Belief Shadowing*

Łukasz Białek[1], Barbara Dunin-Kęplicz[1], and Andrzej Szałas[1,2]

[1] Institute of Informatics
University of Warsaw, Poland
[2] Department of Computer and Information Science
Linköping University, Sweden
{bialek,keplicz,andrzej.szalas}@mimuw.edu.pl

Abstract. Adapting beliefs to new circumstances, like belief change, update, revision or merging, typically requires deep and/or complex adjustments of belief bases even when adaptations happen to be transient. We present a novel, lightweight and tractable approach to a new kind of beliefs' interference which we call *belief shadowing*. Put simply, it is a transient swap of beliefs when part of one belief base is to be shadowed by another belief base representing new observations and/or beliefs of superior agents/teams. In this case no changes to belief bases are needed. This substantially improves the performance of systems based on doxastic reasoning. We ensure tractability of our formal framework, what makes it suitable for real-world applications.
The presented approach is based on a carefully chosen four-valued paraconsistent logic with truth values representing truth, falsity, incompleteness and inconsistency. Moreover, potentially undesired or forbidden conclusions are prevented by integrity constrains together with their shadowing machinery.
As an implementation environment we use $4QL^{Bel}$, a recently developed four-valued query language based on the same underlying logic and providing necessary reasoning tools. Importantly, the shadowing techniques are general enough to be embedded in any reasoning environment addressing related phenomena.

1 A New Perspective on Belief Change

When agents act in dynamic environments, belief change/revision/update/merging is inevitable, creating a multitude of problems of theoretical and applied nature [7,25,33]. In the case of group beliefs, like in teamwork, the situation becomes even more complex [11]. In real-world applications, beliefs are contextual, and affected socially, psychologically and emotionally. Some, like "do not harm", are hardly mutable but others, like "avoid slippery surfaces", meant as an indication, are flexible. In fact, known theories of belief update/change/revision/merging do not distinguish between the rigid and transient beliefs. However, in everyday activities we temporarily adjust our beliefs to specific situations with no intention to change them radically. Such a shallow change, not requiring a deeper revision, has not been addressed in the literature.

Our research is devoted to a belief change method inspired by a discussion in [9] where it is pointed out that:

* Supported by the Polish National Science Centre grant 2015/19/B/ST6/02589.

> "In contrast to many existing approaches, we do not assume that an agent entering a group changes its beliefs. However, group beliefs prevail over individual ones [...] When the group is dismissed, agents continue to act according to their individual beliefs. These can be revised to reflect information acquired during cooperation."

When beliefs are flexible or changing frequently, it hardly makes sense to adjust the entire belief base accordingly. The better choice is to override them for the time being. Technically, the kind of belief interference we address here is a swap of beliefs, potentially transient, which we call *belief shadowing*. Given two belief bases, it may turn out that one of them is more important or up to date and, therefore, shadowing the contents of respective parts of the other one. A swap from individual to group beliefs and then perhaps back to individual ones is a typical example of shadowing beliefs when the cooperation is completed.

The ability of shadowing rather than updating, revising or merging beliefs can result in a substantial improvement of the performance of agent systems using doxastic reasoning which is particularly important from the point of view of systems' engineering.

As no belief revision is required, belief shadowing may introduce inconsistencies. Moreover, information is frequently missing, so both paraconsistent and paracomplete reasoning is needed. Such forms of reasoning and their applications are discussed in many contexts, e.g., in [4,5] and references there. Classically, to model beliefs, modal logics equipped with Kripke-like semantics are used [11,12,32,43]. However, an idealized modal notion of beliefs does not fit contemporary agent systems, where one needs to robustly cope with incomplete and inconsistent information about environments in the spirit of [20]:

> "Inconsistency robustness is information system performance in the face of continually pervasive inconsistencies – a shift from the previously dominant paradigms of inconsistency denial and inconsistency elimination attempting to sweep them under the rug. Inconsistency robustness is a both an observed phenomenon and a desired feature: [...] an observed phenomenon because large information-systems are required to operate in an environment of pervasive inconsistency. [...] a desired feature because we need to improve the performance of large information system."

Belief bases represent snapshots of the environment and the agents' mindsets, both evolving over time. In AI systems this evolution should be supervised especially when rules are machine learned or data mined in a human-free manner. In order to supervise the contents of belief bases we decided to introduce integrity constraints. Though they are well-known in database systems, their shadowing is novel, pertaining admissible modes of behavior at desired abstraction levels, in a semantically coherent manner.

A coherent, tractable and comprehensive framework to belief shadowing that we aim for, amounts to:

- introducing a lightweight belief shadowing framework;
- introducing integrity constraints and their shadowing;
- providing a tractable reasoning engine allowing one to apply these constructs in pragmatic applications.

The belief shadowing framework is *lightweight* in the sense that the shadowing operator is efficient and does not require changes in the belief bases involved as well as agents' familiarity with details of other agents' belief bases. The former requirement is crucial for systems' performance and for efficient swapping between contexts in which different shadowings apply. The later is vital in cooperation/teamwork. To our best knowledge no other research realizes these goals: the presented solution is original and general enough to be embedded in many programming frameworks. As a computational engine for belief bases we adopt and extend the $4QL^{Bel}$ four-valued rule language, recently developed in [6].

The rest of the paper is structured as follows. Section 2 introduces a scenario used to illustrate the most important and novel features of our approach. Next, Section 3 presents belief bases and discusses their role. Then, Section 4 presents the underlying logic and Section 5 briefly recapitulates the rule-based language $4QL^{Bel}$. In Section 6 we extend $4QL^{Bel}$ to $4QL^{Bel+}$ by adding constraints. Section 7 introduces the shadowing operator formally. Section 8 presents properties of shadowing, in particular its complexity. In Section 9 we discuss related work. Finally, Section 10 concludes the paper.

2 An Emergency Room Scenario

To illustrate the approach, we shall consider an ER (Emergency Room) service which specializes in handling emergency situations. ER is usually operated by several emergency physicians, whose main role is to deliver a professional treatment. As a common practice emergency physicians consult a therapy with other specialists. Simple cases are dealt with internally or after a single consultation. More difficult ones may require gathering an MDM (i.e., multidisciplinary meeting). While MDM participants may propose a variety of treatments, the chosen one prevails and is applied. In terms of different beliefs this means that the physician's beliefs may be defeated, though not necessarily revised. On the other hand, patients naturally follow their individual beliefs. Specifically, they may concern rejections of various treatments, like those violating their religious rules. The refusal of blood transfusion or organ transplants is a typical case. Also, according to legal regulations valid in many countries, patients may refuse life-sustaining treatments what, on the other hand, may be obligatory for medical staff.

The main goal of an emergency physician on duty, Mark, is to apply necessary treatments to patients brought to the ER. Mark can either decide on his own about the treatment or call an MDM. Finally, the selected treatment may be unacceptable to the patient. We will show that belief shadowing is useful in modeling such situations.

3 Belief Bases

In many papers, e.g., related to the AGM theory of belief revision (for survey see [33]),[3] a belief base consists of a set of formulas of the underlying logic, not necessarily closed on consequences. In this research, a belief base may consist of more than one such

[3] AGM is an acronym referring to names of originators of the theory: C. Alchourrón, P. Gärdenfors and D. Makinson [2].

a set, each representing a feasible world state. Rather than the traditional consequence relation, we use querying machinery assigning truth values to the results, reflecting the contents of the queried belief base.

The idea of multiple sets of ground literals is close in spirit to Kripke structures. Belief bases represent multiple alternative and/or complementary views on the world. As a paraconsistent and paracomplete four-valued logic, involving truth values $\mathfrak{t}$ (true), $\mathfrak{f}$ (false), $\mathfrak{i}$ (inconsistent) and $\mathfrak{u}$ (unknown), is used, a belief base combines world states into one compact structure, capable of storing beliefs originating from nondeterministic environments. Belief bases are systems' passive components reacting on requests and queries via a suitable query processing engine. The presented ideas can be implemented in many modern belief base systems robust to inconsistencies.

In order to formally define belief bases, we extend the definition of [9,10] by assuming that constraints are their inherent parts. Let *Const* be a fixed finite set of constants, *Var* be a fixed finite set of variables and Rel be a fixed finite set of relation symbols. By a *positive literal* we understand an expression of the form $r(\bar{e})$, where $r \in Rel$ and $\bar{e}$ is a vector consisting of variables and/or constants. A *negative literal* is an expression of the form $\neg\ell$, where ℓ is a positive literal. Literals without variables are called *ground*. We always identify $\neg\neg\ell$ with ℓ. In the rest of the paper we sometimes use $3i$ as an abbreviation for *incomplete* and/or *inconsistent* information. In particular, by $3i$-*worlds* we shall understand finite sets of ground literals with all constants belonging to *Const*.

Definition 1. *By a* belief base over a set of constants *Const we understand any pair* $\mathcal{B} = \langle \Delta, \mathcal{C} \rangle$ *consisting of:*

- $\Delta = \{M_1, \ldots, M_k\}$, *where* $k \geq 1$ *and for* $i = 1, \ldots, k$, M_i *is a* $3i$*-world;*
- $\mathcal{C}$, *being finite set of (universally closed) formulas of the underlying logic, which are true in* Δ.[4]

Formulas from the set $\mathcal{C}$ *are called* constraints *of* $\mathcal{B}$. *If* $k = 1$ *then* $\mathcal{B}$ *is called* deterministic *otherwise it is called* indeterministic. ◁

Each M_i in a belief base represents a possible or complementary (perhaps incomplete and/or inconsistent) view of the world. In the ER scenario Mark's beliefs can be represented by a belief base $\mathcal{B}_M = \langle \Delta_M, \mathcal{C}_M \rangle$ with Δ_M containing several modules with his initial observations, results of medical tests and measurements, etc. These observations lead to alternative diagnoses which, in turn, may result in alternative treatments. Therefore, Δ_M includes a module `data` containing data coming from various sources, like Mark's observations, measurement and test results, patient medical files (if accessible), etc., and a module `diagnosed`, containing facts as to the patient's illness severity, impairment level, permission for overall treatment and separate permissions for specific treatments ($\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{u}$). Mark's constraints, $\mathcal{C}_M$, may contain statements like:

> if patient's life is not at risk and he/she is conscious, his/her permissions regarding treatments are obeyed.

[4] A formal definition of the underlying logic as well as of the truth value of a formula in Δ is given in Section 4.

As a result of an MDM, some of Mark's beliefs and constraints may be shadowed by beliefs and constraints of others. Moreover, Mark can disregard some beliefs of his patients if they are hazardous to their health and lives.

Constructions presented in the sequel provide means for formal and executable specifications of a variety of scenarios. Belief shadowing deliver means for clear modules' specification and the detection of integrity constraints' violations.

4 A Logic of Beliefs

In the syntax of the underlying logic we assume truth constants $\mathfrak{t}, \mathfrak{f}, \mathfrak{i}$ and $\mathfrak{u}$, propositional connectives $\neg, \vee, \wedge, \rightarrow$, quantifiers $\forall, \exists$, operators $A \in T$, $A = t$, where A is a formula, $T \subseteq \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}, \mathfrak{u}\}$, $t \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}, \mathfrak{u}\}$ and belief operator $\mathrm{Bel}_{\mathcal{B}}\big(A\big)$ where $\mathcal{B}$ is a belief base. To define semantics of the logic let us start with *truth ordering* on truth values, denoted by $\leq_t$, being the reflexive and transitive closure of ordering: $\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}$.[5] For $t, t_1, t_2 \in \{\mathfrak{f}, \mathfrak{u}, \mathfrak{i}, \mathfrak{t}\}$, the semantics of $\neg, \wedge, \vee$ is given by:

$$\neg\mathfrak{f} \stackrel{\text{def}}{=} \mathfrak{t},\ \neg\mathfrak{u} \stackrel{\text{def}}{=} \mathfrak{u},\ \neg\mathfrak{i} \stackrel{\text{def}}{=} \mathfrak{i},\ \neg\mathfrak{t} \stackrel{\text{def}}{=} \mathfrak{f}; \tag{1}$$

$$t_1 \wedge t_2 \stackrel{\text{def}}{=} \min\{t_1, t_2\};\ \ t_1 \vee t_2 \stackrel{\text{def}}{=} \max\{t_1, t_2\}; \tag{2}$$

$$\mathrm{Bel}_{\mathcal{B}}\big(t\big) \stackrel{\text{def}}{=} t, \tag{3}$$

where $\min, \max$ are the minimum and maximum wrt $\leq_t$. The *truth value* of a literal ℓ wrt a $3i$-world L and an assignment v : *Var* $\longrightarrow$ *Const*, denoted by $\ell(L, v)$, is defined as follows, where $v(\ell)$ denotes the ground literal obtained from ℓ by substituting all occurrences of x in ℓ by $v(x)$:

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

The above definition is extended to all formulas by setting:

$$\begin{aligned}
&t(L, v) \stackrel{\text{def}}{=} t \text{ for } t \in \{\mathfrak{f}, \mathfrak{u}, \mathfrak{i}, \mathfrak{t}\};\\
&(\neg A)(L, v) \stackrel{\text{def}}{=} \neg(A(L, v));\\
&(A \odot B)(L, v) \stackrel{\text{def}}{=} A(L, v) \odot B(L, v), \text{ for } \odot \in \{\wedge, \vee\};\\
&(A \rightarrow B)(L, v) \stackrel{\text{def}}{=} A(L, v) \leq_t B(L, v);\\
&\big(\forall x(A(x))\big)(L, v) \stackrel{\text{def}}{=} \min\{A(a)(L, v) \mid a \in \textit{Const}\};\\
&\big(\exists x(A(x))\big)(L, v) \stackrel{\text{def}}{=} \max\{A(a)(L, v) \mid a \in \textit{Const}\};\\
&(A \in T)(L, v) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ when } A(L, v) \in T; \\ \mathfrak{f} \text{ otherwise}; \end{cases}\\
&(A = t)(L, v) \stackrel{\text{def}}{=} (A \in \{t\})(L, v).
\end{aligned}$$

[5] For motivations behind $\leq_t$ see, e.g., [3,39].

Given $v : \mathit{Var} \longrightarrow \mathit{Const}$ and $\mathcal{B} = \langle \Delta, \mathcal{C} \rangle$,

$$A(\Delta, v) \stackrel{\text{def}}{=} A\Big(\bigcup_{L \in \Delta} L\ , v\Big);$$

$$\big(\mathrm{Bel}_{\mathcal{B}}(A)\big)(v) \stackrel{\text{def}}{=} \begin{cases} \textsc{Lub}\{\ A(L, v) \mid L \in \Delta\} \text{ when} \\ \qquad \text{for all } C \in \mathcal{C},\ C(\Delta, v) = \mathfrak{t}; \\ \mathfrak{u} \quad \text{otherwise;} \end{cases}$$

$$A(\mathcal{B}, v) \stackrel{\text{def}}{=} \begin{cases} A(\Delta, v) & \text{when for all } C \in \mathcal{C}, C(\Delta, v) = \mathfrak{t}; \\ \mathfrak{u} & \text{otherwise;} \end{cases}$$

where LUB is the least upper bound wrt *information ordering* defined as the reflexive and transitive closure of the ordering shown in Figure 1. Observe that C, being a constraint, does not contain free variables, so v in $C(\Delta, v)$ is redundant. In similar cases we will use $C(\Delta)$ rather than $C(\Delta, v)$, and $C(\mathcal{B})$ rather than $C(\mathcal{B}, v)$.

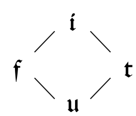

Fig. 1: Information ordering.

5 A Rule-Based Language for Beliefs

The $4QL^{Bel}$ language [6], an extension of 4QL [28,29,39], is a four-valued rule language designed for doxastic reasoning with paraconsistent and paracomplete belief bases. A unique feature of the 4QL-based language family is the presence of truth values $\mathfrak{t}$, $\mathfrak{f}$, $\mathfrak{i}$, $\mathfrak{u}$ as well as the unrestricted use of negation in both conclusions and premises of rules while retaining intuitive results and tractable query evaluation. Though the full definition of $4QL^{Bel}$ is available in [6], for clarity we recall the most important constructs of the language.

```
module moduleName:
    domains: ...
    relations: ...
    rules: ...
    facts: ...
end.
```

Module 1: Syntax of $4QL^{Bel}$ modules.

The language inherits a fair amount of elements from 4QL, including basic program syntax and semantics. The $4QL^{Bel}$ program consists of modules, structured as shown in Module 1. Sections **domains** and **relations** are used to specify domains and signatures of relations used in rules.[6] $4QL^{Bel}$ rules have the following form, where $\langle Formula \rangle$ is an arbitrary formula of the logic presented in Section 4:

$$\langle Literal \rangle \ \text{:–}\ \langle Formula \rangle\,. \tag{4}$$

[6] Due to the page limit, we will later skip them.

Facts are rules with the empty $\langle Formula \rangle$ part (being $\mathfrak{t}$). In such cases we simply write $\langle Literal \rangle$. A rule of the form (4) is "fired" for its ground instantiations when the truth value of $\langle Formula \rangle$ is $\mathfrak{t}$ or $\mathfrak{i}$.[7] As the effect:

– $\langle Literal \rangle$ is added to the set of conclusions when the truth value of $\langle Formula \rangle$ is $\mathfrak{t}$; (5)

– $\langle Literal \rangle$ and $\neg\langle Literal \rangle$ are added to the set of conclusions when the truth value of $\langle Formula \rangle$ is $\mathfrak{i}$. (6)

Note that ' :– ' is formalized in the 4QL-based languages by a generalization of the Shepherdson's implication [38] rather than by the $\rightarrow$ connective (see [39]). The implication $\rightarrow$ is more suitable for evaluating formulas while the former one reflects rule evaluation principles (5)–(6). To define the semantics of ' :– ' we use ordering $\leq_i$ which is reflexive and transitive closure of $\mathfrak{f} = \mathfrak{u} < \mathfrak{t} < \mathfrak{i}$. The implication $\rightsquigarrow$, corresponding to ' :– ', is defined by:

$$(A \rightsquigarrow B)(L, v) \stackrel{\text{def}}{=} A(L, v) \leq_i B(L, v). \tag{7}$$

When the set of truth values is restricted to $\{\mathfrak{t}, \mathfrak{f}, \mathfrak{u}\}$ or $\{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}$, the implication $\rightsquigarrow$ is the three valued implication of [38]. The semantics of rules is given by:

$$(C :\!- B)(L, v) \stackrel{\text{def}}{=} B(L, v) \rightsquigarrow C(L, v). \tag{8}$$

What distinguishes 4QL$^{\text{Bel}}$ from 4QL, is a support for doxastic reasoning due to use the $\text{Bel}()$ operator, enhancing advanced agents' reasoning. It will further be extended by providing means for belief and constraints shadowing.

Modules serve to structure belief bases. If m is a module name, $m.A$ expresses *references to* m. Semantically, one can view relation symbols within a module m as (implicitly) extended by prefix '$m.$'. In order to maintain a clear semantics and tractability, a certain form of acyclicity of references is required, close in spirit to stratification in logic programming and deductive databases [1] but concerning formulas with the operator '$\in T$' rather than negation.

Definition 2. *The* reference graph *of a set of modules Π consists of nodes labeled by names of modules occurring in Π, assuming that there is an edge between m and n iff premises of a rule in m contain an expression $A \in T$ or $A = t$, where A is a formula containing a reference of the form $n.B$. A 4QL^{Bel}* program *is set of modules whose reference graph is acyclic.* ◁

The semantics of 4QL$^{\text{Bel}}$ modules is given by *well-supported models* in the sense of [29]. A $3i$-world is a *model* of a module M if all rules of M, understood as implications (8), are true in the model. Intuitively, a model is well-supported when it consists of ground literals (if any) assuming that all literals it contains are conclusions of reasoning starting from facts. As shown in [6], for each 4QL$^{\text{Bel}}$ program, its well-supported model exists, is uniquely determined and can be computed in deterministic polynomial time wrt the size of all domains and number of modules.

[7] That is, the value of $\langle Formula \rangle$ contains some truth.

Note that a module specifies its well-supported model, so it can be identified with $3i$-worlds. In the sequel modules can appear wherever $3i$-worlds are allowed, in particular as elements of $\varDelta$ in a belief base $\langle \varDelta, \mathcal{C} \rangle$.

A high level agents' belief bases architecture is summarized in Figure 2. Note that agents may use multiple belief bases, some of which may be private, some own by groups, and some may be available to all agents. Query manager may be a 4QL$^{\text{Bel}}$ interpreter or another database querying engine.

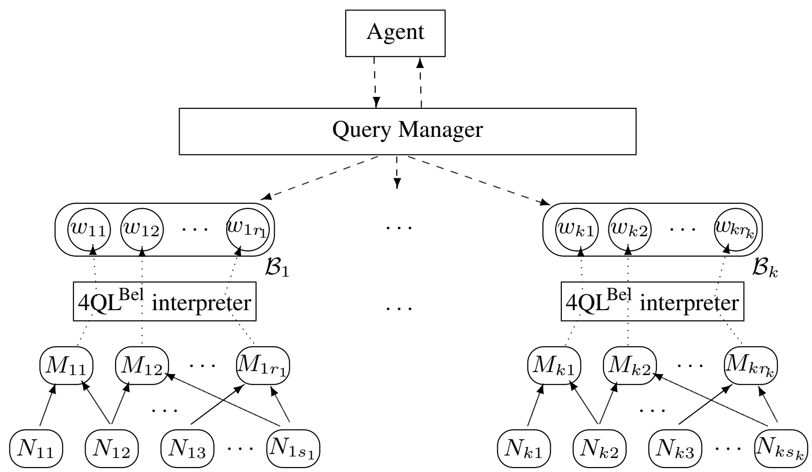

Fig. 2: High level agent's belief bases architecture, where, for $1 \leq l \leq k$, $\mathcal{B}_l$ are belief bases, $M_{l1}, \ldots, M_{lr_l}, N_{l1}, \ldots, N_{ls_l}$ are 4QL$^{\text{Bel}}$ modules, $w_{l1}, \ldots w_{lr_l}$ are the $3i$-worlds being respectively well-supported models of $M_{l1}, \ldots M_{lr_l}$, solid lines represent queries among modules, dotted lines represent correspondences between modules and $3i$-worlds, and dashed lines represent agent's queries.

6 Adding Integrity Constraints

Belief shadowing, similarly to belief revision or update, may result in creating undesirable conclusions. For example, some treatments can cause complications when applied to patients of a specific characteristics. To avoid such risky cases one could construct a specific rule. A better idea, however, is to formulate a general integrity constraint preventing patients from risky complications.

The idea of constraints is not new in information systems (see, e.g., [8,21,24,34,36]), where a distinction between hard and soft constraints might be desirable [31]. Hard constraints cannot be violated while soft ones are flexible and often considered as preferences whose violation should be avoided as long as possible. In our case a distinction between non-shadowable ("hard") and shadowable ("soft") constraints also appears useful. To avoid terminological misunderstandings we shall further call them *rigid* and *flexible* ones, respectively. For example, a patient's constraint concerning refusal of blood transfusion, when rigid, could not be shadowed, making a transfusion unacceptable regardless the circumstances. When being a flexible one, it can be shadowed allowing for blood transfusion.

In rule languages constraints are typically expressed by rules with empty heads expressing what is disallowed. Dually, in our approach constraints express what should always be true. The separation between constraints and rules gives the former ones an axiom-like flavor. Here constraints are formulated in modules in two subsections, separating rigid and flexible ones, $\mathcal{C} = \mathcal{C}_R \cup \mathcal{C}_F$. The distinction does not influence their semantics unless they appear in the context of the shadowing operator (see Section 7). Importantly, we require constraints to be true. This might seem restrictive in a four-valued framework. However, formulas of the form '$A \in T$' can be used, so requirements like "A should be true or inconsistent" can easily be expressed by '$A \in \{\mathfrak{t}, \mathfrak{i}\}$' being $\mathfrak{t}$ when the truth value of A is in the set $\{\mathfrak{t}, \mathfrak{i}\}$.

Though specified within modules, constraints may be local (limited to a single module) and global (span over multiple modules). Local constraints refer solely to relations in the same module. Accordingly, global constraints can contain literals referring to multiple, perhaps all, modules as long as references do not create cycles in the reference graph. Technically, to avoid cycles, additional modules can be created as a container for constraints. Such modules can be viewed as being "above" modules referenced by non-local constraints.

For notational compactness, in the sequel we shall use variables: T to represent treatments, P – patients and X – values of other types. Module 2 provides an example of global constraints related to the ER scenario. Assume that a woman, Pat, arrives to ER. Her (local) constraints are shown in Module 3.

```
module globalConstr:
    constraints:
        rigid: ∀P[data.patient(P)→ status.alive(P)=t].
        flexible:
            ∀T ∀P [(medicines.affects(T, bloodCoagulability)∧status.isPregnant(P)=t )
                → ¬mark.apply(T,P)]
end.
```

Module 2: A module expressing global constraints.

```
module pat:
    constraints:
        rigid: ∀T[accepts(T)∈{t, f, u}].
        flexible: ∀T[accepts(T)→medicines.isHomeopathic(T)].
    facts:
        severity(3).
        impairment(4).
        accepts(homeopatic1).
        ¬ accepts(bloodTransfusion).
end.
```

Module 3: The `pat` module.

We extend Definition 2 to deal with constraints as follows.

Definition 3. *By the* reference graph *of a set Π of modules with constraints we mean the reference graph for Π seen as a 4QL^{Bel} program (disregarding constraints), augmented with edges from m to n whenever constraints of m contain a reference to n, i.e., a subexpression of the form '$n.$'.* ◁

Definition 4. *By a 4QL^{Bel+} program we mean a set of 4QL^{Bel} modules with constraints such that its reference graph (in the sense of Definition 3) is acyclic and all constraints in modules are true.* ◁

Note that for every 4QL$^{\text{Bel+}}$ program Π, well-supported models of Π's modules are required to exist and, as in the case of 4QL$^{\text{Bel}}$ programs, are uniquely determined. For a 4QL$^{\text{Bel+}}$ module m, by $wsm(m)$ we denote the well-supported model of m. Using this correspondence between modules and well-supported models, being themselves $3i$-worlds, we can specify any belief base $\mathcal{B} = \langle \Delta, \mathcal{C} \rangle$ by:

$$\mathcal{B} = \langle m_1, \dots, m_k; n_1, \dots, n_l \rangle , \tag{9}$$

where $m_1, \dots, m_k, n_1, \dots, n_l$ are modules of a 4QL$^{\text{Bel+}}$ program. In such a case, $\Delta = \{wsm(m_1), \dots, wsm(m_k)\}$ and $\mathcal{C}$ consists of constraints of $n_1, \dots, n_l$ with rigid and flexible constraints collected from $n_1, \dots, n_l$, respectively. To simplify notation, we sometimes use single modules to represent belief bases, assuming that

$$\text{module } m \text{ represents the belief base } \langle m; m \rangle. \tag{10}$$

Of course, specifications of belief bases of the form (9) inherit all advantages of rule-based specifications. In particular, comparing to Definition 1, 4QL$^{\text{Bel+}}$-based specifications are typically much more concise and easier to understand and maintain.

7 The Shadowing Operator

To avoid semantical complexity, we treat shadowing as a formal expression rather than a belief base. However, to simplify presentation, syntactically we treat such formal expressions as belief bases. Thus, slightly abusing notation, we allow them to occur in the $\mathrm{Bel}()$ operator. Belief shadowing is defined by $\mathrm{Bel}_{\mathcal{B}_1 \,\mathsf{as}\, \mathcal{B}_2}(A)$ intuitively returning $\mathrm{Bel}_{\mathcal{B}_2}(A)$ when it is $\mathfrak{t}$, $\mathfrak{i}$ or $\mathfrak{f}$, or $\mathrm{Bel}_{\mathcal{B}_1}(A)$, when $\mathrm{Bel}_{\mathcal{B}_2}(A)$ is $\mathfrak{u}$. However, suitable constraints have to be validated. If they are not, $\mathrm{Bel}_{\mathcal{B}_1 \,\mathsf{as}\, \mathcal{B}_2}(A)$ returns $\mathfrak{u}$ for any query A.

Belief shadowing, denoted by as , is a left-associative operation. That is, $\mathcal{B}_1 \,\mathsf{as}\, \mathcal{B}_2 \,\mathsf{as}\, \mathcal{B}_3 \stackrel{\text{def}}{=} (\mathcal{B}_1 \,\mathsf{as}\, \mathcal{B}_2) \,\mathsf{as}\, \mathcal{B}_3$. To define belief shadowing we need an auxiliary operator $\rtimes$ allowing one to fuse beliefs from possibly different belief bases. Let, in (11) and Definitions 5, 6, $\mathcal{B}^1 = \langle \Delta^1, \mathcal{C}_R^1 \cup \mathcal{C}_F^1 \rangle$ and $\mathcal{B}^2 = \langle \Delta^2, \mathcal{C}_R^2 \cup \mathcal{C}_F^2 \rangle$ be belief bases. Then:

$$\mathrm{Bel}_{\mathcal{B}_1 \rtimes \mathcal{B}_2}(A) \stackrel{\text{def}}{=} \begin{cases} \mathrm{Bel}_{\mathcal{B}_2}(A) \text{ when } \mathrm{Bel}_{\mathcal{B}_2}(A) \in \{\mathfrak{t}, \mathfrak{f}, \mathfrak{i}\}; \\ \mathrm{Bel}_{\mathcal{B}_1}(A) \text{ when } \mathrm{Bel}_{\mathcal{B}_2}(A) = \mathfrak{u}. \end{cases} \tag{11}$$

We are now ready to define integrity constraints and belief shadowing, $\mathcal{B}_1 \,\mathsf{as}\, \mathcal{B}_2$, the central concepts of our approach.

Definition 5. *By* integrity constraints of $\mathcal{B}_1\,as\,\mathcal{B}_2$ *we understand the set* $\mathcal{C}_R^1 \cup \mathcal{C}_R^2 \cup \mathcal{C}_F^2$ *with* $\mathcal{C}_R^1 \cup \mathcal{C}_R^2$ *being* rigid constraints *and* $\mathcal{C}_F^2$ *being* flexible constraints of $\mathcal{B}_1\,as\ \mathcal{B}_2$. ◁

Definition 6. *The* belief operator over belief base $\mathcal{B}_1$ shadowed by *belief base* $\mathcal{B}_2$*,* $\mathrm{Bel}_{\mathcal{B}_1\,as\,\mathcal{B}_2}\big(\,\big)$*, is defined by:*

$$\mathrm{Bel}_{\mathcal{B}_1\,as\ \ \mathcal{B}_2}\big(A\big) \stackrel{\text{def}}{=} \begin{cases} \mathrm{Bel}_{\mathcal{B}_1 \rtimes \mathcal{B}_2}\big(A\big) & \textit{when for any} \\ & C \in \mathcal{C}_R^1 \cup \mathcal{C}_R^2 \cup \mathcal{C}_F^2, C'(\mathcal{B}_1) = \mathfrak{t} \\ \mathfrak{u} & \textit{otherwise,} \end{cases}$$

where C' *is obtained from* C *by substituting references to* $\mathcal{B}_1$ *in subformulas of the form* $\mathrm{Bel}_{\ldots\mathcal{B}_1\ldots}\big(\ldots\big)$ *by* $\mathcal{B}_1 \rtimes \mathcal{B}_2$. ◁

Though constraints of belief bases are always true, as required in Definition 4, constraints of $\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2$ may be unsatisfied for some $\mathcal{B}_1$ and $\mathcal{B}_2$. When this occurs, we assume that any query to $\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2$ returns the empty set of tuples with the truth value $\mathfrak{u}$. Note that some queries may also return $\mathfrak{u}$ when constraints are satisfied but $\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2$ contains no facts supporting or denying such queries. These cases can be distinguished without recalculating constraints, e.g., using the query $\mathrm{Bel}_{\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2}\big(\mathfrak{t}\big)$ which returns $\mathfrak{t}$ when constraints are satisfied and $\mathfrak{u}$ otherwise.

When the patient is treated in ER, final diagnoses concerning Pat are placed in the $3i$-world `diagnosed`, shown as Module 4. According to Module 4, Pat's current status is much worse than she assumes. Together with facts 6–9 of Module 3, Pat's flexible constraints are shadowed by module `diagnosed` as well, so that more invasive treatments become allowed. On the other hand, violating rigid constraints is unsupported: for arbitrary module `m`, whenever constraints of '`pat as m`' are not satisfied, $\mathrm{Bel}_{\texttt{pat as m}}\big(A\big)$ returns the value $\mathfrak{u}$, for any formula A.

```
module diagnosed:
  constraints:
    flexible:
      ∀T∀X[(accepts(T)∧severity(X)∧X≤8)→ medicines.isHomeopathic(T)].
  facts:
    severity(9).      impairment(8).
    heartFailure().   ¬heartStopped().
    headInjury(severe).
end.
```

Module 4: The module with diagnoses of Pat's condition.

Apart from diagnostic data, Mark's belief base, $\mathcal{B}_M$, contains also a module `mark` used for actual selection of the treatment, shown as Module 5. It contains (respective parts of) Mark's medical knowledge like rules concerning head injuries and heart pathology.[8] For simplicity, rather than in general form, we instantiate them with Pat related modules. Observe that $\mathcal{B}_M$ can also contain modules with alternative treatments, so it may remain indeterministic in this respect.

[8] Defibrillation is crucial when the heart has lost natural rhythm but still maintains electrical activity. However, it is discouraged after the hearth has already stopped. In such cases, cardiac massage and appropriate drugs should be used instead.

Mark's individual decisions may be shadowed by MDM decisions as specified by a flexible constraint in Module 5. Note that constraints make Mark's belief base robust wrt inconsistencies. Firstly, any constraints' violating attempts to update Mark's belief base should be rejected by the belief base management system. Secondly, any shadowing violating Mark's rigid constraints returns the value $\mathfrak{u}$ for every query, placing (shadowed) Mark out of the game for the time being.

```
1 module mark:
2   constraints:
3     rigid:
        ∀P[(data.patient(P)∧status.atRisk(P)∧status.conscious(P))→status.obey(P)].
4     flexible: ∀P[data.patient(P)→¬apply(headSurgery,P)].
5   rules:
6     apply(defibrillation, pat) :–
          Bel_{pat as diagnosed}(heartFailure()∧¬heartStopped()).
      apply(cardiacMassage, pat) :– Bel_{pat as diagnosed}(heartStopped()).
7     apply(mri, pat) :–
          Bel_{pat as diagnosed}(headInjury(medium)∨ headInjury(severe)).
8     mdmNeeded() :– Bel_{pat as diagnosed}(headInjury(severe)).
9 end.
```

Module 5: The `mark` module.

One can query about Mark's beliefs concerning treatments that should be applied to Pat using $\mathrm{Bel}_{\mathcal{B}_M}\big(\text{apply(T, pat)}\big)$ which, in our case, returns "defibrillation" and "mri" (magnetic resonance imaging), both with the truth value $\mathfrak{t}$.

```
module mdmMemeber:
  rules:
    apply(headSurgery, pat) :– Bel_{pat as diagnosed}(headInjury(severe)).
end.
```

Module 6: The module representing an MDM member.

Observe that the rule in line 8 of Module 5 makes mdmNeeded() true. When Mark requests consultations, queries about Pat's treatment can be asked to Mark being an MDM member by $\mathrm{Bel}_{\mathcal{B}_M \text{ as } \texttt{mdmMember}}\big(\text{apply(T, pat)}\big)$ which now returns "defibrillation", "mri" and "headSurgery" since Mark's flexible constraint disallowing the surgery has been shadowed by his MDM membership.

In order to shadow flexible constraints of a belief base $\mathcal{B}$ without affecting other beliefs, the empty belief base $\texttt{flexible} \stackrel{\text{def}}{=} \langle\{\emptyset\}, \emptyset\rangle$ can be used. Indeed, according to Definition 6, in '$\mathcal{B}$ `as flexible`' only rigid constraints of $\mathcal{B}$ remain. Also, for any literal l, $\mathrm{Bel}_{\texttt{flexible}}\big(l\big) = \mathfrak{u}$, so the original beliefs of $\mathcal{B}$ are not affected. In the scenario, to shadow Pat's flexible constraints solely, Mark can use the expression '`pat as flexible`'.

8 Properties of Shadowing

For any belief bases, $\mathcal{B}_1$, $\mathcal{B}_2$, the operator $\mathrm{Bel}_{\mathcal{B}_1 \text{ as } \mathcal{B}_2}\big(\big)$ satisfies (KD45n) axioms.

Proposition 1. *For any belief bases, $\mathcal{B}_1$, $\mathcal{B}_2$ and formula A,*

$$\mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(A) \rightarrow \neg \mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(\neg A); \tag{12}$$

$$\mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(A) \rightarrow \mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(\mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(A)); \tag{13}$$

$$\neg \mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(A) \rightarrow \mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(\neg \mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}(A)). \tag{14}$$

◁

Note, however, that axioms (12)–(14) do not have the classical meaning. For example, when the truth value of A is $\mathfrak{i}$, the implication (12) holds but it does not mean that (an inconsistent) belief in A prevents (inconsistent) belief in $\neg A$. Indeed, in this case, both $\mathrm{Bel}_{\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2}(\mathfrak{i})$ and $\neg\mathrm{Bel}_{\mathcal{B}_1\,\mathrm{as}\,\mathcal{B}_2}(\neg\mathfrak{i})$ are $\mathfrak{i}$.

Complexitywise, similarly to [6], we have the following propositions, where for any $4\mathrm{QL}^{\mathrm{Bel+}}$ program Π, $\#D$ denotes the sum of the sizes of all domains of Π; $\#\Pi$ denotes the number of modules in Π.

Proposition 2. *For every $4QL^{Bel+}$ program Π, both:*

- *checking the existence of the well-supported models of modules of Π;*
- *computing well-supported models of modules of Π,*

can be done in PTIME in $\max\{\#D, \#\Pi\}$. ◁

Proposition 3. *Given belief bases $\mathcal{B}_1, \mathcal{B}_2$ expressed using modules of a $4QL^{Bel+}$ program Π, the problem of computing queries involving expressions of the form* $\mathrm{Bel}_{\mathcal{B}_1\, \textit{as}\, \mathcal{B}_2}()$ *has deterministic polynomial time complexity in* $\max\{\#D, \#\Pi\}$. ◁

Since shadowing is defined in terms of belief base queries, as a consequence of the corresponding result of [6], we have the following proposition.

Proposition 4. *Assuming that domains are linearly ordered, every polynomially computable shadowing can be expressed in $4QL^{Bel+}$ using the* `as` *operator.* ◁

From the perspective of systems' engineering, the above complexity results are important. However, even tractability does not guarantee scalability over big data. Belief shadowing can be made horizontally scalable when recursive queries are not allowed, assuming that all $4\mathrm{QL}^{\mathrm{Bel}}$ modules are already computed and belief bases consist of the resulting $3i$-worlds. In this case, $\mathrm{Bel}()$-free formulas are equivalent to first-order (and non-recursive SQL) queries which can be evaluated in a horizontally scalable manner. Queries involving belief operators can also be easily horizontally distributed (with a separate thread evaluating a given query in each $3i$-world of a given belief base).

9 Related Work

Beliefs and their modifications are intensively tackled in many contexts. A fundamental issue is the definition of different kinds of beliefs [11,12,15,32,22] together with sophisticated structures like belief sets and belief bases [16,17,37]. Our approach builds on paraconsistent and paracomplete belief bases understood as in [9,10]. Moreover, we equip them with constraints, creating a convenient reasoning engine with built-in safety tools vital for maintaining belief bases. As we have shown, constraints are naturally applicable in belief shadowing and can be shadowed, too.

As agent systems act in dynamic environments, belief update and revision are in the mainstream of the area. For representative approaches see [19,26,27,30,33] and references there. Importantly, belief revision has been found one of the most fundamental research topics [14,18] aiming at consistent and deterministic solutions. Among others, the well known AGM [2] model was developed as a theoretical framework for adequate belief modification practices. It inspired a large body of work over many years. For surveys see [13] and references there. A significant amount of AGM extensions and improvements have been proposed, including paraconsistent ones [35,41,40]. Apart from undeniable profits, belief modifications can be computationally expensive, and create some other issues, like underdetermination (inability to determine rules to be defeated). Our belief shadowing significantly differs from belief update or revision and provides a remedy for these issues.

An alternative framework, belief merging, is addressed in many sources (for an overview see [25]). The authors study merging several belief bases in the presence of integrity constraints. The presented solutions do not allow inconsistent belief bases which forces the authors to look for consistency preserving belief merging operators. Our framework is more general: both input belief bases and the resulting beliefs can be inconsistent which offers flexibility of the specifications. Also, the complexity of belief merging is typically high (see [23]) while our framework guarantees tractability.

Another aspect of beliefs' dynamics is addressed in [9,10], where transformations of initial raw beliefs into more abstract, mature ones have been modeled. Belief dynamics is approached there via epistemic profiles permitting to model both beliefs related to states of the environment and deliberative processes of agents. Belief shadowing can contribute to express epistemic profiles flexibly and efficiently. Apart from the area of belief changes, our approach is rooted in the field of paraconsistent reasoning [5]. It is based on a logic derived from [6,9,29,42].

10 Conclusions

We have provided a novel, tractable and natural framework for modeling everyday human-like belief shadowing. The framework focuses on belief changes in dynamic environments. We have identified a broad niche where known belief change techniques can be substantially improved by developing a lightweight method of belief shadowing.

Besides providing efficient solution to the addressed phenomena, belief shadowing is also meant to complement belief revision/update/merging when these methods are difficult or impossible to apply. Firstly, when an agent acts in an unknown environment, frequent belief revisions might be needed. As such revisions may be computationally demanding, the shadowing machinery can serve as a "buffer" gathering new observations. Deeper revisions could then be postponed till the proper moment. Secondly, belief revisions might be hardly applicable, for example, when many rules contribute to a particular conclusion contradicting the observed reality. Then, belief shadowing provides a more nuanced means than just to live with inconsistency. Dynamic reasoning with beliefs and their interferences calls for safety mechanism preventing from forbidden states. This is well visible in machine learning, in particular data/rule mining. To ensure the required properties of belief bases at various abstraction levels, we have defined a constraint shadowing technique.

Last but not least, to illustrate how belief and constraint shadowing may be embedded into an existing rule language, we have extended $4QL^{Bel}$ by adding constraints and shadowing operator. The obtained $4QL^{Bel+}$ language provides tractable querying machinery and is strong enough to express all shadowings computable in deterministic polynomial time. Interestingly, it permits to combine paracomplete and paraconsistent reasoning with lightweight versions of nonmonotonic and doxastic reasoning. As we indicated, it is horizontally scalable wrt non-recursive queries.

References

1. Abiteboul, S., Hull, R., Vianu, V. (eds.): Foundations of Databases: The Logical Level. Addison-Wesley, Boston, MA, USA (1995)
2. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)
3. de Amo, S., Pais, M.: A paraconsistent logic approach for querying inconsistent databases. Int. J. of Approx. Reasoning 46, 366–386 (2007)
4. Bertossi, L.E., Hunter, A., Schaub, T. (eds.): Inconsistency Tolerance, LNCS, vol. 3300. Springer, Heidelberg, New York (2005)
5. Béziau, J.Y., Carnielli, W., Gabbay, D. (eds.): Handbook of Paraconsistency. College Publications, London (2007)
6. Białek, Ł., Dunin-Kęplicz, B., Szałas, A.: Rule-based reasoning with belief structures. In: Kryszkiewicz, M., et.al. (eds.) Proc. ISMIS'2017. LNAI, vol. 10352, pp. 229–239. Springer (2017)
7. Bochman, A.: A Logical Theory of Nonmonotonic Inference and Belief Change. Springer-Verlag New York, Inc., New York, NY, USA (2001)
8. Dechter, R.: Constraint Processing. Morgan Kaufmann, San Francisco, CA, USA (2003)
9. Dunin-Kęplicz, B., Szałas, A.: Taming complex beliefs. Transactions on Comp. Collective Intelligence XI LNCS 8065, 1–21 (2013)
10. Dunin-Kęplicz, B., Szałas, A.: Indeterministic belief structures. In: Jezic, G., et.al. (eds.) Agent and Multi-Agent Systems: Technologies and Applications, Adv. in Int. Syst. Comp., vol. 296, pp. 57–66. Springer (2014)
11. Dunin-Kęplicz, B., Verbrugge, R.: Teamwork in Multi-Agent Systems. A Formal Approach. John Wiley & Sons (2010)
12. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. The MIT Press, Cambridge, MA (2003)
13. Fermé, E., Hansson, S.O.: Agm 25 years: Twenty-five years of research in belief change. J. of Philosophical Logic 40(2), 295–331 (2011)
14. Gärdenfors, P.: Conditionals and changes of belief. In: Niiniluoto, I., Tuomela, R. (eds.) The Logic and Epistemology of Scientific Belief. North-Holland, Amsterdam (1978)
15. Hadley, R.F.: The many uses of 'belief' in AI. Minds and Machines 1(1), 55–73 (1991)
16. Hansson, S.O.: Taking belief bases seriously. In: Prawitz, D., Westerståhl, D. (eds.) Logic and Phil. of Sci. in Uppsala, pp. 13–28. Springer (1994)
17. Hansson, S.O.: Revision of belief sets and belief bases. In: Dubois, D., Prade, H. (eds.) Belief Change, pp. 17–75. Springer, Dordrecht (1998)

18. Hansson, S.O.: A Textbook of Belief Dynamics. Theory Change and Database Updating. Kluwer Academic Publishers, Amsterdam, Netherlands (1999)
19. Herzig, A., Rifi, O.: Propositional belief base update and minimal change. Artificial Intelligence 115(1), 107 – 138 (1999)
20. Hewitt, C., Woods, J. (eds.): Inconsistency Robustness. College Pub., London (2015)
21. van Hoeve, W.J., Katriel, I.: Global constraints. Foundations of AI 2, 169 – 208 (2006)
22. Huber, F.: Formal representations of belief. In: Zalta, E.N. (ed.) The Stanford Enc. of Philosophy. Stanford University, spring 2016 edn. (2016)
23. Konieczny, S., Lang, J., Marquis, P.: Da^2 merging operators. Artif. Intell. 157(1-2), 49–79 (2004)
24. Konieczny, S., Pino Pérez, R.: Merging with integrity constraints. In: Hunter, A., Parsons, S. (eds.) Proc. ECSQARU'99 Conf.: Symbolic and Quantitative Approaches to Reasoning and Uncertainty. LNCS, vol. 1638, pp. 233–244. Springer (1999)
25. Konieczny, S., Pino Pérez, R.: Merging information under constraints: A logical framework. J. Log. Comput. 12(5), 773–808 (2002)
26. Lang, J.: Belief update revisited. In: Proc. 20th IJCAI. pp. 2517–2522. Morgan Kaufmann (2007)
27. Liberatore, P.: The complexity of belief update. Artificial Intelligence 119(1), 141 – 190 (2000)
28. Małuszyński, J., Szałas, A.: Living with inconsistency and taming nonmonotonicity. In: de Moor, O., et.al. (eds.) Datalog 2.0. LNCS, vol. 6702, pp. 384–398. Springer (2011)
29. Małuszyński, J., Szałas, A.: Partiality and inconsistency in agents' belief bases. In: Barbucha, D., et.al. (eds.) Proc. of the 7th KES AMSTA Conf. pp. 3–17. IOS Press (2013)
30. Marchi, J., Bittencourt, G., Perrussel, L.: A syntactical approach to belief update. In: Gelbukh, A., et.al. (eds.) Proc. MICAI 2005, pp. 142–151. Springer, Berlin, Heidelberg (2005)
31. Meseguer, P., Rossi, F., Schiex, T.: Soft constraints. In: Rossi et al. [36], pp. 281 – 328
32. Meyer, J.J.C., van der Hoek, W.: Epistemic Logic for AI and Theoretical Computer Science. Cambridge Univ, Press, NY (1995)
33. Peppas, P.: Belief revision. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of KR, pp. 317–359. Elsevier (2008)
34. Pigozzi, G.: Belief merging and judgment aggregation. In: Zalta, E. (ed.) The Stanford Enc. of Philosophy. Stanford University, winter 2016 edn. (2016)
35. Priest, G.: Paraconsistent belief revision. Theoria 67(3), 214–228 (2001)
36. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming, Foundations of AI, vol. 2, Supplement C. Elsevier (2006)
37. Santos, Y.D., Ribeiro, M.M., Wassermann, R.: Between belief bases and belief sets: Partial meet contraction. In: Proc. 2015 Int. Conf. on Defeasible and Ampliative Reasoning. DARe'15, vol. 1423, pp. 50–56. CEUR-WS.org, Aachen, Germany (2015)
38. Shepherdson, J.: Negation in logic programming. In: Minker, J. (ed.) Foundations of Deductive Databases and Logic Programming, pp. 19–88. Morgan Kaufmann, Burlington, MA (1988)
39. Szałas, A.: How an agent might think. Logic Journal of the IGPL 21(3), 515–535 (2013)
40. Testa, R.R., Coniglio, M.E., Ribeiro, M.M.: Paraconsistent belief revision based on a formal consistency operator. CLE E-Prints 15(8), 01–11 (2015)
41. Testa, R.R., Coniglio, M.E., Ribeiro, M.M.: AGM-like paraconsistent belief change. Logic Journal of the IGPL 25(4), 632–672 (2017)
42. Vitória, A., Małuszyński, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
43. Wooldridge, M.: Reasoning About Rational Agents. MIT Press, Cambridge MA (2000)