Perceiving Rules under Incomplete and Inconsistent Information*

Barbara Dunin-Kęplicz and 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 inconsistency is an immanent property of realistic domains, we lean towards a more pragmatic and flexible solution. Assuming that we have various disambiguation methods

* Supported by the Polish National Science Centre grants 2011/01/B/ST6/02769 and CORE 6505/B/T02/2011/40.

J. Leite et al. (Eds.): CLIMA XIV, LNAI 8143, pp. 256–272, 2013.

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 $\neg \ell \stackrel{\text{def}}{=} R(\bar{\tau})$. 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]).

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

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

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

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

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:

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

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 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 $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x \alpha(x))(L, v) = \min_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$,
  where min is the minimum w.r.t. ordering (1);
- $(\exists x \alpha(x))(L, v) = \max_{a \in \mathit{Const}} (\alpha^x_a)(L, v)$,
  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*. $\triangleleft$

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. ◁

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*. ◁

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 $\mathcal{B}^{\mathcal{E}}$. ◁

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]

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

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).

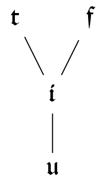

Fig. 1. Knowledge ordering $\leq_k$

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* $M_i.\ell :\!- b$ to an epistemic profile $\mathcal{E} = \{M_1, ..., M_n\}$ as follows:

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

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 $\ell :\!- b$ 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: $\mathcal{E}' = \mathcal{E} \cup \{\ell :\!- b\},\ F' = \mathcal{E}'(\mathcal{C})$.

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=$S, \ldots$) `IN`$\{\mathfrak{f}\}$ **then**
    go to `[End]`
  end if
`[Parsing]`
`[Case 1]` ▷ rule head and body recognized
  if $l \in F_R \wedge \underset{j \in 1..m,\ k \in 1..i_m}{\forall} b_{j_k} \in F_R$ **then** ▷ $l, b_{j_k} \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}$
    go to `[Evaluation]`
`[Case 2]` ▷ only rule body recognized
  else if $\underset{j \in 1..m,\ k \in 1..i_m}{\forall} b_{j_k} \in F_R$ **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 (hammer, painting, use(hammer, nail), 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 $B_{T''_{borrowed}}$):

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

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

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. Atkinson, K., Bench-Capon, T., McBurney, P.: Computational representation of practical argument. Synthese 152, 157–206 (2005)
2. Austin, J.L.: How to Do Things with Words, 2nd edn. Clarendon Press, Oxford (1975) Edited by J. O. Urmson, M. Sbisa
3. de Amo, S., Pais, M.: A paraconsistent logic approach for querying inconsistent databases. International Journal of Approximate Reasoning 46, 366–386 (2007)
4. Dignum, F., Dunin-Kęplicz, B., Verbrugge, R.: Creating collective intention through dialogue. Logic Journal of the IGPL 9, 145–158 (2001)
5. Dunin-Kęplicz, B., Strachocka, A., Szałas, A., Verbrugge, R.: Perceiving Speech Acts under Incomplete and Inconsistent Information. In: KES AMSTA. Frontiers of Artificial Intelligence and Applications, vol. 252, pp. 255–264. IOS Press (2013)
6. Dunin-Kęplicz, B., Szałas, A.: Epistemic profiles and belief structures. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 360–369. Springer, Heidelberg (2012)
7. Walther, D., Lutz, C., Wolter, F., Wooldridge, M.: ATL satisfiability is indeed EXPTIME-complete. Journal of Logic and Computation 16(6), 765–787 (2006)
8. Dziubiński, M., Verbrugge, R., Dunin-Kęplicz, B.: Complexity issues in multiagent logics. Fundamenta Informaticae 75(1-4), 239–262 (2007)
9. Fages, F.: 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. Małuszyński, J., Szałas, A.: Living with Inconsistency and Taming Nonmonotonicity. In: de Moor, O., Gottlob, G., Furche, T., Sellers, A. (eds.) Datalog 2010. LNCS, vol. 6702, pp. 384–398. Springer, Heidelberg (2011)
12. Małuszyński, J., Szałas, A.: Partiality and Inconsistency in Agents' Belief Bases. In: KES-AMSTA. Frontiers of Artificial Intelligence and Applications, vol. 252, pp. 3–17. IOS Press (2013)

13. Mascardi, V., Demergasso, D., Ancona, D.: Languages for programming BDI-style agents: an overview. In: Corradini, F., De Paoli, F., Merelli, E., Omicini, A. (eds.) WOA 2005 - Workshop From Objects to Agents, pp. 9–15 (2005)
14. Parsons, S., McBurney, P.: Argumentation-based dialogues for agent coordination. Group Decision and Negotiation 12, 415–439 (2003)
15. Prakken, H.: Formal systems for persuasion dialogue. The Knowledge Engineering Review 21(2), 163–188 (2006)
16. Searle, J.R.: Speech Acts. Cambridge University Press, Cambridge (1969)
17. Vitória, A., Małuszyński, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
18. Walton, D., Krabbe, E.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany (1995)
19. Cohen, P.R., Levesque, H.J.: Rational interaction as the basis for communication. Technical Report 433, SRI International, Menlo Park, CA (1988)
20. Kaiser, M., Dillmann, R., Rogalla, O.: Communication as the basis for learning in multi-agent systems. In: ECAI 1996 Workshop on Learning in Distributed AI Systems (1996)
21. Paglieri, F., Castelfranchi, C.: Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in MAS. In: Rahwan, I., Moraïtis, P., Reed, C. (eds.) ArgMAS 2004. LNCS (LNAI), vol. 3366, pp. 78–94. Springer, Heidelberg (2005)
22. Costantini, S.: Learning by knowledge exchange in logical agents. In: WOA 2005, Dagli (2005)
23. Mcburney, P., Parsons, S.: Tenacious Tortoises: A formalism for argument over rules of inference. In: Computational Dialectics (ECAI 2000 Workshop) (2000)
24. Singh, M.: A semantics for speech acts. Annals of Mathematics and Artificial Intelligence, pp. 47–71. Springer, Netherlands (1993) Print ISSN: 1012-2443
25. Linder, B., Hoek, W., Meyer, J.-J. C.: Actions that make you change your mind. In: KI 1995: Advances in Artificial Intelligence, vol. 98, pp. 185–196 (1995)
26. Prakken, H.: Modelling Reasoning about Evidence in Legal Procedure. In: Proceedings of the Eighth International Conference on Artificial Intelligence and Law, pp. 119–128 (2001)
27. Bench-Capon, T.J.M., Prakken, H.: Using Argument Schemes for Hypothetical Reasoning in Law. Artificial Intelligence and Law 18(2), 153–174 (2010)
28. Gabbay, D., Hunter, A.: Making Inconsistency Respectable: A Logical Framework for Inconsistency in Reasoning, Part I - A Position Paper. In: Jorrand, P., Kelemen, J. (eds.) FAIR 1991. LNCS, vol. 535, pp. 19–32. Springer, Heidelberg (1991)
29. Winslett, M.: Updating logical databases. Cambridge University Press (1990)
30. van Harmelen, F., Lifschitz, V., Porter, B.: Handbook of Knowledge Representation. Elsevier Science (2007)
31. Alferes, J.J., Leite, J.A., Pereira, L.M., Przymusinska, H., Przymusinski, T.C.: Dynamic Logic Programming. In: Procs. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning, Trento, Italy, pp. 98–109 (1998)
32. Alferes, J.J., Brogi, A., Leite, J., Moniz Pereira, L.: Evolving Logic Programs. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 50–61. Springer, Heidelberg (2002)
33. Béziau, J.-Y., Carnielli, W.A., Gabbay, D.M.: Handbook of paraconsistency. College publications (2007)
34. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press (2008)