Tractable Reasoning about Group Beliefs[*]

Barbara Dunin-Kęplicz[1], Andrzej Szałas[1,2], and Rineke Verbrugge[3]

[1] Institute of Informatics, Warsaw University, Poland
keplicz@mimuw.edu.pl
[2] Dept. of Computer and Information Science, Linköping University, Sweden
andrzej.szalas@mimuw.edu.pl
[3] Institute of Artificial Intelligence, University of Groningen, The Netherlands
L.C.Verbrugge@rug.nl

Abstract. In contemporary autonomous systems, like robotics, the need to apply group knowledge has been growing consistently with the increasing complexity of applications, especially those involving teamwork. However, classical notions of common knowledge and common belief, as well as their weaker versions, are too complex. Also, when modeling real-world situations, lack of knowledge and inconsistency of information naturally appear. Therefore, we propose a shift in perspective from reasoning in multi-modal logics to querying paraconsistent knowledge bases. This opens the possibility for exploring a new approach to group beliefs. To demonstrate expressiveness of our approach, examples of social procedures leading to complex belief structures are constructed via the use of epistemic profiles. To achieve tractability without compromising the expressiveness, as an implementation tool we choose 4QL, a four-valued rule-based query language. This permits both to tame inconsistency in individual and group beliefs and to execute the social procedures in polynomial time. Therefore, a marked improvement in efficiency has been achieved over systems such as (dynamic) epistemic logics with common knowledge and ATL, for which problems like model checking and satisfiability are PSPACE- or even EXPTIME-hard.

Keywords: Cooperation, reasoning for robotic agents, formal models of agency, knowledge representation, tractability.

1 A New Perspective on Beliefs

Classical approaches to common knowledge capture the essence of the mutuality involved in what it means to deal with common knowledge, as contrasted with distributed knowledge. According to the usual understanding, the essence of these notions is *consensus* between group participants. This is clearly visible in the notion of general knowledge E-KNOW_G (every agent in group G knows), and propagation of consensus, through iterations E-KNOW_G^k up to common knowledge C-KNOW_G, which informally can be seen as an infinitely iterated stack of general knowledge operators. This manner of building common knowledge, originating from epistemology and modal logic, captures "what every fool knows" [28, 44], [23, Chapter 2]. Indeed common knowledge is

helpful in drawing common consequences from commonly known premises, which is invaluable in creating models of others. But this comes at the price of super-polynomial complexity, causing grave problems when engineering multi-agent systems for use in time-critical situations [5, 24].

Theories dealing with various notions of knowledge and teamwork in multi-agent systems have been developed, among them the multi-modal logic TEAMLOG [23]. TEAMLOG allows to precisely model notions motivating agents to cooperate, such as collective intention, which integrates a strictly cooperative team together into a whole, and collective commitment, leading directly to team action based on a social plan that delineates how subgoals have been delegated to agents that have committed to perform them. Since teamwork occurs in various very diverse forms, it would not suffice to introduce one iron-clad notion of collective intention or collective commitment. Instead, using the expressive power of TEAMLOG, both notions should be calibrated to fit a variety of circumstances. The elements that vary from context to context are the levels of agents' awareness about the agent itself, other agents and the environment. Various forms of knowledge and beliefs constitute a fundamental layer of TEAMLOG.

As the role of group knowledge has recently evolved, it may instead be useful for participants to preserve their individual beliefs, while at the same time being a member of a larger group structure with group beliefs that govern the group's behavior. Instead of "what every fool knows", group knowledge would then tend to express synthetic information extracted from the information delivered by individuals. Thus, more so than in classical epistemic and doxastic logical approaches, there should be a clear distinction between agents' individual informational stances and the groups' ones. Consensus is not a requirement anymore, as group members do not necessarily adopt group conclusions. It suffices that during the group's lifetime they obey them.

In autonomous systems, the need to apply group knowledge has been growing with the increasing complexity of real-world applications, especially those involving cooperation or teamwork. A field that particularly expanded recently is robotics. In fact, contemporary robotics has now advanced so far that it has become necessary to investigate performance issues. Since more and more intelligent robots are able to autonomously perform sophisticated and precise maneuvers, we inevitably approach the era of strict cooperation among robots, software agents and people. Typical examples of such cooperation are emergency situations or catastrophes [2, 17, 24, 37, 47].

During robots' cooperation, an attempt to create consensus seems to be superfluous. Instead, in time-critical situations it is essential to *reduce the complexity* of both communication and reasoning. It is often too computationally costly to establish and reason about common beliefs and common knowledge. Especially when the information derives from different sources and is *imprecise*, problems arise due to the properties discussed in [21], including limited accuracy of sensors and other devices, restrictions on time and other resources, unfortunate combinations of environmental conditions, and limited reliability of physical devices. This combination of properties inevitably introduces *inconsistencies* on many different levels: in the information available to individual agents, between different agents, as well as between agents and groups and between groups and groups.

Even though in classical logical approaches, inconsistency immediately trivializes reasoning — "Ex falso sequitur quodlibet" — we intend to avoid such an effect. Robots are often sent to unknown terrains and face a need to sensibly proceed regardless of their ignorance and/or inconsistent information. This leads us to a paraconsistent approach, i.e., an approach that tolerates inconsistencies.[1] Thus, instead of fighting with inconsistencies, we treat them as first-class citizens. Typically, they need to be resolved sooner or later, depending on the situation in question, but in some reasonable cases they can even remain unresolved (see, e.g., [29]).

How to formally model such complicated situations? First of all, Dunin-Kęplicz and Szałas [19, 21] have proposed a shift in perspective: from reasoning in multi-modal systems of high complexity to querying (paraconsistent) knowledge bases. This has led to a novel formalization of complex beliefs. In order to bridge the gap between idealized logical approaches and their actual implementations, the novel notion of *epistemic profile* serves as a tool for transforming preliminary beliefs into final ones.

An epistemic profile reflects an agent's individual reasoning capabilities: it defines a schema in which an agent reasons and deals with conflicting information and ignorance. These skills are achievable by combining various forms of reasoning, including belief fusion, disambiguation of conflicting beliefs, and completion of lacking information. More formally, an epistemic profile corresponds to a function mapping finite sets of ground literals to ground literals (see Definition 3.3). As epistemic profiles can be devised analogously both on an individual and a group level, we achieve a uniform treatment of individual and group beliefs.

Various challenges occurring when building epistemic profiles can be solved with the use of 4QL, a four-valued rule-based query language designed by Małuszyński and Szałas [40, 42, 53].[2] Our approach builds on ideas underlying 4QL, which allows for negation in premises and conclusions of rules. It provides simple, yet powerful constructs (modules and external literals) [40, 41] and more general multisource formulas [53] for expressing non-monotonic rules reflecting, among others, lightweight forms of default reasoning [51], auto-epistemic reasoning [45], defeasible reasoning [49], and the local closed world assumption [27]. Importantly, 4QL enjoys tractable query computation and captures all tractable queries; this means that 4QL can express exactly those properties which can be checked in deterministic polynomial time with respect to the size of the database domain (see [41] for details). Therefore, 4QL is a natural implementation tool opening the space for a diversity of applications by providing firm foundations for paraconsistent knowledge bases used by external applications. This paper is part of a larger research program started in [18–21, 25]. The main contributions of this article are (see also Table 1):

- Providing a tractable methodology for modeling *group beliefs* that ensures a proper treatment of inconsistent or lacking information, while avoiding unwanted effects like logical omniscience;
- Implementing examples of *social procedures*, leading to complex belief structures, via the use of epistemic profiles and 4QL;

---

[1] Paraconsistency has a long tradition and is intensively investigated (see, e.g., [4]).

[2] See also http://4ql.org, which provides an open source experimental interpreter of 4QL.

Table 1. Shift in perspective on group beliefs

Traditional approaches	The new approach
"What every fool knows"	Synthetic information extracted from individuals or other groups
Holistic knowledge	Selected aspects only
Consensus	Group members not forced to adopt group conclusions: only required to obey them during the group's lifetime
Logical omniscience	Incomplete/inconsistent beliefs allowed
Monotonicity	Non-monotonic resolution of incomplete/inconsistent beliefs offered
Homogeneity (typically)	Heterogeneity: reasoning is individualized; heterogeneous information sources allowed
Reasoning intractable	Tractability: reasoning in deterministic polynomial time

- Showing how to *tame inconsistency and incompleteness* in individual and group beliefs;
- Showing that social procedures for creating group beliefs, expressed in 4QL and using lightweight forms of non-monotonic reasoning, can be executed in *deterministic polynomial time*.

In this paper we focus on belief formation rather than belief maintenance and revision. Such dynamic aspects, for which 4QL is eminently suitable, will be presented in future work.

The rest of the paper is structured as follows. Section 2 presents a robot rescue scenario to be used as running example, while Section 3 presents the logical background on belief structures, epistemic profiles and 4QL. The heart of the paper includes Section 4, which introduces methods for creating group beliefs in 4QL according to agents' and groups' epistemic profiles. Section 5 focuses on solving the problem of conflicting information at the group level. Section 6 provides a formalization of the robot rescue scenario. Section 7 discusses the influence of group beliefs on members' individual beliefs. In Section 8, we show that social procedures expressible in 4QL are in fact tractable. We end with a discussion and topics for future research in Section 9.

2   Running Example: Robot Rescue Scenario

Consider a group of robots, each equipped with a temperature sensor. In our running example, their beliefs, as hardwired by the robots' manufacturer, are expressed by the following rules:

- if $\left(temperature \leq 65^o C\right)$ then operating is safe; $\hfill (1)$
- if $\left(65^o C < temperature \leq 80^o C\right)$ then risk of damage is serious; $\hfill (2)$
- if $\left(80^o C < temperature\right)$ then it is certain that operating is impossible. $\hfill (3)$

Assume that there is fire in certain regions, resulting in a high temperature in these regions and their neighborhoods. Let a surveillance team *team* = $\{r_1, \ldots, r_k\}$ ($k > 1$) of robots be formed, whose group beliefs include the one that searching for victims is more important than preserving robots. An example of a group belief can be:

> – enter the affected region and search for victims unless it is certain
>   that operating in the region is impossible. (4)

To formalize these and related rules we shall use the following relations, where R represents regions:

- *temp*(R, T): temperature in R is T;
- *risk*(R): situation in R is risky;
- *allowed*(R): entering R is allowed (perhaps also in a risky situation);
- *search*(R): search for victims in R.

Let us emphasize that each agent (robot) is equipped with its individual knowledge base, so it has individual beliefs about these relations. We also assume that geographic information system (GIS)-based information about subregions and robots' locations is available via the following relations:

- *close*(P, R): robot P is close to R;
- *subreg*(S, R): S is a subregion of R.

We use this robot rescue scenario throughout the paper.

3   Preliminaries

In what follows we assume that domains of objects are finite and that agents' reasoning is grounded in knowledge bases rather than in arbitrary theories. That is, in reasoning we allow rules and facts and consider well-supported models only.

3.1   Language, Belief Structures and Epistemic Profiles

We view *epistemic profiles* as the general means to express a variety of strategies for belief acquisition and formation. In order to apply them here, we present a summary of some of the most important definitions from [19–21, 40, 42]. The semantical structures *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.

In a multi-agent system, 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. Observe that in this way, various perspectives of agents involved are taken into consideration and merged. Similarly, groups may be members of larger groups, perhaps containing individuals, too, etc.

Table 2. Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$ (see [40, 54])

$\wedge$	f	u	i	t		$\vee$	f	u	i	t		$\rightarrow$	f	u	i	t			$\neg$
f	f	f	f	f		f	f	u	i	t		f	t	t	t	t		f	t
u	f	u	u	u		u	u	u	i	t		u	t	t	t	t		u	u
i	f	u	i	i		i	i	i	i	t		i	f	f	t	f		i	i
t	f	u	i	t		t	t	t	t	t		t	f	f	t	t		t	f

As to the language, we use the classical first-order language over a given vocabulary without function symbols, presented in [21, 40, 53]. 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 3.1. A *literal* is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with $\bar{\tau}$ being a sequence of arguments, $\bar{\tau} \in (Const \cup Var)^k$, 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 $\neg \ell \stackrel{\text{def}}{=} R(\bar{\tau})$. ◁

Though we use classical first-order syntax, the semantics substantially differs from the classical one as truth values t, i, u, 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:

$$f < u < i < t, \tag{5}$$

as argued in [1, 40, 54]. Similarly, the semantics of quantifiers in formulas $\forall x A(x)/\exists x A(x)$ is defined using ordering (5), by taking the minimum (respectively, maximum) of the truth values of $A(a)$ for $a \in \Delta$, where Δ is the domain of x.

As a reminder from [40, 54], the truth tables for conjunction $\wedge$ and disjunction $\vee$ are defined as minimum and maximum with respect to the truth ordering, respectively. The implication $\rightarrow$ is a four-valued extension of classical implication, and is used to interpret 4QL-clauses. Whenever the body of a clause has value f or u, the truth value of the whole clause is defined to be t. This reflects our intention not to draw conclusions from false or unknown information: a clause with a body that is f or u is always satisfied, so one does not need to update its head. On the other hand, from an inconsistent body, we want to conclude that the head is also inconsistent. Thus, for a body with value i, the implication is t if the head is i, and f otherwise. If the body takes value t and the head takes value t or i, the implication as a whole is t. Note that, in contrast to classical two-valued logic, it is not the case that $\varphi \rightarrow \psi$ is equivalent to $\neg\varphi \vee \psi$, so the classical abbreviations cannot be used.

Let $v : Var \longrightarrow Const$ 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 3.2. The *truth value* $\ell(L, v)$ of a literal ℓ with respect to a set of ground literals L and valuation v, is defined by:

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

Belief structures can now be defined as in [19,21].[3] Here, the concept of an *epistemic profile* is the key abstraction involved in belief formation. If S is a set, then $\text{FIN}(S)$ represents the set of all finite subsets of S.

Definition 3.3. Let $\mathbb{C} \stackrel{\text{def}}{=} \text{FIN}(\mathcal{G}(\textit{Const}))$ be the set of all finite sets of ground literals over constants in *Const*. Then:

  – an *epistemic profile* is any function $\mathcal{E} : \text{FIN}(\mathbb{C}) \longrightarrow \mathbb{C}$;
  – by a *belief structure over epistemic profile* $\mathcal{E}$ is meant a structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$ with $\mathcal{C} \subseteq \mathbb{C}$ being a nonempty finite set of *constituents*,[4] and $F \stackrel{\text{def}}{=} \mathcal{E}(\mathcal{C})$ being the *consequent* of $\mathcal{B}^{\mathcal{E}}$. $\qquad \lhd$

Importantly, final beliefs are represented as consequents.

3.2   The 4QL Rule Language

The rule language 4QL has been introduced in [40] and further developed in [42, 53]. Beliefs in 4QL are distributed among *modules*, illustrated by the following example.

Example 3.4. Consider the scenario specified in Section 2. With each robot we associate a module containing relations '*temp*', '*risk*', '*search*'. With the group '*team*' we associate a module containing relations '*risk*', '*search*', '*allowed*'. The geographic information system module '*gis*' contains relations '*subreg*' and '*close*'. $\qquad \lhd$

The 4QL language allows for negation in premises and conclusions of rules. It is based on the four-valued logic described in Section 3.1. The semantics of 4QL is defined by *well-supported models* [40–42, 53], 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 with respect to the number of constants occurring in the module" [40, 42].

Thanks to this correspondence and the fact that 4QL captures PTIME, the constituents and consequents of Definition 3.3, being PTIME-computable, can be directly implemented as 4QL modules (see also Theorem 8.1).

---

[3] Their indeterministic version is introduced and discussed in [22].
[4] That is, a constituent is any set $C \in \mathbb{C}$.

Remark 3.5. Note that this prevents the unfortunate effects of the *logical omniscience problem* (for a survey of the problem, see, e.g., [28, 38, 44, 52]): to check whether a formula A belongs to a set of beliefs of an individual or a group, one only has to determine what is its truth value in the respective consequent. Formula A can be considered as a query to a corresponding 4QL module, so tractability is preserved. As 4QL allows to express PTIME-computable queries only, intractable/uncomputable classes of valid formulas (e.g., expressing the consequences of the Peano axioms for first-order arithmetic) cannot be expressed as valid beliefs unless explicitly added to knowledge bases.   ◁

For specifying rules and querying modules, we adapt the language of [53]. To define the language, we need the notion of multisource formulas defined as follows.

Definition 3.6. 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;
- $T \subseteq \{\mathsf{t}, \mathsf{i}, \mathsf{u}, \mathsf{f}\}$.

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:

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

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

Example 3.7. The following formula:

$$\exists S(gis.subreg(S, R) \wedge temp(S, T) \wedge T > 65) \tag{6}$$

states that there is a subregion of R with the temperature T exceeding 65. The 'gis' module stores information about subregions; the part '$gis.subreg(S, R)$' of (6) uses this module.[5] More precisely, formula (6), understood as a query, returns triples $\langle region, temperature, value \rangle$ such that the truth value of formula (6) is *value* when $R = region$ and $T = temperature$.

The formula:

$$\big(\exists S(gis.subreg(S, R) \wedge temp(S, T) \wedge T > 65)\big) \in \{\mathsf{t}, \mathsf{i}, \mathsf{u}\}. \tag{7}$$

is true when the value of formula (6) is t, i or u, and is false otherwise.   ◁

---

[5] It is assumed that formulas without a module label refer to the current module.

Definition 3.8.

- *Rules* are expressions of the form:

$$conclusion :– premises. \tag{8}$$

  where *conclusion* is a positive or negative literal and *premises* are expressed by a multisource formula.
- A *fact* is a rule with empty premises (such premises 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 $m.A \in T$. $\triangleleft$

Openness of the world is assumed, but rules can be used to close it locally or globally. Rules may be distributed among modules. Here follows an example, using the robot rescue scenario of Section 2.

Example 3.9. Consider the following rules within a module, say m, of a given robot:

$$risk(R) :– \ close(R) \wedge [\text{formula }(7) = \mathbf{t}]. \tag{9}$$
$$\neg allowed(R) :– temp(R, T) \wedge T > 80. \tag{10}$$

Rule (9) expresses the fact that region R is risky for the robot if it is close to R and formula (7) is true. Rule (10) states that the robot is not allowed to enter regions where the temperature exceeds 80^oC.

One can query module m using multisource formulas like $m.risk(R)$, $m.allowed(R)$, $m.risk(R) \in \{\mathbf{t}, \mathbf{i}\}$, etc. $\triangleleft$

4   Between Individual and Group Beliefs

Group beliefs gather conclusions of reasoning processes of the agents involved. Therefore, they are generally more synthetic than beliefs of group members, and deal with selected aspects only. If not stated differently, group beliefs prevail over individual ones. If a group belief about some aspect is missing or is inconsistent, an agent should be able to grab adequate information from its individual belief base or possibly complete it non-monotonically. These features should be reflected in the *epistemic profiles* (as discussed in Section 3.1).

4.1   Adjusting 4QL to Epistemic Profiles

To simplify formalization of epistemic profiles in 4QL, we shall identify consequents of robot r (or group of robots G) with a 4QL module having the same name r (respectively, G). For a truth value w, we write:

- $m.A = w$ to stand for $m.A \in \{w\}$;
- $m.A \neq w$ to stand for $m.A \in \{\mathbf{t}, \mathbf{i}, \mathbf{u}, \mathbf{f}\} - \{w\}$.

Although all phenomena presented in this paper are expressible in 4QL, we shall also use notation extending 4QL, yet simplifying the formalizations we need. For a group of robots $G = \{r_1, \ldots, r_k\}$ ($k \geq 1$), we introduce the following notation:

- $\exists r \in G[A(r)] \stackrel{\text{def}}{=} A(r_1) \vee \ldots \vee A(r_k)$;
- $\forall r \in G[A(r)] \stackrel{\text{def}}{=} A(r_1) \wedge \ldots \wedge A(r_k)$;
- $\#\{r \in G \mid A(r)\}$ is the number of members of G making A true (A is assumed here not to have free variables other than r); we shall also use the abbreviation $\#G \stackrel{\text{def}}{=} \#\{r \in G \mid \mathbf{t}\}$ (the number of members of G).

Example 4.1. Consider the robot rescue scenario. Typical rules for the robots can be:

$$search(R) :\!-\, team.search(R) = \mathbf{t}. \tag{11}$$

$$\neg search(R) :\!-\, temp(R, T) \wedge T > 80. \tag{12}$$

The first rule states that the robot should start searching for victims in region R if $search(R) = \mathbf{t}$ is a *team*'s belief. If the temperature excludes the possibility of robots' operation (see rule (3)), then the conclusion is $\neg search(R)$. Of course, rules (11)–(12) may lead to inconsistency when the temperature in a given region is over $80^o C$ and *team* still believes that searching that region is in order. This inconsistency can easily be resolved. If rules (11)–(12) are in a module, say m, then the robot may use a rule like:

$$\neg search(R) :\!-\, m.search(R) = \mathbf{i}. \tag{13}$$

Of course, one can define more refined solutions than *(13)*.   ◁

4.2   Establishing Group Belief

Common knowledge and its weaker approximations, such as iterated general knowledge, can be viewed as a paradigmatic form of group knowledge. However, for many applications this is too much to ask for. After all, when using standard modal logics, such as in [23, 44], the levels of iterated general beliefs harbor the risk of combinatorial explosion. Even for a group as small as three agents, $G = \{1, 2, 3\}$, we have:

$$\text{E-Bel}_G(p) \Leftrightarrow \text{Bel}(1, p) \wedge \text{Bel}(2, p) \wedge \text{Bel}(3, p); \tag{14}$$

$$\text{for } k \geq 1: \text{E-Bel}_G^{k+1}(p) \Leftrightarrow \text{E-Bel}_G(\text{E-Bel}_G^k(p)). \tag{15}$$

Observe that (15), when written in full, has 3^{k+1} conjuncts, so the complexity of building levels of general belief is exponential in the number of required levels, therefore not computable in polynomial time. Thus, for time-critical applications, one should completely change the approach to group belief.

Actually, full-fledged general and common belief is not needed for many real-world applications. The necessary shared belief state may result from agreement, some example methods of which will be listed in Section 4.3. On the other hand, the notion of *distributed knowledge* is sometimes referred to as "what a wise person would know".
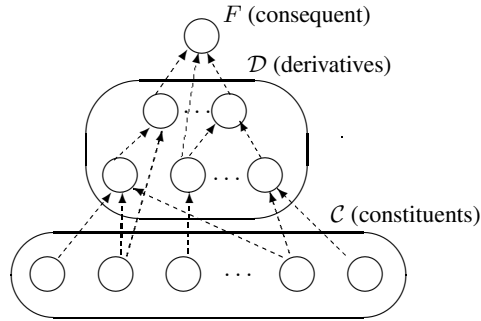
Fig. 1. Implementation framework for belief structures and epistemic profiles. Arrows indicate belief fusion processes.

This wise person would pull together the individual knowledge of group members, and would draw only classical conclusions from the combined information [44]. Distribution of reasoning is also an important feature of our approach, but why should we limit ourselves to classical reasoning only? Group knowledge may go even further than traditional distributed knowledge or belief: when starting from the same individual beliefs of the group members, a variety of reasoning methods and other techniques may lead to much more far-reaching conclusions. Epistemic profiles are introduced to encapsulate the variety of techniques used.

4.3   Building Epistemic Profiles

Creation of group beliefs takes place in the broader context of producing *derivatives*, understood as a complex process of drawing conclusions by different, temporarily existing, virtual subgroups or intermediate views [20, 21].[6] When the final *consequent* has been reached, the virtual subgroups involved may (but do not have to) disintegrate, while the consequent itself is spread among initial group members. This whole process, reflected in Figure 1 (from [20] with permission), can take place at any level of group aggregation.[7]

Using well-known heuristics, agents and groups have the possibility to complete their knowledge. Several reasoning methods can be used in the context of 4QL, as discussed in [18, 20, 25]:

- non-monotonic reasoning including the local closed-world assumption;
- default reasoning, circumscription, etc.;
- defeasible reasoning;
- methods inspired by argumentation theory.

A variety of social procedures, in combination with the reasoning methods above, may be used to establish different types of group knowledge or belief:

[6] Note that derivatives do not occur in the definition of belief structures. They are used to define epistemic profiles in a better structured and more readable manner.

[7] Though epistemic profiles may, in general, be of arbitrary complexity, in the current paper we only allow 4QL-based reasoning guaranteeing tractability.

– public announcements [16];
– different voting methods [48];
– methods involving power relations [11, 35].

Example 4.2. Assume that agents in group G vote about the truth value of the formula:

$$temp(R, T) \wedge T > 65. \tag{16}$$

A simple way to encode such majority voting is:

$$risk(R) :\!- \#\{r \in G \mid r.[(16)] = \mathbf{t}\} > \#\{r \in G \mid r.[(16)] = \mathbf{f}\}.$$

The above rule can be made more subtle, e.g., by setting:

$$risk(R) :\!- \#\{r \in G \mid r.[(16)] \in \{\mathbf{t}, \mathbf{i}\}\} > \#\{r \in G \mid r.[(16)] \in \{\mathbf{f}, \mathbf{u}\}\}.$$

Of course, such voting may be made more context-dependent by using relations other than those occurring in (16). ◁

It may be profitable to investigate the consequences of making group decisions based on voting rather than, for example, lengthy persuasion dialogues, such as those needed to establish collective intentions [13].

In appropriate circumstances, one may choose *seeing* rather than *communicating* as a method to create group belief. This can be seen as an analogy to "co-presence" [12]: by joint attention, the information is seen by everybody and everybody knows that the others in the group see this, and so on. Formally, this is more restrictive than the majority voting of the above example; for the robot rescue example such "co-presence" could follow the rule:

$$risk(R) :\!- \#\{r \in G \mid r.[(16)] = \mathbf{t}\} = \#G.$$

The relevant combination of social procedures and reasoning techniques is to be implemented as individual and group epistemic profiles by means of multisource formulas and 4QL modules.

4.4   Creating Virtual Groups

Sometimes a *virtual group* is created (among other reasons) in order to establish appropriate group beliefs. Whenever this happens, the virtual group's reasoning method has to be fixed, either implicitly or explicitly, and then represented in the virtual group's epistemic profile.

However complex the process of drawing a consequent from the constituents may be in terms of subgroups involved, at the end, the resulting consequent is seen by members of the initial group only. Analogously, in order to answer a question in daily life, you may look at Wikipedia, ask experts, and ask friends what they think about the issue. When you have finally drawn your conclusion, you often forget about the details of this process and do not necessarily communicate your final conclusion to all people

involved, but only to those who need to know. This makes the process less complex and safer from the perspective of information security and, not to forget, also more relaxed.

The next important issue is a proper organization of reasoning processes and information sharing between different groups and/or agents belonging to different groups at the same time. As in everyday life, during an agent's reasoning and activities as a member of one group, the beliefs of other groups to which the agent belongs are temporarily suspended or hidden. In this situation, the agent sees only its individual and the current group beliefs. This way, switching between groups becomes simple and computationally efficient.[8] When a group belief is formed, this does not force each member to change its individual informational stance (Section 7). Relaxing this postulate creates an important difference from the attainment of common knowledge in the modal logic framework.

5   Conflicting Information

Whenever conflicting information appears, it may be resolved on the individual or group level in a similar way. If there is no means to resolve it within a given time and other constraints, the group can resort to less resource-demanding kinds of heuristics. As to timing, there are at least three strategies:

- "Killing inconsistency at the root": to solve them as soon as possible;
- On the other extreme, "living with inconsistency": to postpone disambiguation to the last possible moment (or even forever);
- Intermediate: to solve inconsistency each time new relevant information appears.

In the sequel, we focus on techniques for resolving inconsistencies, as those are generally independent of timing strategies.

5.1   Examples of Techniques

The context of the following simple examples is a group of robots in the rescue scenario deciding on the truth value of $search(X)$, which is crucial in their decision making about whether action is needed.

Example 5.1. One can resolve potential inconsistencies using one of the following example policies.

- Search if at least one group member is convinced to do this:
$$search(R) :- \exists r \in team[r.search(R) = \mathsf{t}]. \tag{17}$$

- Search if no group member opposes:
$$search(R) :- \forall r \in team[r.search(R) \neq \mathsf{f}]. \tag{18}$$

---

[8] See also the discussion in Section 7, in particular Figure 2.

– Search if at least one group member is convinced to do this and no group member opposes:

$$search(R) :- (17) \land (18). \tag{19}$$

Of course, there are many other reasonable ways for resolving inconsistencies, some of them discussed below.   ◁

In more complex scenarios, techniques for resolving inconsistency may reflect knowledge about the application domain involving legal regulations, argumentation, or other accepted strategies, such as the social procedures on which we focus next.

5.2   Social Procedures Solving Inconsistencies

In the subsequent example cases, the robots use different procedures to resolve inconsistent information about whether an area is risky, *risk(reg)*.

Case A: peer-to-peer   Solving inconsistencies among peers may not be immediately possible. A possibility is to ignore the i-values and decide that on the group level, $risk(reg)$ is true. This solution takes the majority vote among the t and f votes only and is computationally very simple, as the following example solutions indicate.

Example 5.2. Suppose $G = \{r_1, r_2, r_3\}$ and one agent assigns value i to $risk(reg)$ while two other agents assign t. It seems reasonable that the group then considers $risk(reg)$ to be true. The following rule formalizes this approach.

$$risk(reg) :- \exists r \in G(r.risk(reg) = i) \land \#\{r \in G \mid r.risk(reg) = t\} = 2.$$

Of course, this solution may be modified in particular cases, for example, when the agent voting for i is much more reliable in estimating risk than other team members. A rule like the above can be generalized to arbitrary numbers of agents, for example "the majority among the agents who are voting t or f, assigns t to $risk(reg)$" can be formalized as

$$risk(reg) :- \#\{r \in G \mid r.risk(reg) = f\} < \#\{r \in G \mid r.risk(reg) = t\}. \qquad ◁$$

Example 5.3. Let again $G = \{r_1, r_2, r_3\}$. Now suppose two agents assign value u to $risk(reg)$, while one agent assigns t to it. What should be done with this lack of information? In case of majority voting, it seems fine to ignore the u votes and restrict to taking the majority among the t and f votes. Also for larger groups, even if there are many agents assigning u to the formula, it still makes sense to compute the majority among the t and f votes only, as done in Example 4.2.   ◁

Case B: with authority or outside expert   Let us describe several possible procedures using the framework of 4QL, in the context of the robot rescue scenario.

Procedure B1: *A group belief identified with the leader's or an expert's belief*

Suppose $expLead$ is a consequent of an expert or leader knowledge base, deciding whether certain regions R are risky. If the expert's or leader's value of $risk(R) = \mathbf{t}$, then the group value corresponds. The following rules can then be used to express *team*'s consequents as to the risk:

$$risk(R) :- expLead.risk(R) = \mathbf{t}.$$
$$\neg risk(R) :- expLead.risk(R) \in \{\mathbf{u}, \mathbf{i}, \mathbf{f}\}.$$

Procedure B2: *Conditional choice between leader, expert, and majority*

A safer choice is to use all information about $risk(R)$ based on trustworthiness:

"If there is an outside expert on $risk(R)$, then we take his decision that $risk(R) = \mathbf{t}$ as the group decision; else, if the leader's evaluation of $risk(R)$ is $\mathbf{t}$, then we take on the leader's decision as group belief; else, we cast a majority vote."

This is reflected in the following rules, where exp is a group of outside experts and $lead$ is the leader:

$$risk(R) :- \exists e \in exp[e.risk(R) = \mathbf{t}]. \tag{20}$$
$$risk(R) :- \forall e \in exp[e.risk(R) \neq \mathbf{t}] \land lead.risk(R) = \mathbf{t}. \tag{21}$$
$$\begin{aligned} risk(R) :- \ &\forall e \in exp[e.risk(R) \neq \mathbf{t}] \land lead.risk(R) \neq \mathbf{t} \land \\ &\text{'}risk(R) = \mathbf{t} \text{ wins voting'}. \end{aligned} \tag{22}$$

Note that the voting in the last line can be formalized along the lines of Example 4.2.

To infer negative conclusions as to $risk(R)$, one can add rules negating conclusions and premises of (20)–(22). For example, adding such negations in rule (20), we obtain:

$$\neg risk(R) :- \neg \exists e \in exp[e.risk(R) = \mathbf{t}].$$

One could also close the relation *risk* in various ways. If rules (20)–(22) are defined in module m, then the simplest closure can be obtained using the following rule (in a module other than m):

$$\neg risk(R) :- m.risk(R) \neq \mathbf{t}.$$

6   A Formalization of the Robot Rescue Scenario

Let us now formalize an illustrative example of an epistemic profile for the module *team* using the robot rescue scenario of Section 2. Recall that 4QL modules can be identified with sets of literals. In what follows we use this identification.

The *team*'s belief structure consists of constituents:

- consequents of each robot $r_1, \ldots, r_k$ in *team*;
- the gis module.

To define *team*'s epistemic profile, we use the following derivatives:

- *allClose*, containing the relation *risk*, calculated according to votes of all agents close to a given region;
- *safe*, containing the relation *allowed*, stating that searching a given region is allowed (no certainty of damaging robots there).

The above derivatives are used for illustration purposes only.

The module *allClose* contains, among others, the following rules:

$$risk(R) :\!- \ \#\{r \in team \mid gis.close(r, R) = \mathsf{t} \wedge r.risk(R) = \mathsf{t}\} > $$
$$\#\{r \in team \mid gis.close(r, R) = \mathsf{t} \wedge r.risk(R) \neq \mathsf{t}\}.$$
$$\neg risk(R) :\!- \ \#\{r \in team \mid gis.close(r, R) = \mathsf{t} \wedge r.risk(R) = \mathsf{t}\} \leq $$
$$\#\{r \in team \mid gis.close(r, R) = \mathsf{t} \wedge r.risk(R) \neq \mathsf{t}\}.$$

The module *safe* contains the rule:
$$\neg allowed(R) :\!- \ \exists r \in team\big(gis.close(r, R) = \mathsf{t} \wedge r.temp(R, T) = \mathsf{t} \wedge T > 80\big).$$

The *team*'s consequent can be defined, for example, by the following rules:

$$risk(R) :\!- \ allClose.risk(R). \tag{23}$$
$$\neg risk(R) :\!- \ allClose.(\neg risk(R)) \wedge safe.allowed(R) \neq \mathsf{f}. \tag{24}$$
$$search(R) :\!- \ safe.allowed(R) \neq \mathsf{f}. \tag{25}$$

Of course, robots may have individual beliefs about *risk* and $search(R)$ contradicting (23)–(25). These inconsistencies can be resolved by a rule similar to (13), concluding that a robot cannot search regions where it cannot operate without being damaged.

7 From Groups Down to Agents

Group belief may be naturally used to clarify agents' individual beliefs. For example, if for some agent r the value of P is u or i, and for group G the value became one of t, f, then generally it makes sense for r to adopt this latter truth value. Formally, this could be handled by a default rule in the agent's epistemic profile, where we distinguish between a constituent of r, denoted by c, and its consequent, denoted by r:

if $c.P \in \{\mathsf{i}, \mathsf{u}\}$ and $G.P \in \{\mathsf{t}, \mathsf{f}\}$ (prerequisite) and it is consistent that "special situation (S) does not occur" (justification), then $r.P$ becomes $G.P$.

This is achieved in 4QL by placing the following rules in the module implementing consequent r:

$$P :\!- \ c.P \in \{\mathsf{i}, \mathsf{u}\} \wedge G.P = \mathsf{t} \wedge S \in \{\mathsf{f}, \mathsf{u}\}.$$
$$\neg P :\!- \ c.P \in \{\mathsf{i}, \mathsf{u}\} \wedge G.P = \mathsf{f} \wedge S \in \{\mathsf{f}, \mathsf{u}\}.$$

This way, coherence of knowledge can be maintained. The process does call for calculating a new well-supported model. Such downward reflection is useful when the group decides about critical situations. Then each individual should follow this.

When a decision is not life-critical, different opinions remain possible. For example, when a jury decides that the Best Paper Prize should be given to A while an individual jury member would have preferred B, (s)he can keep her/his opinion while the group decision stands. Similarly when a program chair decides that a certain paper is acceptable for the proceedings, individual program committee members do not need to agree to the group decision. The mode of adaptation to group beliefs needs to be included in everyone's epistemic profile. This real-world model of the information flow between a group and its individual members fits to many contexts better than common knowledge.

While 4QL does not allow for circular dependencies between modules, it is important to note that there may be loops in managing beliefs when circular dependencies among individuals and groups occur in applications, for example, due to the used updating policies. However, it is the responsibility of application designers to avoid such loops. The following example illustrates this issue.

Example 7.1. Consider a module m being a constituent of a group belief structure with consequent G. Let m consist only of facts of the form:

$$s(a), \neg s(b), s(c), \ldots,$$

and let G consist of the rule:

$$r(X) :- \; m.s(X) = \mathsf{t}. \tag{26}$$

Suppose further that an external application implements the following policy of updates:

$$m.s(X) \text{ becomes false whenever } G.r(X) \text{ becomes true;} \tag{27}$$
$$m.s(X) \text{ becomes true whenever } G.r(X) \text{ becomes unknown.} \tag{28}$$

If, at some point, a fact $s(d)$ becomes true in m then rule (26) makes $G.r(d)$ true. However, when $G.r(d)$ becomes true, the update (27) makes $m.s(d)$ false which, in turn, makes $G.r(d)$ unknown (the premise of rule (26) becomes false). According to (28), whenever $G.r(d)$ becomes unknown, $m.s(d)$ becomes true, and a loop occurs. However, this loop is caused by the design of the update policy, being out of the scope of 4QL itself. ◁

To avoid loops, no circularity should be allowed. To achieve non-circularity, one may equip each agent with several belief structures:

- the main one modeling the agent's beliefs;
- a separate belief structure associated with each group the agent belongs to;
- other structures, when needed.

Figure 2 shows an example of such an architecture. Agent A has its "main" belief structure B with consequent F, which becomes a constituent of belief structures $G_1, \ldots, G_k$
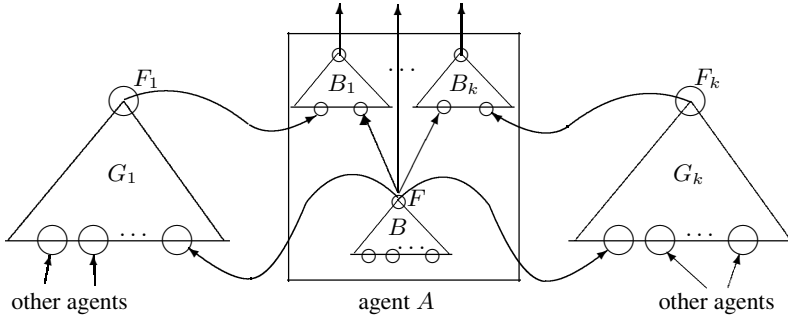
Fig. 2. Architecture of groups and agents. Arrows indicate the use of consequents of belief structures as constituents of other belief structures.

of groups A belongs to. For every group belief structure G_i $(1 \leq i \leq k)$ there is an associated belief structure B_i within A, with constituents F and F_i. Each belief structure B_i exists as long as A is a member of group G_i. This way agent A can easily switch between belief structures according to the role it plays at a given moment. When B_i is being deleted, belief structure B may be updated using information from B_i, but circularity is avoided.

8   Complexity

Consider a static situation without knowledge base updates. Thus, we have a snapshot of a system consisting of, say, k individuals and n groups, each of them computing its consequents according to its epistemic profile (Definition 3.3). Since data complexity of 4QL is PTime and 4QL captures PTime (see [41,42]), we have the following result, where, as usually, finite domains are assumed.

Theorem 8.1. *Assume that the number of constituents of each individual as well as the number of belief structures associated with each individual/group is bounded by a constant. Let k be the number of agents and let n be the number of groups under consideration.*

- *If each constituent and epistemic profile involved is implemented in 4QL, then the complexity of computing them all is $O\big((k{+}n){*}p(|Const|)\big)$, where p is a polynomial and Const is the set of constants occurring in constituents and epistemic profiles.*
- *Every epistemic profile/belief structure computable in deterministic polynomial time (PTIME with respect to data complexity) can be expressed in 4QL (assuming linear ordering on Const is given).*   ◁

Note that in this way, tractability is achieved. Though complexity depends on k and n, these parameters reflect the numbers of individuals and groups involved in a given mission. Such individuals and groups must have been generated somehow, so we can safely assume the existence of computational capacity to handle them.

If system dynamics is considered, Theorem 8.1 guarantees that every time updates need to be performed, they can be done in deterministic polynomial time. In fact, the role of 4QL is to provide firm foundations for knowledge bases used by applications external to 4QL. The application's behavior and complexity are controlled by its designers (see also Example 7.1).

Note that there is a price to pay for tractability of our approach. Namely, we neither allow disjunctive facts nor disjunctive conclusions of rules, so without additional constructs we cannot express unrestricted disjunctive reasoning unless PTime =NPTime. On the other hand, all tractable reasoning schemata are captured by our framework, so all tractable forms of disjunctive reasoning are expressible, too, though not necessarily directly.

9   Discussion and Conclusions

In the current literature on knowledge and beliefs, modal logic-based approaches are dominant. Even though they suit very well to idealized epistemic theories, they are hardly applicable to real-world complex scenarios. In order to apply them one needs to use some restricted versions of modal logics, like in [36, 46]. However, neither of these approaches deals with *inconsistent* knowledge bases and [36] deals solely with limited forms of *incomplete* multi-agent knowledge. In contrast, in the current paper we offer a novel approach to group beliefs, intended to bridge the gap between theory and applications.

We also introduce a variety of social procedures for creating group beliefs within a paraconsistent four-valued framework offered by 4QL, allowing for tractable reasoning. Importantly, our approach does not share unwanted omniscience effects like closure under logical consequence or irrelevant belief handling.

To the best of our knowledge, a paraconsistent approach to beliefs has so far mainly been pursued in the context of belief revision [43, 50], not the creation of group beliefs. These other approaches substantially differ from ours. Their models are based on criteria and rationality indexes [50] or on relevant logic [43].

Accepting four rather than two logical values considerably simplifies our approach where one is not forced to find general embeddings of $\{t, i, u, f\}$ into $\{t, f\}$ that would work in all considered contexts. Instead, we offer a framework in which such embeddings can much more easily be obtained either totally or partially, or even avoided altogether, in a highly context- and user-dependent manner. To our knowledge, such flexibility, expressiveness and at the same time tractability has not been achieved before.

We have taken into account that agents are heterogeneous in the ways that they reason; this in contrast to classical epistemic logics, which view agents as if they were homogeneous; a recent exception is the work by Liu [39]. Agents' reasoning patterns may differ significantly, which is reflected in the epistemic profiles of individual agents as well as of different (sub-)groups. Another approach to heterogeneity of information sources is proposed in [6, 10, 31, 32], where multi-context systems are considered. Inconsistencies in multi-context systems are addressed, e.g., in [7, 26]. To fuse knowledge from various contexts, bridge rules are used. However, the associated reasoning problems are typically of high complexity [7].

Developed from logic programming, answer set programming has been used to formalize qualitative decision making in individual agents [9, 30]. However, to decide whether a given program has some answer set is NP-complete already in the propositional case while non-grounded programs have exponentially higher complexity [8]. In a multi-agent epistemic context Baral et al. [3] provide an answer set programming approach to group beliefs; in their approach, reasoning problems such as the muddy children puzzle, however, inherit their high complexity from modal logic. In addition, like the modal approaches discussed above, the above-mentioned answer set programming approaches allow only consistent models and do not provide means for disambiguating inconsistencies.

There are also many techniques for resolving inconsistencies other than those presented and discussed in our paper. In particular, one could apply techniques known from defeasible reasoning [49]. One could also think of applying Boolean games in which control over specific literals is assigned to different agents. For some examples from the extensive literature, we refer to [11, 33, 35, 48].

We have also proposed some extensions to 4QL, allowing one to express a rich repertoire of combinations of social procedures with non-monotonic reasoning techniques and inconsistency disambiguation, based on the possibilities of 4QL. Although these extensions can be expressed in "pure" 4QL, we have achieved their substantial simplification here, which also is a novel contribution.

We have represented epistemic profiles, belief structures and social procedures for creating group belief in 4QL, discussing a number of example procedures of increasing intricacy. Theorem 8.1 then shows that all these aspects can be executed in polynomial time. This is a marked improvement over some of the most well-known logics for multi-agent systems. More precisely, for modal logics incorporating common knowledge or common belief, model checking is PSPACE-complete, while the satisfiability problem is EXPTIME-complete [16, 23, 44]. For logics of propositional control and coalition logics, both model checking and satisfiability are PSPACE-complete [35]. Finally, for alternating-time temporal logic (ATL), both model checking and satisfiability are even EXPTIME-complete [34, 55]. In real-time applications like time-critical teamwork, the advantages of using a tractable approach such as the one advocated here are essential.

In future work, we will also apply our approach to finding tractable solutions for classical puzzles in epistemic logic, such as the wise men, the muddy children, and the sum and product puzzle. These are classically formalized and solved by constructing (dynamic) epistemic models, which are often huge in terms of the input [14, 15, 28, 44].

This paper is part of a larger research program. Here, we focus on belief formation in heterogeneous groups, while dynamical aspects, such as maintenance of group beliefs and belief revision, are left to future research. A general problem in robotics is how the activities of different groups dovetail and interleave together. This needs to be smartly organized to allow agents to smoothly switch between activities in different groups. While the focus of this paper is agents' *reasoning* via individual and group epistemic profiles, in future work we will discuss the organizational part of group activities.

References

1. de Amo, S., Pais, M.: A paraconsistent logic approach for querying inconsistent databases. International Journal of Approximate Reasoning 46, 366–386 (2007)
2. Balakirsky, S., et al.: Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup Rescue. Journal of Field Robotics 24(11-12), 943–967 (2007)
3. Baral, C., Gelfond, G., Son, T.C., Pontelli, E.: Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 259–266. International Foundation for Autonomous Agents and Multiagent Systems (2010)
4. Bézieau, J.J., Carnielli, W., Gabbay, D. (eds.): Handbook of Paraconsistency. College Publications (2007)
5. Bordini, R., Dastani, M., Dix, J., El Fallah-Seghrouchni, A. (eds.): Multi-Agent Programming: Languages, Platforms and Applications. Springer (2009)
6. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence, pp. 385–390. AAAI Press (2007)
7. Brewka, G., Eiter, T., Fink, M., Weinzierl, A.: Managed multi-context systems. In: Walsh, T. (ed.) Proc. of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, pp. 786–791. IJCAI/AAAI (2011)
8. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Commun. ACM 54(12), 92–103 (2011)
9. Brewka, G.: Answer sets and qualitative decision making. Synthese 146(1-2), 171–187 (2005)
10. Casali, A., Godo, L., Sierra, C.: A language for the execution of graded BDI agents. Logic Journal of the IGPL 21(3), 332–354 (2013)
11. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory. Morgan & Claypool Publishers (2011)
12. Clark, H., Marshall, C.: Definite reference and mutual knowledge. In: Joshi, A., Webber, B., Sag, I. (eds.) Elements of Discourse Understanding, pp. 10–63. Cambridge University Press (1981)
13. Dignum, F., Dunin-Kęplicz, B., Verbrugge, R.: Creating collective intention through dialogue. Logic Journal of the IGPL 9, 145–158 (2001)
14. Ditmarsch, H.P.v., Ruan, J., Verbrugge, L.C.: Model checking sum and product. In: Zhang, S., Jarvis, R.A. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 790–795. Springer, Heidelberg (2005)
15. van Ditmarsch, H.P., Ruan, J., Verbrugge, R.: Sum and product in dynamic epistemic logic. Journal of Logic and Computation 18(4), 563–588 (2008)
16. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, vol. 337. Springer (2007)
17. Doherty, P., Heintz, F., Kvarnström, J.: High-level mission specification and planning for collaborative unmanned aircraft systems using delegation. Unmanned Systems 1(1), 75–119 (2013)
18. Dunin-Kęplicz, B., Strachocka, A., Szałas, A., Verbrugge, R.: A paraconsistent approach to speech acts. In: Proc. Workshop on Argumentation in Multi-Agent Systems, pp. 59–78. IFAAMAS (2012)
19. 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)

20. Dunin-Kęplicz, B., Szałas, A.: Paraconsistent distributed belief fusion. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) Intelligent Distributed Computing VI. SCI, vol. 446, pp. 56–69. Springer, Heidelberg (2013)
21. Dunin-Kęplicz, B., Szałas, A.: Taming complex beliefs. In: Nguyen, N.T. (ed.) Transactions on CCI XI. LNCS, vol. 8065, pp. 1–21. Springer, Heidelberg (2013)
22. Dunin-Kęplicz, B., Szałas, A.: Indeterministic belief structures. In: Jezic, G., Kusek, M., Lovrek, I., Howlett, R. J., Jain, L.C. (eds.) Agent and Multi-Agent Systems: Technologies and Applications. AISC, vol. 296, pp. 57–66. Springer, Heidelberg (2014)
23. Dunin-Kęplicz, B., Verbrugge, R.: Teamwork in Multi-Agent Systems: A Formal Approach. Wiley (2010)
24. Dunin-Kęplicz, B., Verbrugge, R., Slizak, M.: TEAMLOG in action: A case study in teamwork. Comput. Sci. Inf. Syst. 7(3), 569–595 (2010)
25. Dunin-Kęplicz, B., Strachocka, A.: Perceiving rules under incomplete and inconsistent information. In: Leite, J., Son, T.C., Torroni, P., van der Torre, L., Woltran, S. (eds.) CLIMA XIV 2013. LNCS (LNAI), vol. 8143, pp. 256–272. Springer, Heidelberg (2013)
26. Eiter, T., Fink, M., Schüller, P.: Approximations for explanations of inconsistency in partially known multi-context systems. In: Delgrande, J.P., Faber, W. (eds.) LPNMR 2011. LNCS (LNAI), vol. 6645, pp. 107–119. Springer, Heidelberg (2011)
27. Etzioni, O., Golden, K., Weld, D.: Sound and efficient closed-world reasoning for planning. Artificial Intelligence 89, 113–148 (1997)
28. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press (1995)
29. Gabbay, D., Hunter, A.: Making inconsistency respectable: A logical framework for inconsistency in reasoning. In: Jorrand, P., Kelemen, J. (eds.) FAIR 1991. LNCS, vol. 535, pp. 19–32. Springer, Heidelberg (1991)
30. Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents - The Answer-Set Programming Approach. Cambridge University Press (2014)
31. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: How we can do without modal logics. Artificial Intelligence 65(1), 29–70 (1994)
32. Giunchiglia, F., Serafini, L., Giunchiglia, E., Frixione, M.: Non-omniscient belief as context-based reasoning. In: IJCAI, vol. 93, pp. 9206–9203 (1993)
33. Harrenstein, P., van der Hoek, W., Meyer, J.J., Witteveen, C.: Boolean games. In: van Benthem, J. (ed.) Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 287–298. Morgan Kaufmann Publishers Inc. (2001)
34. van der Hoek, W., Lomuscio, A., Wooldridge, M.: On the complexity of practical ATL model checking. In: Proc. of the 5th AAMAS, pp. 201–208. ACM (2006)
35. van der Hoek, W., Wooldridge, M.: On the logic of cooperation and propositional control. Artificial Intelligence 164(1-2), 81–119 (2005)
36. Lakemeyer, G., Lespérance, Y.: Efficient reasoning in multiagent epistemic logics. In: De Raedt, L., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P. (eds.) Proc. of ECAI 2012 - 20th European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 498–503. IOS Press (2012)
37. Landén, D., Heintz, F., Doherty, P.: Complex task allocation in mixed-initiative delegation: A UAV case study. In: Desai, N., Liu, A., Winikoff, M. (eds.) PRIMA 2010. LNCS, vol. 7057, pp. 288–303. Springer, Heidelberg (2012)
38. Levesque, H.J.: A logic of implicit and explicit belief. In: Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI 1984), pp. 198–202 (1984)
39. Liu, F.: Diversity of agents. In: Agotnes, T., Alechina, N. (eds.) Proc. of the ESSLLI Workshop on Resource-bounded Agents, pp. 88–98 (2006)
40. 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)

41. Małuszyński, J., Szałas, A.: Logical foundations and complexity of 4QL, a query language with unrestricted negation. Journal of Applied Non-Classical Logics 21(2), 211–232 (2011)
42. Małuszyński, J., Szałas, A.: Partiality and inconsistency in agents' belief bases. In: Barbucha, D., et al. (eds.) Proc. KES-AMSTA. Frontiers of Artificial Intelligence and Applications, vol. 252, pp. 3–17. IOS Press (2011)
43. Mares, E.: A paraconsistent theory of belief revision. Erkenntnis 56, 229–246 (2002)
44. Meyer, J.J.C., van der Hoek, W.: Epistemic Logic for Computer Science and Artificial Intelligence. Cambridge University Press (1995)
45. Moore, R.: Possible-world semantics for autoepistemic logic. In: Proc. 1st Nonmonotonic Reasoning Workshop, pp. 344–354 (1984)
46. Nguyen, L.A.: On modal deductive databases. In: Eder, J., Haav, H.-M., Kalja, A., Penjam, J. (eds.) ADBIS 2005. LNCS, vol. 3631, pp. 43–57. Springer, Heidelberg (2005)
47. Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., Burion, S.: Human-robot teaming for search and rescue. IEEE Pervasive Computing 4(1), 72–79 (2005)
48. Nurmi, H.: Voting Procedures under Uncertainty. Springer (2002)
49. Nute, D.: Defeasible logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, pp. 353–395. Oxford University Press (1994)
50. Priest, G.: Paraconsistent belief revision. Theoria 67, 214–228 (2001)
51. Reiter, R.: A logic for default reasoning. Artificial Intelligence Journal 13, 81–132 (1980)
52. Sim, K.: Epistemic logic and logical omniscience: A survey. International Journal of Intelligent Systems 12, 57–81 (1997)
53. Szałas, A.: How an agent might think. Logic Journal of the IGPL 21(3), 515–535 (2013)
54. Vitória, A., Małuszyński, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
55. Walther, D., Lutz, C., Wolter, F., Wooldridge, M.: ATL satisfiability is indeed EXPTIME-complete. Journal of Logic and Computation 16(6), 765–787 (2006)