude:

Theorem 6 *If the size of the proof of s is M, then the size $|CS|$ of the Commitment Store at the end of the open-minded inquiry is $|CS| = M$.*

Proof. The total amount of information shared by all $assert$ locutions (a_i denotes number of assertions by agent i) uttered in the dialogue is:

$$\sum_{i=1}^{n} \sum_{j=1}^{a_i} |1| = |CS| = M$$

$\lhd$

Recall that n denotes the total number of agents, each holding a certain amount of (relevant) information, such that the proof of the inquiry topic from the union of all agents' belief bases is of size M (from Theorem 6). The communication complexity is polynomial in the total amount of information relevant to the proof.

Theorem 7 *Communication complexity of inquiry is $O(nM)$.*

Proof. In general there can be up to $n-1$ $requests$ per one $assert$. Thus, there can be at most M $asserts$ (agents cannot repeat assertions), $M \times (n-1)$ $requests$ and at most 2 $join$ and $leave$ locutions per one $assert$. Altogether $(n+2) \times M$ locutions exchanged before dialogue termination[5]. Therefore the communication complexity is $O(nM)$. $\lhd$

Recall that the size of the domain of the proof is N, which is the upper limit on the size of Query Store (see Theorem 5).

Theorem 8 *Computational complexity of a narrow-minded inquiry is: $M \times O(N^k)$.*

Proof. In the narrow-minded strategy, after each $assert$ the well-supported model of the CS has to be computed, which is in $O(N^k)$ (see Section 2). Thus each such step takes $O(N^k)$. However, at the termination time, the conclusion is known (obtainable in $O(1)$). Computational complexity of narrow-minded inquiry is thus $M \times O(N^k)$. $\lhd$

Theorem 9 *Computational complexity of termination of open-minded inquiry is: $O(1)$.*

Proof. Handling each $assert$ amounts to adding a rule to CS^t, which is in $O(1)$. Handling each $request$ is in $O(1)$ as it amounts to sending the whole QS^t back to the agent. $\lhd$

Theorem 9 shows that the major factor in the complexity of the termination problem of the open-minded inquiry is the communication complexity.

Theorem 10 *Obtaining the conclusion of a terminated open-minded inquiry is $O(N^k)$.*

Proof. Recall that computing the well-supported model of CS is in $O(N^k)$, where N is the size of domain. For open-minded strategy the computation of the well-supported model is only needed after the dialogue terminates, i.e., once per dialogue. $\lhd$

Characteristics	**Open-minded**	**Narrow-minded**
Open vs. Closed System	open (at least one $assert$ per $join$)	
Addressing	one-to-all	
Coordination	asynchronous	
Properties	sound and complete	not sound and not complete
Communication Complexity	$O(nM)$	$O(nM)$
Computational Complexity (Termination)	$O(1)$	$O(MN^k)$
Computational Complexity (Obtaining Conclusion)	$O(N^k)$	$O(1)$
Total Store Size	$M+N$	

Table 6: Results for open- and narrow-minded inquiries

4 Related Work and Conclusions

Exploring paraconsistency and paracompleteness in argumentation is not new: there is a number of formalisms that do not trivialize when inconsistent premises (for a survey see [Walton *et al.*, 2008; Besnard and Hunter, 2008]). In [Black and Hunter, 2009] a formal bi-party inquiry dialog system is proposed where DeLP is used to deal with ignorance and inconsistency. In [Takahashi and Sawamura, 2004] the logic of multi-valued argumentation (LMA) is used and agents can argue using multi-valued knowledge base. In [Prakken, 2010] ASPIC+, a framework for structured argumentation with possible inconsistent knowledge bases and defeasible rules is given. However, none of these formalisms

---

[5] Notice that even for hybrid forward-backward chaining, this is the pessimistic time complexity.

handles inconsistency and ignorance the way 4QL does. Usually the inconsistent premisses yield conclusions (e.g., 'undecided') which cannot be further dealt with.

As indicated in [Dignum and Vreeswijk, 2003; Traum, 2004], several new issues arise when contemplating the plurality of dialogue participants. Multi-party issues were also studied in [Yuan *et al.*, 2011], where a distributed argumentation system was given together with a multi-party dialogue game for computing the defensibility of an argument from consistent knowledge bases. In [Vreeswijk and Hulstijn, 2004], a simple multi-party inquiry dialogue assumed communication in turns with no termination criterion.

Leaving behind the realm of two-valued logical approaches to bi-party dialogues, we arrived at a solution for multi-party, paraconsistent and paracomplete inquiry. We investigated four inquiry strategies, conditional on different policies for opening and closing threads. The relevant results were evaluated against the paraconsistent and paracomplete distributed knowledge of the group.

The general outcome of our research calls for reconsidering normative models of dialogues by introducing two additional logical values: $\mathfrak{i}$ and $\mathfrak{u}$. Specifically, the novelty lies in understanding the very nature of the dialogue's goal, leading to a better discernment between inquiry and discovery and more applications of inquiry.

In future work, we intend to investigate hybrid forward-backward chaining techniques for a dialogue system, where the locutions can contain a set of rules. Next, we plan to research methods for handling inconsistencies and uncertainty in the Commitment Store via a $challenge$ locution.

5 Acknowledgments

The authors would like to thank Andrzej Szałas for his comments which greatly improved this paper.

References

[Alferes *et al.*, 2002] J. J. Alferes, A. Brogi, J. A. Leite, and L. M. Pereira. Evolving logic programs. In *Proceedings of JELIA 2002*, volume 2424 of *LNCS*, pages 50–61. Springer, 2002.

[Besnard and Hunter, 2008] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[Black and Hunter, 2009] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[Dignum and Vreeswijk, 2003] F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In *Workshop on Agent Communication Languages*, volume 2922 of *LNCS*, pages 212–230. Springer, 2003.

[Dunin-Kęplicz and Verbrugge, 2010] B. Dunin-Kęplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley, 2010.

[Dunin-Kęplicz *et al.*, 2014] B. Dunin-Kęplicz, A. Szałas, and R. Verbrugge. Tractable reasoning about group beliefs. In *2nd international Workshop on Engineering Multi-Agent Systems (EMAS 2014)*, LNAI. Springer, 2014.

[Dunin-Keplicz *et al.*, 2015] Barbara Dunin-Keplicz, Alina Strachocka, Andrzej Szalas, and Rineke Verbrugge. Paraconsistent semantics of speech acts. *Neurocomputing*, 151:943–952, 2015.

[Fagin *et al.*, 1995] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[Kushilevitz and Nisan, 1997] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, New York, NY, USA, 1997.

[Małuszyński and Szałas, 2011] J. Małuszyński and A. Szałas. Living with inconsistency and taming nonmonotonicity. In *Datalog Reloaded*, volume 6702 of *LNCS*, pages 384–398. Springer-Verlag, 2011.

[Małuszyński and Szałas, 2013] J. Małuszyński and A. Szałas. Partiality and inconsistency in agents' belief bases. In *KES-AMSTA*, volume 252 of *Frontiers in Artificial Intelligence and Applications*, pages 3–17. IOS Press, 2013.

[McBurney and Parsons, 2001] P. McBurney and S. Parsons. Chance discovery using dialectical argumentation. In *New Frontiers in Artificial Intelligence*, volume 2253 of *LNCS*, pages 414–424. Springer, 2001.

[Papadimitriou and Mihalis, 1997] Christos H. Papadimitriou and Yannakakis. Mihalis. On the complexity of database queries. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* ACM, 1997.

[Prakken, 2010] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[Schöning, 2008] Uwe Schöning. *Logic for Computer Scientists*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

[Singh, 1998] M. P. Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, December 1998.

[Szałas, 2013] A. Szałas. How an agent might think. *Logic Journal of IGPL*, 21(3):515–535, 2013.

[Takahashi and Sawamura, 2004] T. Takahashi and H. Sawamura. A logic of multiple-valued argumentation. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 800–807. IEEE Computer Society, 2004.

[Traum, 2004] D. Traum. Issues in multiparty dialogues. *Advances in Agent Communication*, pages 201–211, 2004.

[Vardi, 1982] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 137–146, New York, NY, USA, 1982. ACM.

[Vitória *et al.*, 2009] A. Vitória, J. Małuszyński, and A. Szałas. Modeling and reasoning with paraconsistent rough sets. *Fundamenta Informaticae*, 97(4):405–438, 2009.

[Vreeswijk and Hulstijn, 2004] G.A.W. Vreeswijk and J. Hulstijn. A free-format dialogue protocol for multi-party inquiry. In *In Proc. of the Eighth Int. Workshop on the Semantics and Pragmatics of Dialogue (Catalog '04)*, pages 273–279, 2004.

[Walton and Krabbe, 1995] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany (NY), 1995.

[Walton *et al.*, 2008] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[Yuan *et al.*, 2011] Jinping Yuan, Li Yao, Zhiyong Hao, Fang Liu, and Tangming Yuan. Multi-party dialogue games for distributed argumentation system. In *IAT*, pages 329–332. IEEE Computer Society, 2011.

Paraconsistent Multi-party Persuasion in TalkLOG

Barbara Dunin-Kęplicz and Alina Strachocka

Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
keplicz,astrachocka@mimuw.edu.pl

Abstract. Some conflicts appearing in multi-agent settings may be resolved via communication. In this paper, besides *conflicts of opinions*, paradigmatically resolved by *persuasion*, we study resolution of *conflicting justifications of opinions*. To cope with agents' *ignorance* and *inconsistencies*, often arising from perception and interactions, our formal framework TalkLOG employs a 4-valued logic with two additional logical values: *unknown* and *inconsistent*. Within TalkLOG we study such properties of persuasion as: termination, soundness and completeness. Another critical issue is complexity of agents' communication, typically interleaved with reasoning. In TalkLOG tractability of both aspects is obtained thanks to the implementation tool: rule-based 4-valued language 4QL.

1 Requirements for Resolving Conflicts via Persuasion

The overall goal of this research is a communication protocol for resolving conflicts by a group of agents situated in dynamic and unpredictable environments where up-to-date, unambiguous and complete information is hardly obtainable. Within our dialogue system, TalkLOG, we introduce a new approach to logical modeling of conversing agents, obeying the following principles of communication:

1. Agents' informational stance is **paraconsistent** (i.e., tolerating inconsistencies) and **paracomplete** (i.e., tolerating lack of information). Particularly, inconsistent and incomplete conclusions do not terminate reasoning, but can be further dealt with.
2. Agents are able to complete and disambiguate missing and ambiguous information.
3. **Flexible**, **multi-party** conversations are considered.
4. **Tractable** protocols are built to allow for practical applicability.
5. **Dynamics** of communication model involves beliefs change during dialogue.

Contemporary approaches to flexible communication draw upon Walton and Krabbe's semi-formal theory of dialogue, adapting the normative models of human communication to multi-agent settings. The dialogue that aims at resolving conflicts in their typology is **persuasion**, characterized as follows: "The *initial situation* of a *persuasion dialogue* (or *critical discussion*), is a clash or conflict of points of view. The *main goal* is a resolution of the initial conflict by verbal means. This means that if the dialectical process is to be successful at least one of the parties involved in the conflict will have to *change its point of view* at some stage in the dialogue. The *internal aim of each party* is to persuade the others to take over its point of view" [1].

In the whole spectrum of approaches to persuasion starting from the seminal volume [1], through more formal works [2, 3], the two-party, two-valued models prevail. Only recently multi-party aspects have been studied [4, 5], while non-classical approaches [6, 7] did not treat inconsistencies as first-class citizens. In contrast to them, the contribution of this paper is a formal, dynamic model of a tractable, paraconsistent and paracomplete multi-party persuasion, featuring *Classical* and *Deep Persuasion*, to solve *conflicts of opinions* and *conflicts of justifications of opinions*, respectively. Opposed to [6, 7], our solution is built upon the four-valued logic of [8] with an intuitive semantics behind the two new truth values: *unknown* ($\mathfrak{u}$) and *inconsistent* ($\mathfrak{i}$). Such choice permits to rationally cope with agents' ignorance and inconsistencies usually resulting from agents' interactions and the complexity of the environment.

Another critical issue is complexity of agents' communication, typically interleaved with reasoning. Instead of reasoning in logical systems of high complexity, in TalkLOG we query paraconsistent knowledge bases. To this end we use 4QL - a DATALOG$^{\neg\neg}$-like four-valued rule-based query language. In the light of the new perspective we prove such properties of paraconsistent persuasion as: termination, convergence to the merged outcome, soundness and completeness; similarly to our results obtained for inquiry [9].

The paper is structured as follows. First, in Section 2 related work is reviewed. Next, Section 3, briefly recalls the underpinnings of our solution. The main matter convey Sections 4 and 5, concerning formalization of persuasion and analysis of its properties, respectively. Finally, Section 6 concludes the paper.

2 Related Work

Walton and Krabbe introduced two types of semi-formal persuasion dialogues: permissive (PPD - everyday conversations) and rigorous (RPD - model of reasoned argument). Their "persuasion dialogue generally takes the form of a sequence of *questions and replies, or attacks and defenses* where each side takes a turn to make a move. A *move is a sequence of locutions* advanced by a participant at a particular point in the sequence of dialogue" [1]. As in PPD and RPD replies cannot be postponed, since each player's move has to pertain to the adversary's preceding move, these dialogue types do not offer a more nuanced handling of the burden of proof, which is important for increased flexibility of interlocutors. Moreover no formal properties are given.

PWA protocol [3], although suffered from similar modeling limitations (see also [10] for discussion) was a formal approach allowing to analyze formal properties, among others, termination and outcomes of dialogues. Prakken's system [2] was the first to allow alternative replies and postponing replies, thus permitting much flexibility in persuasion. Conflict resolution in [3] hinged on the preference relation between arguments, while in [2] on the priorities of reasoning rules. Still all above mentioned approaches were two-party and required that support of an argument was consistent.

Multi-party aspects were introduced to persuasion by Bonzon et al. [4], where agents shared the set of arguments, but had different attack relations among them. Although agents were privately assigned to two adverse groups, they independently proposed moves to the central authority who selected the move to play. The outcomes were juxtaposed with the merged argumentation system [11].

Several argumentation systems dealt with ignorance or inconsistency, although not necessarily applied to persuasion. Sawamura et al. [7] proposed a framework for multiple-valued argumentation (LMA) where agents can argue using multi-valued knowledge base. In [6] ASPIC+, a framework for structured argumentation with possible inconsistent knowledge bases and defeasible rules was given. However, none of these formalisms handles inconsistency or lack of information the way 4QL does. Usually the inconsistent premisses yield conclusions (e.g., 'undecided') which terminate the reasoning process, thus cannot be further dealt with, unlike in our approach.

3 4QL as an Implementation Tool

TalkLOG uses the logical language introduced in [12–14]. This allows to encode agents' informational stance in the rule-based query language 4QL[1] defined in [12], further developed in [13], and based on the 4-valued logic of [8]. 4QL features:

- Possibly many, perhaps distributed information sources.
- Four logical values ($\mathfrak{t}$, $\mathfrak{f}$, $\mathfrak{i}$, $\mathfrak{u}$).
- Unrestricted negation (in premisses and conclusions of rules).
- Simple tools: rules, modules and multi-source formulas to formulate and enrich (lightweight versions of) (Local) Closed World Assumption, autoepistemic reasoning, default reasoning, defeasible reasoning, etc.
- Modular architecture to deal with unknown or inconsistent conclusions without enforcing termination of the reasoning process.
- PTime complexity of computing queries while capturing all tractable queries.

For convenience, both the underlying 4-valued logic and 4QL are recalled from [8, 12–14] in Appendix[2]. In what follows all sets are finite except for sets of formulas; domain and language are fixed and the programs and rules are ground. We deal with the classical first-order language over a given vocabulary without function symbols. Γ denotes the set of all facts; Π denotes the set of all rules.

The semantics of 4QL is defined by *well-supported models*, i.e., models consisting of (positive or negative) ground literals, where each literal is a conclusion of a derivation starting from facts. For any set of rules, such a model is uniquely determined and computable in deterministic polynomial time $O(N^k)$ where N is the size of domain and $k = max(s, t)$ where s is the maximal arity of relations and t is the maximum number of free variables. As we deal here with ground programs, $t = 0$. When s is a bound constant, what takes place in qualitative not quantitative reasoning, being all practical applications of 4QL, tractability is achieved.

4 Persuasion in TalkLOG

Although traditionally persuasion arises from a conflict of **opinions** [1], the following example illustrates other possibilities.

Example 1 (John & Mark). Two friends John and Mark are saving up money together (expressed by $save(money)$) and every week they are paying an agreed amount into their common bank account. However, John wants to buy a motorcycle with that money (expressed by the rule $save(money)$:– $buy(moto)$) and Mark plans to open a small bar ($save(money)$:– $buy(bar)$). As there is no outspoken initial conflict of opinions regarding saving money ($save(money)$ is $\mathfrak{t}$ for both of them), classically, no persuasion can commence. However, the concealed disagreement concerns their motivations. ◁

---

[1] Open-source implementation of 4QL is available at `4ql.org`.

[2] Availalble at `http://4ql.org/downloads/appendix.pdf`.

In TalkLOG, if the friends want to resolve the issue immediately (instead of fighting over saved money later), they can enter into a discussion about their *differing motivations*. Ultimately, as an outcome from such dialogue, they would:

- abandon one of the goals and focus on the other, or
- continue to save money for both goals, or
- give up on saving at all since they could not come to an agreement.

We formalize motivation (warrant or justification) as the *proper proof* of a formula (see Def. 2). To this end, the notion of the *dependence set of a literal ℓ from a program P* is needed. Intuitively, it consists of literals reachable via backward chaining on P from ℓ.

Definition 1 (Dependence Set). Let ℓ be a literal and P a 4QL program. The *dependence set* of ℓ from P, denoted $\mathcal{D}_{P,\ell}$ is the set of literals such that:

- $\neg\ell, \ell \in \mathcal{D}_{P,\ell}$,
- if there is a rule $\ell' :\!- b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}$ in P, such that $\ell' \in \mathcal{D}_{P,\ell}$ then $\forall_{j\in 1..m}\forall_{k\in 1..i_j}\ b_{jk}, \neg b_{jk} \in \mathcal{D}_{P,\ell}$. $\triangleleft$

The *proof* of a literal ℓ from a program P is a subprogram S of P generated from the dependence set $\mathcal{D}_{P,\ell}$ by taking all rules and facts of P whose conclusions are in $\mathcal{D}_{P,\ell}$. Notice that proof may contain rules whose premisses evaluate to $\mathfrak{f}$ or $\mathfrak{u}$, thus do not influence the value of ℓ. The definition of *proper proof* disregards such rules.

Definition 2 (Proof, Proper Proof). Let ℓ be a literal, P a 4QL program and $\delta \in \Gamma \cup \Pi$ a fact or a rule. The *proof* of l from P is a 4QL program $S \subseteq P$ such that $\delta \in S$ iff $head(\delta) \in \mathcal{D}_{P,\ell}$. The *proper proof* (p-proof) or *warrant* of l from P denoted $\Phi_{l,P}$, is a subset of $S_{l,P}$ such that $\delta \in \Phi_{l,P}$ iff $body(\delta)(\mathcal{M}_{S_{l,P}}) \in \{\mathfrak{t}, \mathfrak{i}\}$. $\triangleleft$

By equal/different motivations we mean equal/unequal p-proofs. By *equal opinions* we mean equal valuations of the formulas representing opinions. Obviously, equality of warrants entails equality of opinions, but not the other way around. We differentiate between cases where *initial situation* concerns *conflict of opinions* or *conflict of warrants* and we are interested in *how* the initial conflict is resolved, i.e., whether a *common opinion* or a *common warrant* has been reached. Although [1] distinguish three types of points of view (opinions) towards a topic of persuasion: positive, negative and one of doubt, in TalkLOG, the 'doubtful' point of view is expanded, distinguishing the cases when the doubt results from ignorance or from inconsistency.

Definition 3 (Initial Conflict on Topic). Let:

- $\varphi \in \mathbb{C}$ be a ground literal, representing the topic of dialogue,
- P_1 and P_2 be two 4QL programs of agents A_1 and A_2,
- $\mathcal{M}_{P_1}$ and $\mathcal{M}_{P_2}$ be the well-supported models of P_1 and P_2 respectively,
- Φ_{φ,P_1} and Φ_{φ,P_2} be the p-proofs of φ from P_1 and P_2 respectively.

Then:

- an *initial conflict on topic* φ between A_1 and A_2 occurs when:
  - $\varphi(\mathcal{M}_{P_1}) \neq \varphi(\mathcal{M}_{P_2})$, or [conflict of opinion]
  - $\Phi_{\varphi,P_1} \neq \Phi_{\varphi,P_2}$ [conflict of warrant]
- A_1 and A_2 share a *common opinion on* φ if $\varphi(\mathcal{M}_{P_1}) = \varphi(\mathcal{M}_{P_1})$,
- A_1 and A_2 share a *common warrant on* φ if $\Phi_{\varphi,P_1} = \Phi_{\varphi,P_2}$.

The goal of **Classical Persuasion** is a common opinion, while of **Deep Persuasion** - common warrant. Unless stated otherwise, the formalism concerns both dialogues.

4.1 Locutions and Moves

In TalkLOG, the *content* of a locution is either an opinion (belief) of an agent, represented by a literal φ together with its value v; or a piece of evidence, represented by a fact or a rule δ together with its membership function $\mu(\delta)$ (see Definition 4 below).

Definition 4. Let P be a 4QL program and $\delta \in \Gamma \cup \Pi$ be a fact or a rule. Then:

$$\mu_P(\delta) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ when } \delta \in P \wedge \neg\delta \notin P; \\ \mathfrak{f} \text{ when } \delta \notin P \wedge \neg\delta \in P; \\ \mathfrak{i} \text{ when } \delta, \neg\delta \in P; \\ \mathfrak{u} \; otherwise. \end{cases}$$

Definition 5. A *locution* is a tuple of the form $\langle Agent, Dialogue, SpeechAct, Content\rangle$ for simplicity denoted further as $SpeechAct^{Dialogue}_{Agent}\langle Content\rangle$, where:

- $Agent$ is the identifier of the sender of the utterance: $Agent \in \mathcal{X}_{Ag}$,
- $SpeechAct$ is the type of locution: $SpeechAct \in \{assert, assertBel, concede, why, retract, adopt, retractBel\}$,
- $Dialogue$ is the identifier of the dialogue: $Dialogue \in \mathcal{X}_{Dial}$,
- $Content$ is the propositional content of locution, dependent on speech act type. $\triangleleft$

The format and intended meaning of locutions permitted in TalkLOG persuasion[3], i.e.:

- assertions of evidence or beliefs,
- questioning evidence or beliefs,
- concessions, retraction or adoption of evidence,
- retraction of beliefs,

are given in Tab. 1, where $x \in \mathcal{X}_{Ag}$, $d \in \mathcal{X}_{Dial}$ and:

- $\varphi \in \mathbb{C}$ is a ground literal, $\delta \in \Gamma \cup \Pi$ is a fact or a rule,
- $v, \mu(\delta), v' \in \mathbb{T}$ is a truth value, such that $v' \neq \mathfrak{u}$,
- B is a body of a rule.[4]

Table 1: Formats and intended meaning of permissible locutions in TalkLOG persuasion.

Locution Type	**Format**	**Intended Meaning**
assert	$assert^d_x\langle\delta, \mu(\delta)\rangle$	asserting attitude towards evidence δ
concede	$concede^d_x\langle\delta, \mu(\delta)\rangle$	conceding/agreeing with evidence δ
assertBel assertBel	$assertBel^d_x\langle\varphi, v\rangle$ $assertBel^d_x\langle B, v\rangle$	asserting attitude towards an opinion φ asserting attitude towards an opinion B
why	$why^d_x\langle\varphi, v'\rangle$	questioning opinion φ
retract	$retract^d_x\langle\delta\rangle$	retracting evidence δ
adopt	$adopt^d_x\langle\delta\rangle$	adopting evidence δ
retractBel	$retractBel^d_x\langle\varphi, v\rangle$	retracting opinion φ

The set of all locutions that match the format presented in Tab. 1 is denoted $\mathcal{U}$. A **move** is a sequence of locutions uttered by the same agent in the same timepoint. $Moves$ denotes the set of all permissible persuasion moves (where neither agent's knowledge base nor its beliefs can change during a move), so in a single move an agent cannot:

- assert different points of view towards the same belief,
- assert and retract the same belief,
- both (concede or assert) and (retract or adopt) a piece of evidence.

---

[3] We skip 'operational' locutions like $requestAll, join, leave$ for simplicity.

[4] $assertBel\langle B, v\rangle$ is a notation for a sequence of $assertBel$ concerning literals in B.

4.2 Dialogue Stores

TalkLOG persuasion merges two approaches:

- resolving conflicts of opinions via argumentation (and embedded 4QL mechanisms),
- employing dedicated conflict resolution methods to adjudicate between conflicting pieces of evidence. To this end we have chosen social choice theory methods [15] (particularly, voting) for resolving conflicts unsolvable via argumentation.

In the mentalistic approach [16–18] the semantics of locutions was defined by means of their pre- and post-conditions; for a related paraconsistent semantics consult [19]. However, recently pre-conditions are realized via *relevance function* (see Section 4.3), while post-conditions – as updates of so-called *dialogue stores*. TalkLOG persuasion requires: Query (QS), Dispute (DS), Resolved Dispute (RDS) and Commitment Stores (CS). This permits to validate correctness of moves using public information [20] (i.e., these stores' content) rather than internal states of agents.

QS QS contains beliefs and questions uttered by agents. Agents may inspect QS to find questions that need answering or opinions of others that can be questioned.

DS DS contains pieces of evidence put forward by agents to support a belief or respond to a question. Agents may query DS to support or undermine a piece of evidence submitted by another agent.

RDS contains both traces of resolved conflicts as well as unanimously accepted pieces of evidence. To comply with those decisions, participants are required to adopt a piece of evidence accepted by the group (using *adopt*) or abandon one (using $retract$ locution) via voting, but consult e.g., [21].

CS CS contains the agreed-upon pieces of evidence, which once accepted are never deleted. The fact that CS grows monotonically is important for both complexity considerations and analysis of the properties of TalkLOG persuasion.

QS contains tuples of the form $\langle Q, F, V, X\rangle$ where $Q \in \{bel, why\}$ denotes tuple type, $F \in \mathbb{C}$ ground literal, $V \in \mathbb{T}$ a point of view towards the formula, and $X \in \mathcal{X}_{Ag}$ the sender. QS Update Function defines how QS changes after a move:

- $assertBel_A(\varphi, v)$ results in creating a new tuple $\langle bel, \varphi, v, A\rangle$,
- $why_A(\varphi, v)$ results in creating a new tuple $\langle why, \varphi, v, \bot\rangle$,
- $retractBel_A(\varphi, v)$ results in removing $\langle bel, \varphi, v, A\rangle$ from QS, and, if that was the last opinion v about φ in QS, removing also tuple $\langle why, \varphi, v, \bot\rangle$ (if one exists).

Definition 6. Let $F_d^t : \{bel, why\} \times \mathbb{C} \times \mathbb{T} \times \mathcal{X}_{Ag}$ and $m^t = u_1; \ldots; u_k$ be the move received at time t. Then, *QS Update Function* $update_{QS} : F_d^t \times Moves \to F_d^t$ is:

$$update_{QS}(S, u_1; \ldots; u_k) \overset{\text{def}}{=} \begin{cases} update_{QS}(step_{QS}(S, u_1), u_2; \ldots; u_k) & \text{if } k > 1 \\ step_{QS}(S, u_1) & \text{if } k = 1, \end{cases}$$

where $step_{QS} : F_d^t \times \mathcal{U} \to F_d^t$ is a one-step update function defined as follows:

$$step_{QS}(S, u) \overset{\text{def}}{=} \begin{cases} S \cup \{\langle why, \varphi, v, \bot \rangle\} & \text{if } u = why^{d}_{A}\langle \varphi, v \rangle; \\ S \cup \{\langle bel, \varphi, v, A \rangle\} & \text{if } u = assertBel^{d}_{A}\langle \varphi, v \rangle; \\ S \setminus \{\langle bel, \varphi, v, A \rangle\} & \text{if } u = retractBel^{d}_{A}\langle \varphi, v \rangle \wedge \\ & \exists_{\langle bel, \varphi, v, A' \rangle \in S} \text{ s.t. } A' \neq A; \\ S \setminus \{\langle bel, \varphi, v, A \rangle, \langle why, \varphi, v, \bot \rangle\} & \text{if } u = retractBel^{d}_{A}\langle \varphi, v \rangle \wedge \\ & \neg\exists_{\langle bel, \varphi, v, A' \rangle \in S} \text{ s.t. } A' \neq A; \\ S & \text{otherwise.} \end{cases}$$

Definition 7 (Query Store). Let m^t be the move received at time t, and $update_{QS}$ be the QS Update Function. Then, *Query Store* of persuasion d at time t, initiated by A_{init}, is a finite set of tuples, denoted $QS^t_d : \{bel, why\} \times \mathbb{C} \times \mathbb{T} \times \mathcal{X}_{Ag}$, s.t.:

$$QS^t_d \overset{\text{def}}{=} \begin{cases} \{\langle bel, s, v_i, A_{init} \rangle\} & \text{if } t = 0 \\ update_{QS}(QS^{t-1}_d, m^t) & \text{otherwise} \end{cases}$$

Dispute Store contains tuples of form $\langle \delta, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}} \rangle : (\mathit{\Gamma} \cup \mathit{\Pi}) \times \mathbb{N}^4$. We will write:

$$DS[\delta, k] \overset{\text{def}}{=} n_k,\ DS[\neg\delta, k] \overset{\text{def}}{=} DS[\delta, \neg k],\ \text{where } k \in \mathbb{T}.$$

We use $DS[\delta]$ to test if an entry for δ is in DS, and $DS[\delta, k]$++ to increment n_k counter for δ:

$$DS[\delta] \overset{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ iff } \langle \delta, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}} \rangle \in DS; \\ \mathfrak{f} \text{ iff } \langle \delta, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}} \rangle \notin DS; \end{cases}$$

$$\begin{aligned} DS[\delta, k]\text{++} \overset{\text{def}}{=} & \{\langle x, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}} \rangle \in DS | x \neq \delta\} \cup \\ & \{\langle \delta, x_{\mathfrak{t}}, x_{\mathfrak{f}}, x_{\mathfrak{i}}, x_{\mathfrak{u}} \rangle \in DS : x_k = DS[\delta, k] + 1\} \end{aligned}$$

DS Update Function defines the way DS is updated after each move:

- $assert$ results in creating a new tuple for the propositional content of assertion (unless already exist) and increasing the support counter for the asserted value,
- $concede$ increases the support counter for the uttered value if the relevant tuple exist in DS.

Definition 8. Let $m^t = u_1; \ldots; u_k$ be the move received at time t, $\delta \in \mathit{\Gamma} \cup \mathit{\Pi}$; $v \in \mathbb{T}$ and $V^t_d : (\mathit{\Gamma} \cup \mathit{\Pi}) \times \mathbb{N}^4$. Then, *DS Update Function* $update_{DS} : V^t_d \times Moves \rightarrow V^t_d$ is:

$$update_{DS}(S, u_1; \ldots; u_k) \overset{\text{def}}{=} \begin{cases} update_{DS}(step_{DS}(DS, u_1), u_2; \ldots; u_k) & \text{if } k > 1 \\ step_{DS}(DS, u_1) & \text{if } k = 1 \end{cases}$$

where $step_{DS} : V^t_d \times \mathcal{U} \rightarrow V^t_d$ is a one-step update function defined as follows:

$$step_{DS}(DS, u) \overset{\text{def}}{=} \begin{cases} X[\delta, v] + + & \text{where } X = DS \cup \{\langle \delta, 0, 0, 0, 0 \rangle\} \text{ and} \\ & u = assert^{d}_{A}\langle \delta, v \rangle \text{ and } DS[\delta] = \mathfrak{f}; \\ DS[\delta, v] + + & \text{if } (u = assert^{d}_{A}\langle \delta, v \rangle \vee u = concede^{d}_{A}\langle \delta, v \rangle) \text{ and } DS[\delta] = \mathfrak{t}; \\ DS & \text{otherwise.} \end{cases}$$

Definition 9 (Dispute Store). Let m^t be the move received at time t and $update_{DS}$ be the DS Update Function. Then, *Dispute Store* of persuasion d at time t is a finite set of tuples, denoted as $DS^t_d : (\mathit{\Gamma} \cup \mathit{\Pi}) \times \mathbb{N}^4$ such that:

$$DS^t_d \overset{\text{def}}{=} \begin{cases} \emptyset & \text{if } t = 0 \\ update_{DS}(DS^{t-1}_d, m^t) & \text{otherwise} \end{cases}$$

The entries in RDS are of the form $\langle(\neg)\delta, i\rangle$ or $\langle(\neg)\delta, o\rangle$, meaning the piece of evidence is either admitted (*"in"*) or rejected (*"out"*). After each increase of support counter in DS for δ, the voting function is called, which, if there are enough casting votes, adjudicates about δ (unlike the argumentation approach where each move may change the status of an argument). Recall, that each agent A_i votes with its $\mu_{P_{A_i}}(\delta)$.

Definition 10. Let $Status = \{i, o\}$, $\delta \in \mathit{\Gamma} \cup \mathit{\Pi}$, and $n \in \mathbb{N}$ be the number of dialogue participants. Then, a *voting function VF* is any function $VF : (\mathit{\Gamma} \cup \mathit{\Pi}) \times \mathbb{N}^5 \rightarrow Status^2$, such that $VF(\delta, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}}, n) = \emptyset$ iff $n_{\mathfrak{t}} + n_{\mathfrak{f}} + n_{\mathfrak{i}} + n_{\mathfrak{u}} \neq n$, and if $\delta \in \mathit{\Pi}$ (a rule) then $VF(\delta, n_{\mathfrak{t}}, n_{\mathfrak{f}}, n_{\mathfrak{i}}, n_{\mathfrak{u}}, n) \neq \langle i, i\rangle$.

From several possible outcomes of voting over δ, we forbid accepting antithetic rules (expressed by $\langle i, i\rangle$). If $\langle a, b\rangle$ is the outcome for δ, then a is the status of δ and b of $\neg\delta$.

Definition 11 (Resolved Dispute Store). Let m^t be the move received at time t, and:

- DS_d^t be the Dispute Store of dialogue d at time t,
- VF be a voting function,
- $\delta \in \mathit{\Gamma} \cup \mathit{\Pi}$; $v \in \mathbb{T}$; $s_1, s_2 \in Status$,
- $n \in \mathbb{N}$ be the number of dialogue participants.

Then, *Resolved Dispute Store* of persuasion d at time t is a finite set of tuples denoted as $RDS_d^t : (\mathit{\Gamma} \cup \mathit{\Pi}) \times Status$ such that:

- $RDS_d^0 = \emptyset$
- $RDS_d^t = RDS_d^{t-1} \cup X$, where: $X = \{\langle\delta, s_1\rangle \cup \langle\neg\delta, s_2\rangle \mid$
  (a) $concede_S^d\langle\delta, v\rangle \in m^t \vee assert_S^d\langle\delta, v\rangle \in m^t$,
  (b) $\langle s_1, s_2\rangle = VF(\delta, DS_d^t[\delta, \mathfrak{t}], DS_d^t[\delta, \mathfrak{f}], DS_d^t[\delta, \mathfrak{i}], DS_d^t[\delta, \mathfrak{u}], n)\}$.
- $RDS_d^t = RDS_d^{t-1}$ otherwise. ◁

Commitment Store is updated always after RDS is updated with an entry with status "in". The final value of the Deep Persuasion topic is evaluated within the context of this store. In fact, CS is just an evolving 4QL program [22].

Definition 12 (Commitment Store). Let:

- RDS_d^t be the Resolved Dispute Store of dialogue d at time t,
- m^t be the message received at time t,
- $\delta \in \mathit{\Gamma} \cup \mathit{\Pi}$; $v \in \mathbb{T}$.

Then, *Commitment Store* of a persuasion dialogue d at time t is a 4QL program denoted CS_d^t such that:

- $CS_d^0 = \emptyset$
- $CS_d^t = CS_d^{t-1} \cup X$, where $X = \{\delta \mid$
  (a) $concede_A^d\langle\delta, v\rangle \in m^t \vee assert_A^d\langle\delta, v\rangle \in m^t$,
  (b) $\langle\delta, i\rangle \in RDS_d^t\}$,
- $CS_d^t = CS_d^{t-1}$ otherwise. ◁

The conclusion of persuasion on topic s with initial value v_i is the final value v_f of s, together with the justification (p-proof) $\mathit{\Phi}_{s, CS_d^t}$ for that value, obtained in the dialogue. For the purpose of this paper we distinguish two termination conditions:

- **Impasse**: when no agent has a relevant move to make,
- **Common Opinion**: when all agents agree on the value of the topic.

Definition 13. Let CS^t_d be the Commitment Store of Deep Persuasion d terminating at t, with the topic s of initial value v_i. Then, the *conclusion* of d is $c = \langle v_f, S\rangle$ where:

- $v_f = s(\mathcal{M}_{CS^t_d})$, where $\mathcal{M}_{CS^t_d}$ is the well-supported model of CS^t_d,
- $S = \Phi_{s,CS^t_d}$, i.e., S is the p-proof of s from CS^t_d.

Definition 14. Let QS^t_d be the Query Store and CS^t_d be the Commitment Store of Classical Persuasion d terminating at t with n participants $A_1, \ldots, A_n$, with the topic s of initial value v_i. Then, the *conclusion* of d is $c = \langle v_f, S\rangle$ where:

- $n = |\langle bel, s, v_f, X\rangle \in QS^t_d : X \in \{A_1, \ldots A_n\}|$,
- $S = \Phi_{s,CS^t_d}$, i.e., S is the p-proof of s from CS^t_d.

Obtaining conclusion of terminated Deep Persuasion amounts to computing the well-supported model of CS^t_d just once, at the end of dialogue. Thus, this problem is in $O(N^k)$ (see Section 3) where N is the size of domain and k is the maximal arity of relations. Obtaining conclusion of terminated Classical Persuasion is in $O(N) = |QS^t_d|$ (QS^t_d may possibly contain all beliefs of agents at termination time).

4.3 Move and Locution Relevance

To ensure coherence and focus of persuasion, the notion of **move relevance** is employed after [23]. As move is a sequence of locutions, relevance of each locution is verified on the basis of dialogue stores' content at the given timepoint t and the locution content (arbitrarily, irrelevant locutions are ignored). For ease of presentation, dialogue stores are updated after each relevant locution, so that locution relevance can be defined without reference to preceding locutions in the move. Relevant locutions are:

1. Assertions of a belief $assertBel^d_S\langle\psi, v\rangle$ if it concerns:
   (a) the topic of the dialogue,
   (b) a belief asserted by another agent,
   (c) premisses or head of a rule present in DS, or a fact present in DS.
2. Questions about a belief $why^d_S\langle\varphi, v\rangle$ if the opinion $v \in \{\mathsf{t}, \mathsf{f}, \mathsf{i}\}$ about φ is in QS.
3. Assertions of a fact or rule $assert^d_S\langle\delta, v\rangle$ if
   (a) a question about a belief concerning $head(\delta)$ is in QS,
   (b) it is present in DS (an assertion works as concession then).
4. Concessions of a fact or a rule $concede^d_S\langle\delta, v\rangle$ present in DS.
5. Retractions of a belief $retractBel^d_S\langle\varphi, v\rangle$ if relevant belief of agent S is in QS.
6. Adoptions of a fact or a rule $adopt^d_S\langle\delta, v\rangle$ if $\langle\delta, i\rangle \in RDS$.
7. Rejections of a fact or a rule $reject^d_S\langle\delta, v\rangle$ if $\langle\delta, o\rangle \in RDS$.

An irrelevant move is a move without relevant locutions. Notice that an irrelevant locution $u_i \in m^t$ may be relevant at $t' \neq t$, but a non-permissible move m^t will not be permissible at any other time t'. Lemma 1 illustrates that in a single move an agent can always utter its entire p-proof of a formula.

Lemma 1. *If $\Phi_{s,P} \neq \emptyset$ and $s(\mathcal{M}_P) = v$, then while $\langle why, s, v, \bot\rangle \in QS$ there exists a relevant move $m = u_1; \ldots; u_n$ s.t.*[5]*:*

$$\Phi_{s,P} = \bigcup \{\delta[\mu_P(\delta) \in \{\mathsf{t}, \mathsf{i}\}], \neg\delta[\mu_P(\delta) \in \{\mathsf{f}, \mathsf{i}\}] \in P : \\ \exists_{i \in 1..n}\, u_i = assert\langle\delta, \mu_P(\delta)\rangle \vee u_i = concede\langle\delta, \mu_P(\delta)\rangle\}.$$

Proof follows from move/locution relevance definition.

---

[5] $[\cdot]$ is the Iverson bracket.

4.4 Working Example

The standard example [2] is adapted to the 3-agent case introducing agent Tom and additional steps 4' and 8'. The agents share an ontology, where module top contains top-level beliefs while module $news$ beliefs about a specific information source (here a newspaper). At $t = 0$, Paul's program P^0 consists of modules top_P^0 and $news_P^0$: $P^0 = \{top_P^0, news_P^0\}$. Olga's and Tom's programs are denoted likewise. Table 2 concisely presents evolving programs of three agents. We refer to their elements by numbers (e.g., rule $\neg\texttt{safe}(\texttt{car}) \text{ :- } \texttt{high}(\texttt{maxspeed}, \texttt{car})$ from module top has number 3 and at $t = 0$ is present in Olga's and Tom's programs, as indicated by × in columns O^0 and T^0). Program P^0 uniquely determines the well-supported model for modules top_P^0 and $news_P^0$ ($\mathcal{M}_{top_P^0}$ and $\mathcal{M}_{news_P^0}$ resp.), as given in Table 3. We refer to beliefs by capital letters (e.g., A for $safe(car)$, thus $\mathcal{M}_{top_P^9} = \{A, \neg A, B, C\}$, but $\mathcal{M}_{top_O^0} = \{\neg A, C\}$).

Table 2: Evolving Programs of Paul, Olga and Tom.

	Module *top* :	P^0	O^0	T^0	$O^{4'}$	P^9
1.	$safe(car) \text{ :- } has(airbag, car).$	×				×
2.	$\neg safe(car) \text{ :- } news.reports(airbagExplosions, car),$ $news.reliable(airbagExplosions) \in \{\mathfrak{t}, \mathfrak{u}, \mathfrak{i}\}.$			×		
3.	$\neg safe(car) \text{ :- } high(maxspeed, car).$		×	×	×	×
4.	$has(airbag, car).$	×		×	×	×
5.	$high(maxspeed, car).$		×	×	×	×
	Module *news* :	P^0	O^0	T^0	$O^{4'}$	P^9
6.	$\neg reliable(airbagExplosions) \text{ :- }$ $concerns(airbagExplosions, technology).$	×				×
7.	$reports(airbagExplosions, car).$	×	×	×	×	×
8.	$concerns(airbagExplosions, technology).$	×	×	×	×	×

Table 3: Evolving Beliefs of Paul, Olga and Tom.

	Literals from module *top* :	P^0	O^0	T^0	$O^{4'}$	P^9
A.	$safe(car)$	$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{i}$
B.	$has(airbag, car)$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
C.	$high(maxspeed, car)$	$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$

	Literals from module *news* :	P^0	O^0	T^0	$O^{4'}$	P^9
D.	$reliable(airbagExplosions)$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{f}$
E.	$reports(airbagExplosions, car)$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
F.	$concerns(airbagExplosions, technology)$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$

The complete dialogue conduct is given in Table 4, where column 4 shows the **change** in Query Store between consecutive moves. We elaborate only on the first few steps:

1	`Paul:`	"My car is safe."	$assertBel\langle A, \mathfrak{t}\rangle$
2	`Olga:`	"Why is your car safe?"	$why\langle A, \mathfrak{t}\rangle$
3	`Paul:`	"Since it has an airbag."	$assert\langle 1, \mathfrak{t}\rangle$; $assertBel\langle B, \mathfrak{t}\rangle$; $assert\langle 4, \mathfrak{t}\rangle$
4	`Tom:`	"That is true, but this does not make your car safe."	$concede\langle 4, \mathfrak{u}\rangle$; $assertBel\langle A, \mathfrak{f}\rangle$
4'	`Olga:`	"Yes, exactly."	$concede\langle 4, \mathfrak{u}\rangle$; $assertBel\langle A, \mathfrak{f}\rangle$

`5 Paul:` *"Why does that not make my care safe?*
`6 Tom:` *"Since the newspapers recently reported on airbags expanding without cause.*
`7 Paul:` *"Yes, that is what the newspapers say but that does not prove anything, since newspaper reports are very unreliable sources of technological information."*
`8 Olga:` *"Still your car is still not safe, since its maximum speed is very high."*
`8′ Tom:` *"That's right."*
`9 Paul:` *OK, I was wrong that my car is safe."*

Notice that after Olga's concession (Tab. 4, step 4') her program should change, adopting the fact `has(airbag, car)`. Thus, her belief structure may change such that new reasoning lines become available and/or the already uttered beliefs are no longer up-to-date. The technical contribution of our solution allows to model such dialogues.

Table 4: Conduct of Paul, Olga & Tom persuasion on subject $safe(car)$.

t	S	m^t	ΔQS^t	DS^t	RDS^t	CS^t
1	P	$assertBel\langle A, \mathfrak{t}\rangle$	$+\langle bel, A, \mathfrak{t}, P\rangle$	$\emptyset$	$\emptyset$	$\emptyset$
2	O	$why\langle A, \mathfrak{t}\rangle$	$+\langle why, A, \mathfrak{t}, \perp\rangle$	$\emptyset$	$\emptyset$	$\emptyset$
3	P	$assert\langle 1, \mathfrak{t}\rangle$; $assertBel\langle B, \mathfrak{t}\rangle$; $assert\langle 4, \mathfrak{t}\rangle$	$+\langle bel, B, \mathfrak{t}, P\rangle$	$\langle 1,1,0,0,0\rangle$, $\langle 4,1,0,0,0\rangle$	$\emptyset$	$\emptyset$
4	T	$concede\langle 4, \mathfrak{t}\rangle$; $assertBel\langle A, \mathfrak{f}\rangle$	$+\langle bel, A, \mathfrak{f}, T\rangle$	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,0\rangle$	$\emptyset$	$\emptyset$
4'	O	$concede\langle 4, \mathfrak{u}\rangle$; $assertBel\langle A, \mathfrak{f}\rangle$	$+\langle bel, A, \mathfrak{f}, O\rangle$	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$	4
5	P	$why\langle A, \mathfrak{f}\rangle$	$+\langle why, A, \mathfrak{f}, \perp\rangle$,	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$	4
6	T	$assert\langle 2, \mathfrak{t}\rangle$; $assertBel\langle E, \mathfrak{t}\rangle$; $assertBel\langle D, \mathfrak{u}\rangle$;	$+\langle bel, E, \mathfrak{t}, T\rangle$, $+\langle bel, D, \mathfrak{u}, T\rangle$	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,0\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$	4
7	P	$concede\langle 2, \mathfrak{u}\rangle$; $assertBel\langle D, \mathfrak{f}\rangle$; $assert\langle 6, \mathfrak{t}\rangle$; $assertBel\langle F, \mathfrak{t}\rangle$	$+\langle bel, D, \mathfrak{f}, P\rangle$, $+\langle bel, F, \mathfrak{t}, P\rangle$	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,1\rangle$, $\langle 6,1,0,0,0\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$,	4
8	O	$adopt\langle 4, \mathfrak{t}\rangle$; $assert\langle 3, \mathfrak{t}\rangle$; $assertBel\langle C, \mathfrak{t}\rangle$; $assert\langle 5, \mathfrak{t}\rangle$	$+\langle bel, C, \mathfrak{t}, O\rangle$,	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,1\rangle$, $\langle 6,1,0,0,0\rangle$, $\langle 3,1,0,0,0\rangle$, $\langle 5,1,0,0,0\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$,	4
8'	T	$concede\langle 3, \mathfrak{t}\rangle$; $concede\langle 5, \mathfrak{t}\rangle$		$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,1\rangle$, $\langle 6,1,0,0,0\rangle$, $\langle 3,2,0,0,0\rangle$, $\langle 5,2,0,0,0\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$,	4
9	P	$concede\langle 3, \mathfrak{u}\rangle$; $concede\langle 5, \mathfrak{u}\rangle$		$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,1\rangle$, $\langle 6,1,0,0,0\rangle$, $\langle 3,2,0,0,1\rangle$, $\langle 5,2,0,0,1\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$, $\langle 3, i\rangle$, $\langle \neg 3, o\rangle$ $\langle 5, i\rangle$, $\langle \neg 5, o\rangle$	4, 3, 5
9b	P	$adopt\langle 3, \mathfrak{t}\rangle$; $adopt\langle 5, \mathfrak{t}\rangle$; $retractBel\langle A, \mathfrak{t}\rangle$	$-\langle bel, A, \mathfrak{t}, P\rangle$, $-\langle why, A, \mathfrak{t}, \perp\rangle$,	$\langle 1,1,0,0,0\rangle$, $\langle 4,2,0,0,1\rangle$, $\langle 2,1,0,0,1\rangle$, $\langle 6,1,0,0,0\rangle$, $\langle 3,2,0,0,1\rangle$, $\langle 5,2,0,0,1\rangle$	$\langle 4, i\rangle$, $\langle \neg 4, o\rangle$, $\langle 3, i\rangle$, $\langle \neg 3, o\rangle$, $\langle 5, i\rangle$, $\langle \neg 5, o\rangle$	4, 3, 5

In step 9b Paul retracts his original belief. However, he still disagrees with others on that matter as $safe(car)$ is $\mathfrak{f}$ for Olga and Tom but it is $\mathfrak{i}$ for Paul at time $t = 9$ (see Table 3). Note that Impasse criterion is not met yet, since there are still relevant moves to make, e.g., a concession of rule 1 by Olga and Tom.

5 Selected Properties

The following assumptions about participating agents allow us to verify quality and completeness of the obtained results.

[Cooperativeness] We deal with a finite set of n cooperative agents who do not withhold information. This implicitly constraints the number of queries to dialogue stores per one locution. Agents' belief bases are encoded as finite, ground 4QL programs $P_1, \ldots, P_n$, that share a common ontology.

[Activeness] In between $join$ and $leave$, an agent must make at least one relevant move.

[Compliance] Agents' programs change during dialogue according to the current state of RDS.

[Sincerity] Agents do not lie about their beliefs nor contents of their programs.

[Pragmatism] Particular agents cannot repeat $assert$ and $concede$ locutions.

Agents communicate one-to-all without coordination. Their final beliefs are expressed by the well-supported models $\mathcal{M}_{P_1}, \ldots, \mathcal{M}_{P_n}$ of the programs. Note that agents **can** repeat $assertBel$ locutions, provided they are separated by $retractBel$: simply in the light of new evidence agents' beliefs may change.

Theorem 1. *Persuasion dialogues terminate.*

Proof. Regardless the termination criterion, termination of persuasion follows trivially from the fact that we deal with finite 4QL programs, and from the assumptions of activeness (disallowing joining and leaving dialogue endlessly), pragmatism (disallowing repeating specific locutions), sincerity (disallowing inventing beliefs or evidence), compliance (disallowing infinitely expanding programs, since in the course of dialogue agents' programs change only to reflect resolved conflicts, so the size of CS can be bounded by the union of all participants' programs). $\triangleleft$

Theorem 2. *Persuasion terminating on Impasse is a Deep Persuasion.*

Proof Assume d is a persuasion dialogue on topic s terminating at t on Impasse criterion, with participants $A_1, \ldots, A_n$, s.t. A_i's program at t is P_i^t. Then, A_i's p-proof of s at t is $\mathit{\Phi}_{s,P_i^t}$, A_i's value of s at t is $s(\mathcal{M}_{P_i^t}) = v_i^t$ and dialogue conclusion is $c = \langle v_f, \mathit{\Phi}_{s,CS_d^t} \rangle$, where CS_d^t is the Commitment Store of d at t and $v_f = s(\mathcal{M}_{CS_d^t})$. Assume d is not a Deep Persuasion, i.e., $\exists_i \mathit{\Phi}_{s,P_i^t} \neq \mathit{\Phi}_{s,CS_d^t}$. We will consider the two cases separately: either v_f is not commonly shared by all agents, or it is.
Case 1: $\exists_i\ v_i^t \neq v_f$. First assume $v_i^t \neq \mathfrak{u}$, so $\mathit{\Phi}_{s,P_i^t} \neq \emptyset$. Then, agent A_i has either:

1a. not revealed its current belief, thus could not (properly) prove it, or
1b. uttered its current belief but failed to provide a p-proof of it,
1c. uttered its current belief and provided a proof but other agents did not vote for it,
1d. uttered its current belief and provided a proof but other agents refuted it.

(1a) $\overset{\text{def}}{\equiv}$ $\langle bel, s, v_i^t, A_i \rangle \notin QS_d^t$. Since $assertBel$ concerning dialogue topic can be uttered at any time (unless already present in QS), this contradicts our assumptions ($\lightning$) that d terminated on Impasse.

(1b) $\overset{\text{def}}{\equiv}$ $\exists_{\delta \in \mathit{\Phi}_{s,P_i^t}} DS_d^t[\delta] = \mathfrak{f}$. Consider $\neg$ (1a) $\wedge$(1b) holds. But, $\langle why, s, v_i^t, \bot \rangle \in QS_d^t$ since why can be uttered at any time by anyone, if the relevant belief is present in QS and the relevant tuple for why wasn't uttered yet. Thus, on Lemma 1, agent A_i has a relevant move consisting of all the facts and rules from $\mathit{\Phi}_{s,P_i^t}$, thus $\lightning$.

(1c) $\overset{\text{def}}{\equiv} \exists_{\delta \in \Phi_{s,P_i^t}} : \langle \delta, \sigma \rangle, \langle \neg\delta, \sigma \rangle \notin RDS_d^t$ where $\sigma \in Status$. If $\neg$(1a) $\wedge \neg$(1b) $\wedge$ (1c), then there is an agent who did not vote for entry δ. But asserting an attitude towards a piece of evidence is possible at any time (provided it is not a repetition), thus $\lightning$.

(1d) $\overset{\text{def}}{\equiv} \exists_{\delta \in \Phi_{s,P_i^t}} \delta \notin CS_d^t$. If $\neg$ (1a) $\wedge \neg$(1b) $\wedge \neg$ (1c) $\wedge$ (1d) holds, at least one rule or fact δ from A_i's p-proof of s was not accepted by others. Since A_i still regards δ as an element of p-proof at time t, it didn't comply with group decision, thus $\lightning$.

So finally we have $\neg$ (1a) $\wedge \neg$(1b) $\wedge \neg$ (1c) $\wedge \neg$ (1d), so $\forall_{\delta \in \Phi_{s,P_i^t}} \delta \in CS_d^t$ so $\Phi_{s,P_i^t} \subseteq CS_d^t$. Also $\forall_i CS_d^t \subseteq P_j^t$ (compliance). But $\exists_i : v_i^t \neq v_f = s(\mathcal{M}_{CS_d^t})$, so $P_i^t \setminus CS_d^t \cap \Phi_{s,P_i^t} \neq \emptyset$ but then $\lightning$.

Back to *Case 1* when $v_i^t = \mathfrak{u}$. Then, A_i cannot provide a proof ($\Phi_{s,P_i^t} = \emptyset$), but $v_f \neq \mathfrak{u}$, so $\Phi_{s,CS_d^t} \neq \emptyset$. Assuming A_i is compliant, $CS_d^t \subseteq P_i^t$ but since $\Phi_{s,CS_d^t} \neq \emptyset$ and $\Phi_{s,P_i^t} = \emptyset$ there is a nonempty relevant piece of $P_i^t \setminus CS_d^t \neq \emptyset$ which influences the p-proof Φ_{s,P_i^t} and which was not shared by A_i, thus $\lightning$.

Case 2.$(\forall_i \, v_i^t = v_f)$. If $v_f = \mathfrak{u}$ then $\forall_i \Phi_{s,P_i^t} = \Phi_{s,CS_d^t} = \emptyset$ trivially. Case when $v_f \neq \mathfrak{u}$ and $\exists_i \Phi_{s,P_i^t} \neq \Phi_{s,CS_d^t}$ is analogous to (1c) and (1d) above. $\lhd$

Since the common warrant may be obtained earlier than termination time t imposed by Impasse, formally Deep Persuasion covers a wider set of dialogues than persuasion ending on Impasse. Since we cannot determine at runtime whether a common warrant has been reached, in practice we use Impasse criterion and in the sequel we restrict the notion of Deep Persuasion to Deep Persuasion ending on Impasse.

Theorem 3. *Persuasion terminating on Common Opinion is a Classical Persuasion.*

Proof. Trivially from sincerity. $\lhd$

5.1 Soundness, Completeness and Convergence to Merged Outcome

The merging operator $\sum(s)$ used in analysis of persuasion properties reflects the nature of dialogue and is a consensual merge (see [11]) exploiting a voting mechanism (see Definition 10) for conflict resolution. Merging is an iterative procedure, achieved by joining p-proofs of the merge parameter s and resolving conflicts on the way. The result of merging is a 4QL program defined as follows:

$$\left(\sum_{i=1}^{n} P_i \right)(s) \overset{\text{def}}{=} \bigcup_{IT=0}^{ITMAX} \left(\bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) \right),$$

where for $IT \geq 0$, $k \in \mathbb{T}$:

- $\bigcup \Phi^{IT} \overset{\text{def}}{=} \bigcup_{i=1..n} \Phi_{s,P_i^{IT}}$,
- $ITMAX \overset{\text{def}}{=} IT : \forall_{i=1..n} : P_i^{IT+1} = P_i^{IT}$,
- $P_i^{IT+1} = P_i^{IT} \cup \bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) \setminus \bigcup_{\delta \in \bigcup \Phi^{IT}} OUT(\delta)$,
- $IN(\delta) = \{\delta[a = i], \neg\delta[b = i] : \langle a, b \rangle \in VF(\delta, n_{\mathfrak{t}}^{\delta}, n_{\mathfrak{f}}^{\delta}, n_{\mathfrak{i}}^{\delta}, n_{\mathfrak{u}}^{\delta}, n)\}$,
- $OUT(\delta) = \{\delta[a = o], \neg\delta[b = o] : \langle a, b \rangle \in VF(\delta, n_{\mathfrak{t}}^{\delta}, n_{\mathfrak{f}}^{\delta}, n_{\mathfrak{i}}^{\delta}, n_{\mathfrak{u}}^{\delta}, n)\}$
- $n_k^{\delta} = |\{i \in 1..n : \mu_{P_i^{IT}}(\delta) = k\}|$.

In each iteration, the conflicts in the union of all agents' p-proofs are resolved by voting[6], whose outcomes (sets IN and OUT) are then used to update the programs. The procedure stops naturally when agents' programs stop changing. The proof of s from such a merge is $\mathit{\Phi}_{s,(\sum_{i=1..n} P_i)(s)}$ while the value of the topic s is $s(\mathcal{M}_{(\sum_{i=1}^{n} P_i)(s)})$.

Informally, **soundness** of persuasion means that any conclusion obtained in the dialogue equals the conclusion obtained by a single agent reasoning from a merged knowledge bases of dialogue participants. On the other hand, **completeness** of persuasion means that any conclusion obtained by reasoning from **merged knowledge bases** of participants is obtainable by persuasion carried out by these agents.

Definition 15. Persuasion dialogue d on subject s is *sound* iff whenever it terminates at t with conclusion $c = \langle v_f, S\rangle$, then if $s(\mathcal{M}_{CS_d^t}) = v_f$ then $s(\mathcal{M}_{(\sum_{i=1}^{n} P_i)(s)}) = v_f$.

Definition 16. Persuasion dialogue d on subject s is *complete* iff whenever it terminates at t with conclusion $c = \langle v_f, S\rangle$, then if $s(\mathcal{M}_{(\sum_{i=1}^{n} P_i)(s)}) = v_f$ then $s(\mathcal{M}_{CS_d^t}) = v_f$.

Theorem 4. *Classical Persuasion is not sound and not complete.*

Proof. As a counterexample consider group of agents $G = \{A, B\}$ with programs $P_A = \{a :\!-\, b; b\}$ and $P_B = \{a :\!-\, \neg b; \neg b\}$ participating in dialogue d on subject a. Assume A starts with $assertBel_A^d\langle a, \mathtt{t}\rangle$. From 2 possible moves, B chooses to reply with $assertBel_B^d\langle a, \mathtt{t}\rangle$. Since $|\langle bel, a, \mathtt{t}, X\rangle \in QS_d^2 : X \in G| = |G|$, dialogue ends at $t = 2$ with a conclusion $c = \langle \mathtt{t}, \emptyset\rangle$ (CS_d^2 is empty). However, conclusion c' obtained using any merging operator VF (see Def. 10) on A and B's programs is $c' = \langle \mathtt{u}, \emptyset\rangle$. $\triangleleft$

A subset of Deep Persuasion dialogues called **Iterated Deep Persuasion** can be distinguished when additional restrictions are put on agents regarding querying dialogue stores, namely, when the below steps are repeated one after another in a loop:

A. while (exists other relevant move) do not query DS nor RDS;
B. while (exists other relevant move) do not query RDS;
C. while (exists relevant move) play move; *(i.e., query RDS)*

Theorem 5. *Iterated Deep Persuasion is sound and complete.*

Proof. In Iterated Deep Persuasion, in each iteration IT (starting with $IT = 0$):

Loop A. Agents cannot see new evidence of other agents. Thus each agent i individually (e.g. in one relevant move) reveals only its whole current p-proof $\mathit{\Phi}_{s,P_i^{IT}}$ to others (interleaved with any $assertBel$ or why locutions). So $\bigcup \mathit{\Phi}^{IT} \overset{\text{def}}{=} \bigcup_{i=1..n} \mathit{\Phi}_{s,P_i^{IT}}$ is publicly available in DS after this phase.

Loop B. Agents receive access to evidence of others (can query DS). Each agent i reveals its attitude towards any $\delta \in \bigcup \mathit{\Phi}^{IT}$ (even if not in its p-proof of s), so after this phase $\forall_{\delta \in \bigcup \mathit{\Phi}^{IT}}$ we have all $n_k^\delta = |\{i \in 1..n : \mu_{P_i^{IT}}(\delta) = k\}|$, where $k \in \mathbb{T}$. Thus, voting begins and $\forall_{\delta \in \bigcup \mathit{\Phi}^{IT}}$ we obtain the sets $IN(\delta)$ and $OUT(\delta)$ s.t.:
$IN(\delta) = \{\delta[a = i], \neg\delta[b = i] : \langle a, b\rangle \in VF(\delta, n_{\mathtt{t}}^\delta, n_{\mathtt{f}}^\delta, n_{\mathtt{i}}^\delta, n_{\mathtt{u}}^\delta, n)\}$, and
$OUT(\delta) = \{\delta[a = o], \neg\delta[b = o] : \langle a, b\rangle \in VF(\delta, n_{\mathtt{t}}^\delta, n_{\mathtt{f}}^\delta, n_{\mathtt{i}}^\delta, n_{\mathtt{u}}^\delta, n)\}$.

---

[6] Note P not $\mathit{\Phi}$ in the subscript $\mu_{P_i^{IT}}(\delta)$, since one may vote for δ even if absent from the p-proof.

After this step, $RDS^{IT+1} = RDS^{IT} \cup$

$$\bigcup_{\delta \in \bigcup \Phi^{IT}} \{\langle \delta, i \rangle : \delta \in IN(\delta)\} \cup \bigcup_{\delta \in \bigcup \Phi^{IT}} \{\langle \delta, o \rangle : \delta \in OUT(\delta)\}.$$

Loop C. Agents receive access to RDS thus eventually each agent's program is updated as follows: $$P_i^{IT+1} = P_i^{IT} \cup \bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) \setminus \bigcup_{\delta \in \bigcup \Phi^{IT}} OUT(\delta)$$

Since this is a Deep Persuasion, agents play until Impasse, i.e., when $\forall_i : P_i^{IT+1} = P_i^{IT}$ (when agents' programs stop changing, in loop C all final relevant moves are played if exist). Then, at termination time tt:

$$CS_d^{tt} = \bigcup \{\delta : \langle \delta, i \rangle \in RDS_d^{tt}\} = \bigcup_{IT=0}^{ITMAX} \bigcup_{\delta \in \bigcup \Phi^{IT}} IN(\delta) = \left(\sum_{i=1}^{n} P_i\right)(s),$$

so $\Phi_{s,(\sum_{i=1}^{n} P_i)(s)} = \Phi_{s,CS_d^{tt}}$. ◁

Theorem 6. *Deep and Classical Persuasion possibly converges to the merged outcome.*

Proof. Immediately from Thm. 5 and the fact that Classical Persuasion subsumes Deep.

6 Conclusions

We presented a formalization of multi-party, paraconsistent and paracomplete persuasion in TalkLOG, where agents can argue about beliefs with use of pieces of evidence. Classical Persuasion [1] was investigated and extended to account for more types of initial conflicts of opinion due to the 4-valued approach. Moreover we distinguished Deep Persuasion, which solves conflicts of justifications of opinions, more common in tightly-coupled groups. We succeeded to obtain a unified treatment of both dialogue types, which, in TalkLOG, are differentiated only by termination criterion.

Our model is somewhat complex as it deals with 4 dialogue stores, retaining the effects of agents' moves and resolved conflicts. Such architecture permits to achieve a protocol with public semantics. Specifically, we show that obtaining conclusion of a terminated dialogue is tractable. Our contribution advances the research on computational models of persuasion as we explicitly consider dynamics of belief revision in the course of dialogue. Moreover, we depart from the traditional notion of conflict based on inconsistency, allowing instead for a custom voting mechanism for conflict resolution.

The outcomes of TalkLOG dialogues are juxtaposed with the merged outcomes of the individual informational stances of participants. A non-trivial merge operator (inspired by [11]) exploited the same voting mechanism as in dialogue. Finally, Classical Persuasion turned out to be neither sound nor complete, as the same opinion may be justified by different, in extreme cases even antithetic justifications, which may not be discovered. Thus, we naturally reached deeper than just opinions, namely at their justifications, what led us to distinguishing a new type of dialogue: Deep Persuasion. In a special class of Deep Persuasion, i.e., Iterated Deep Persuasion, soundness and completeness was obtained (assuming again the same voting mechanism in dialogue and merge) at the price of limiting flexibility of agents' communication.

The results were obtained in a paraconsistent, nonmonotonic, multi-party and dynamic setting. Extending this research, we will provide the proof of soundness and completeness of all Deep Persuasion dialogues and investigate more specific complexity results, following our previous work on inquiry [9].

References

1. Walton, D., Krabbe, E.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany (NY) (1995)
2. Prakken, H.: Formal systems for persuasion dialogue. The Knowledge Engineering Review **21**(2) (2006) 163–188
3. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. J. Log. Comput. **13**(3) (2003) 347–376
4. Bonzon, E., Maudet, N.: On the outcomes of multiparty persuasion. In: Argumentation in Multi-Agent Systems. Volume 7543 of LNCS. Springer (2012) 86–101
5. Kontarinis, D., Bonzon, E., Maudet, N., Moraitis, P.: Regulating multiparty persuasion with bipolar arguments: Discussion and examples. In: Modles Formels de l'interaction (MFI'11). (2011)
6. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation **1**(2) (2010) 93–124
7. Takahashi, T., Sawamura, H.: A logic of multiple-valued argumentation. In: AAMAS, IEEE Computer Society (2004) 800–807
8. Vitória, A., Małuszyński, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae **97**(4) (2009) 405–438
9. Dunin-Kęplicz, B., Strachocka, A.: Tractable inquiry in information-rich environments. In: Proceedings of the IJCAI'15, AAAI Press (2015) to appear
10. Prakken, H.: Models of persuasion dialogue. In Simari, G., Rahwan, I., eds.: Argumentation in Artificial Intelligence. Springer US (2009) 281–300
11. Coste-Marquis, S., Devred, C., Konieczny, S., Lagasquie-Schiex, M., Marquis, P.: On the merging of Dung's argumentation systems. Artif. Intell. **171**(10-15) (2007) 730–753
12. Małuszyński, J., Szałas, A.: Partiality and inconsistency in agents' belief bases. In: KES-AMSTA. Volume 252 of Frontiers in Artificial Intelligence and Applications., IOS Press (2013) 3–17
13. Szałas, A.: How an agent might think. Logic J. IGPL **21**(3) (2013) 515–535
14. Małuszyński, J., Szałas, A.: Living with inconsistency and taming nonmonotonicity. In: Datalog Reloaded. Volume 6702 of LNCS., Springer-Verlag (2011) 384–398
15. Brandt, F., Conitzer, V., Endriss, U.: Computational social choice. Multiagent systems (2012) 213–283
16. FIPA: (2002) `http://www.fipa.org/`.
17. Cohen, P.R., Levesque, H.J.: Performatives in a rationally based speech act theory. In: Meeting of the Association for Computational Linguistics. (1990) 79–88
18. Fisher, M.: Representing and executing agent-based systems. In: Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages. (1994) 307–324
19. Dunin-Kęplicz, B., Strachocka, A., Szałas, A., Verbrugge, R.: Paraconsistent semantics of speech acts. Neurocomputing **151** (2015) 943–952
20. Chopra, A.K., Artikis, A., Bentahar, J., Colombetti, M., Dignum, F., Fornara, N., Jones, A.J.I., Singh, M.P., Yolum, P.: Research directions in agent communication. ACM TIST **4**(2) (2013) 20
21. Dunin-Kęplicz, B., Szałas, A., Verbrugge, R.: Tractable reasoning about group beliefs. In: EMAS 2014. Lecture Notes in Artificial Intelligence. Springer Berlin Heidelberg (2014)
22. Alferes, J.J., Brogi, A., Leite, J.A., Pereira, L.M.: Evolving logic programs. In: Proceedings of JELIA 2002. Volume 2424 of LNCS., Springer (2002) 50–61
23. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. Journal of Logic and Computation **15** (2005) 1009–1040