Workshop Notes

Workshop on

Belief change, Non-monotonic reasoning and Conflict Resolution

BNC@ECAI 2012

August 27, 2012
Montpellier, France

held at the

European Conference on Artificial Intelligence

Editors

Sébastien Konieczny
Thomas Meyer

http://cair.meraka.org.za/~bnc2012

Preface

Belief change, non-monotonic reasoning and conflict resolution are well established research areas in Artificial Intelligence. In recent years these topics have become important for designing robots and infobots with convincing reasoning and adaptation capabilities. Numerous recent papers use techniques from belief change to define conflict resolution methods. In particular, several negotiation and judgment aggregation methods are closely related to work in belief revision or belief merging.

The main aim of BNC@ECAI 2012 is to bring together active researchers on work including belief revision, belief merging, reasoning about action, logic programming, inconsistency management, judgment aggregation, negotiation, and other related topics, and to combine ideas from these research topics. A further important trend is the study of the applicability of well known belief change operators and techniques for particular languages that are largely used in applications, such as Horn logics, description logics, or argumentation frameworks.

Submissions to this workshop were reviewed by at least two PC members and were evaluated on relevance and quality. Eighty percent of the submissions have been selected for presentation at the workshop and for inclusion in these Workshop Notes.

We wish to thank the authors, members of the Program Committee, and the Additional Reviewers for their invaluable and much appreciated contributions. Thank you also to Kody Moodley for managing the website, and to Renata Wassermann for agreeing to be the invited speaker. Finally, thank you to Ivan Varzinczak who provided support in producing these workshop notes.

August 2012

Sébastien Konieczny, Thomas Meyer

Table of Contents

Chairs

Sébastien Konieczny	CRIL-CNRS
Thomas Meyer	CSIR Meraka and University of KwaZulu-Natal

Program Committee

Richard Booth	University of Luxembourg
Gerhard Brewka	Leipzig University
James Delgrande	Simon Fraser University
Ulle Endriss	ILLC, University of Amsterdam
Eduardo Fermé	Universidade da Madeira
Weiru Liu	Queen's University Belfast
Maurice Pagnucco	The University of New South Wales
Pavlos Peppas	University of Patras
Gabriella Pigozzi	Université Paris Dauphine
Ramon Pino Perez	Universidad de Los Andes
Torsten Schaub	University of Potsdam
Leon Van Der Torre	ILIAS
Ivan Varzinczak	CSIR Meraka and University of KwaZulu-Natal
Renata Wassermann	University of São Paulo

Additional Reviewers

Benjamin Andres
Ringo Baumann
Ken Halland
Kody Moodley
Orkunt Sabuncu

Website Maintenance

Kody Moodley	CSIR Meraka and University of KwaZulu-Natal

Invited Paper

Belief Revision and Computer Science

Renata Wassermann
Departmento of Computer Science
University of São Paulo, Brazil
renata@ime.usp.br

Abstract

In the 80's AGM theory for Belief Revision was mainly developed by philosophers and logicians, while at the same time, computer scientists were facing concrete problems as database update and model-based diagnoses. Even if there were a few early attempts to implement belief revision systems, they had the flavour of ad hoc solutions to circumvent the computational complexity of the logical theory. Recently, with the advances in computational power, it became more plausible to have implementations of the real theory. In this talk I plan to explore two sides of the relationship between AGM and Computer Science: (i) using AGM for CS, as for example in proposals for AGM revision of system specification and (ii) using CS for AGM, by means of real experimentation with the theory.

Regular Papers

Circumscribing *DL-Lite*

Elena Botoeva and **Diego Calvanese** [1]

Abstract. Classical logics (and hence Description Logics) are monotonic: the set of conclusions increases monotonically with the set of premises. Instead, common-sense reasoning is characterized as non-monotonic: new information can invalidate some of the previously made conclusions. Circumscription is one of the main non-monotonic formalisms whose idea is to minimize (circumscribe) the extension of given predicates. In this paper we study circumscribed *DL-Lite* knowledge bases and show how to compute circumscription of a single predicate (either a concept or a role) in a *DL-Lite*$^{\mathcal{H}}_{bool}$ knowledge base. Unlike other works on circumscribed Description Logics KBs, we are interested not only in checking entailment, but actually in computing circumscription itself. We show that circumscription of a role in *DL-Lite*$^{\mathcal{H}}_{bool}$ requires the language of $\mathcal{ALCHOIQ}$ extended with union or roles, thus is first-order expressible.

1 Introduction

Description Logics (DLs) [2] are acknowledged as computationally well-behaved fragments of first-order logic, and widely used in areas such as Knowledge Representation, Semantic Web and Ontology-Based Data Access for automated reasoning. There has been a continuous interest in non-monotonic extensions of DLs, and a considerable amount of work in that field includes extensions of DLs with default logic [28, 3, 30], with preference relation [19, 13, 7, 10], with circumscription [23, 6, 15, 5], with defeasible logic [25, 14, 32, 16] and with logic of Minimal Knowledge and Negation as Failure [22, 12, 17, 24].

Motivation for non-monotonic reasoning comes from the need to handle real life scenarios when the knowledge about the world is incomplete or changing. One of the motivations for non-monotonic DLs stems from the biomedical domain [26] where DLs are used as a tool for the formalization of ontologies such as SNOMED [11] and GALEN [27]. Another motivation comes from policy languages based on DLs [31, 34], which require non-monotonic reasoning. Prototypical properties and defeasible inheritance in DLs can also be added to the wish list.

In our work we have chosen circumscription as the underlying non-monotonic formalism for two main reasons. First, the semantics of circumscription is sufficiently simple, so circumscribed DLs can be defined in a straightforward way. Second, the existing works on circumscribed DLs [6, 5] show that they are interesting objects to be investigated and one could get nice results if one used a low complexity DL. More precisely, the current approaches to circumscribed DL knowledge bases (KBs) can be divided in two according to the DL used: expressive DLs such as $\mathcal{ALC}$, $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$ [6], and tractable DLs such as $\mathcal{EL}$ and $\mathcal{EL}^{++}$ [4, 5]. The former showed that reasoning in circumscribed $\mathcal{ALC}$ KBs is NEXPTIME$^{\text{NP}}$-hard, while some forms of reasoning in circumscribed $\mathcal{EL}$ KBs are tractable.

In this paper we investigate circumscription in *DL-Lite*$^{\mathcal{H}}_{bool}$, which is a sub-logic of the expressive DL $\mathcal{ALCHI}$ (essentially $\mathcal{ALC}$ with role hierarchy and inverse roles), and a super-logic of *DL-Lite*$_{\mathcal{R}}$, the basic *DL-Lite* logic. *DL-Lite*$^{\mathcal{H}}_{bool}$ is a member of the extended *DL-Lite* family [1], popular for its low complexity of reasoning, notably AC^0 data complexity of answering (atomic) queries. In contrast with the previous works on circumscribed DLs we not only want to check entailment, but also to compute circumscription of *DL-Lite*$^{\mathcal{H}}_{bool}$ KBs. We show that the circumscription of a single predicate (concept or role) in *DL-Lite*$^{\mathcal{H}}_{bool}$ can be expressed in $\mathcal{ALCHOIQ}$ with union of roles.

The paper is organized as follows: In Section 2 we introduce the logic *DL-Lite*$^{\mathcal{H}}_{bool}$ and the notion of circumscription. Section 3 presents circumscribed *DL-Lite* and includes a motivating example. In Section 4, we show how to compute circumscription of a single predicate in *DL-Lite*$^{\mathcal{H}}_{bool}$, and in Section 5, we show how to check entailment in the circumscribed KB. Finally, in Section 6, we draw some conclusions and outline issues for future work.

2 Preliminaries

We introduce the DLs that we adopt in this paper, and then recall the notions about circumscription.

2.1 Description Logics

Here, we present the DL *DL-Lite*$^{\mathcal{H}}_{bool}$, a member of the extended *DL-Lite* family of DLs known for their nice computational properties [8, 1]. Good computational behavior of *DL-Lite*$^{\mathcal{H}}_{bool}$, which is a sub-logic of $\mathcal{ALCHI}$, is achieved by prohibiting concepts of the form $\exists R.C$ and $\forall R.C$. Satisfiability checking in *DL-Lite*$^{\mathcal{H}}_{bool}$ can be done in NP in combined complexity and in AC^0 in data complexity [1].

Let N_C, N_R, and N_a be countably infinite sets of concept, role and individual names, respectively. The language of *DL-Lite*$^{\mathcal{H}}_{bool}$ contains individual names $a, b \in N_a$, atomic concepts $A \in N_C$, and atomic roles $P \in N_R$. Complex roles Q and concepts C of this language are defined as follows:

$$\begin{array}{lcllll} R & ::= & P & \mid & P^- & \\ Q & ::= & R & \mid & \neg R & \\ B & ::= & \bot & \mid & A & \mid \exists R \\ C & ::= & B & \mid & \neg C & \mid C_1 \sqcap C_2 \end{array}$$

The concepts of the form B are called *basic* concepts and roles of the form R are called *basic* roles. Moreover, for a role R, we use R^- to denote P^- when $R = P$, and P when $R = P^-$. A predicate in DLs is either an atomic concept or an atomic role.

[1] KRDB Research Centre, Free University of Bozen-Bolzano, Italy, email: *lastname*@inf.unibz.it

A *DL-Lite*$^{\mathcal{H}}_{bool}$ *TBox*, $\mathcal{T}$, is a finite set of concept and role inclusion axioms (or simply concept and role *inclusions*) of the form:

$$C_1 \sqsubseteq C_2 \qquad \text{and} \qquad R \sqsubseteq Q,$$

and an *ABox*, $\mathcal{A}$, is a finite set of membership assertions:

$$A(a), \quad \neg A(a), \quad P(a,b), \quad \text{and} \quad \neg P(a,b).$$

A *DL-Lite*$^{\mathcal{H}}_{bool}$ KB $\mathcal{K}$ is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$.

The semantics of *DL-Lite*$^{\mathcal{H}}_{bool}$ is defined as usual in DLs. An *interpretation* $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ with non-empty domain $\Delta^{\mathcal{I}}$ and interpretation function $\cdot^{\mathcal{I}}$ that assigns *(i)* to every concept name A a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of the domain; *(ii)* to every role name P a binary relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ over the domain; *(iii)* to every individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

Concept and role constructs are interpreted as follows

$$\begin{array}{lll} (P^-)^{\mathcal{I}} = \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in P^{\mathcal{I}}\} & & \\ (\neg R)^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}} & & \bot^{\mathcal{I}} = \emptyset \\ (\exists R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\} & & \\ (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & & (C_1 \sqcap C_2)^{\mathcal{I}} = C_1{}^{\mathcal{I}} \cap C_2{}^{\mathcal{I}} \end{array}$$

We will use standard abbreviations such as $C_1 \sqcup C_2$ for $\neg(\neg C_1 \sqcap \neg C_2)$, and $\top$ for $\neg\bot$.

The satisfaction relation is defined as follows:

$$\begin{array}{llll} \mathcal{I} \models C_1 \sqsubseteq C_2 & \text{iff } C_1{}^{\mathcal{I}} \subseteq C_2{}^{\mathcal{I}} & \mathcal{I} \models R \sqsubseteq Q & \text{iff } R^{\mathcal{I}} \subseteq Q^{\mathcal{I}} \\ \mathcal{I} \models A(a) & \text{iff } a^{\mathcal{I}} \in A^{\mathcal{I}} & \mathcal{I} \models P(a,b) & \text{iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}} \\ \mathcal{I} \models \neg A(a) & \text{iff } a^{\mathcal{I}} \notin A^{\mathcal{I}} & \mathcal{I} \models \neg P(a,b) & \text{iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \notin P^{\mathcal{I}} \end{array}$$

We say that $\mathcal{I}$ is a *model* of a TBox if (resp., ABox) it satisfies all its axioms (resp., assertions). $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if it is a model of both $\mathcal{T}$ and $\mathcal{A}$. $\mathcal{K}$ is said to be *satisfiable* (or *consistent*) if it has a model.

DL-Lite$^{\mathcal{H}}_{bool}$ is a super-logic of other three *DL-Lite* DLs that differ in the form of allowed TBox inclusions [1]. Here we mention only *DL-Lite*$^{\mathcal{H}}_{core}$, also known as *DL-Lite*$_{\mathcal{R}}$ (in the original paper [8]). A TBox $\mathcal{T}$ is a *DL-Lite*$^{\mathcal{H}}_{core}$ TBox if its concept inclusions are of the form

$$B_1 \sqsubseteq B_2 \quad \text{or} \quad B_1 \sqsubseteq \neg B_2.$$

A *signature* Σ is a set of concept and role names, that is, $\Sigma \subseteq N_C \cup N_R$. Given a KB $\mathcal{K}$, the *signature* $\Sigma(\mathcal{K})$ *of* $\mathcal{K}$ is the alphabet of concept and role names occurring in $\mathcal{K}$ (and likewise for a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, a concept C, and a role R).

We are going to express circumscription in the language of $\mathcal{ALCHOIQ}$, which is $\mathcal{ALCHI}$ extended with nominals ($\mathcal{O}$) and qualified number restrictions ($\mathcal{Q}$). Here we present the missing constructs.

Let R be a basic role, as defined in the previous section. Then complex concepts C in $\mathcal{ALCHOIQ}$ are built according to the following syntax:

$$C ::= A \mid \exists R.C \mid \neg C \mid C_1 \sqcap C_2 \mid \{a_1, \ldots, a_n\} \mid \geq k\, R.C$$

where k is a non-negative integer and n is a positive integer. Here we have three new constructs: qualified existential restriction $\exists R.C$, nominals $\{a_1, \ldots, a_n\}$, and qualified number restriction $\geq k\, R.C$. Note that the construct $\exists R$ can be seen an abbreviation for $\exists R.\top$.

The new constructs are interpreted as follows:

$$\begin{array}{rl} (\exists R.C)^{\mathcal{I}} & = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\} \\ \{a_1, \ldots, a_n\}^{\mathcal{I}} & = \{a_1{}^{\mathcal{I}}, \ldots, a_n{}^{\mathcal{I}}\} \\ (\geq k\, R.C)^{\mathcal{I}} & = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in C^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\} \geq k\} \end{array}$$

In the following, we will use $\mathsf{Funct}(P)$ to abbreviate $\geq 2\, P \sqsubseteq \bot$.

2.2 Circumscription

Circumscription was introduced by McCarthy [23], and has been well studied and explored by Lifschitz [20, 21] and others [18, 33]. It is an important formalism of common-sense reasoning that offers non-monotonic reasoning abilities by circumscribing (minimizing) the extension of specific predicates. Below we briefly present the notion of circumscription and circumscribed theories.

First, for any predicate symbols P, Q of the same arity, $P = Q$ stands for $\forall x(P(x) \equiv Q(x))$, $P \leq Q$ stands for $\forall x(P(x) \rightarrow Q(x))$, and $P < Q$ stands for $(P \leq Q) \wedge \neg(P = Q)$.

Let $\Phi(P)$ be a first-order sentence containing a predicate constant P. Then, by definition, the *circumscription* of P in $\Phi(P)$, denoted $\mathsf{Circ}(\Phi; P)$, is the second-order formula

$$\Phi(P) \wedge \forall p \neg(\Phi(p) \wedge p < P),$$

where p is a predicate variable of the same arity as P.

More generally, we can simultaneously minimize several predicates, which gives *parallel* and *prioritized* circumscription (here we introduce only parallel circumscription). Moreover, we may allow the extension of some predicates to vary in order to make the extension of the minimized predicates smaller. Let P be a tuple of predicate constants, and Z a tuple of function and/or predicate constants disjoint with P, and $\Phi(P, Z)$ a sentence. Then the circumscription of P in $\Phi(P, Z)$ with variable Z, denoted $\mathsf{Circ}(\Phi; P; Z)$, is the sentence

$$\Phi(P, Z) \wedge \forall p z \neg(\Phi(p, z) \wedge p < P),$$

where the notation $P \sim Q$, with $\sim$ being one of $=, \leq, <$, is generalized to tuples of predicates: $P \leq Q$ stands for $P_1 \leq Q_1 \wedge \cdots \wedge P_n \leq Q_n$, similar for $P = Q$. Finally, $P < Q$ stands again for $(P \leq Q) \wedge \neg(P = Q)$.

The models of $\mathsf{Circ}(\Phi; P; Z)$ are the models of Φ such that the extension of P cannot be made smaller without losing the property Φ, even at the price of changing the interpretations of Z. In order to define a model formally, we need to define an order on interpretations.

Let $\mathcal{I}$ and $\mathcal{J}$ be two classical interpretations of Φ. Then we write $\mathcal{I} \leq^{P;Z} \mathcal{J}$ if

- $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$,
- $X^{\mathcal{I}} = X^{\mathcal{J}}$ for every X that does not belong to P, nor to Z,
- $X^{\mathcal{I}} \subseteq X^{\mathcal{J}}$ for every $X \in P$.

We write $\mathcal{I} <^{P;Z} \mathcal{J}$ if $\mathcal{I} \leq^{P;Z} \mathcal{J}$ but not $\mathcal{J} \leq^{P;Z} \mathcal{I}$.

An interpretation $\mathcal{I}$ is a *model* of $\mathsf{Circ}(\Phi; P; Z)$ if it is a model of Φ and it is minimal relative to $\leq^{P;Z}$, i.e., there is no other model $\mathcal{J}$ of Φ such that $\mathcal{J} <^{P;Z} \mathcal{I}$.

To ensure the existence of a model of $\mathsf{Circ}(\Phi; P; Z)$ we need Φ to be well-founded w.r.t. $(P; Z)$. Φ is said to be *well-founded* w.r.t. $(P; Z)$ if for every model $\mathcal{J}$ of Φ there exists a model $\mathcal{I}$ of Φ minimal relative to $\leq^{P;Z}$ and such that $\mathcal{I} \leq^{P;Z} \mathcal{J}$.

A lot of effort has been made to understand in which cases circumscription is first-order expressible, and what its computational properties are [21, 18, 33]. A simple case when circumscription is not first-order expressible is circumscribing a transitive binary predicate. Then circumscription of that predicate is equivalent to the transitive closure, and it cannot be reduced to a first-order sentence.

Below we present some results that help to compute circumscription:

- [21] if Ψ does not contain P, Z, then

$$\mathsf{Circ}(\Phi(P, Z) \wedge \Psi; P; Z) \equiv \mathsf{Circ}(\Phi(P, Z); P; Z) \wedge \Psi$$

- [21] if $\Psi(P)$ contains only negative occurrences of P, then

$$\mathsf{Circ}(\Phi(P) \wedge \Psi(P); P) \equiv \mathsf{Circ}(\Phi(P); P) \wedge \Psi(P),$$

  where an occurrence of P in $\Psi(P)$ is said to be *negative* if P appears negated in the negation normal form (NNF) of $\Psi(P)$.
- [21] if Ψ does not contain P, then

$$\mathsf{Circ}(\forall x\big(\Psi(x) \rightarrow P(x)\big); P) = \forall x\big(\Psi(x) \equiv P(x)\big),$$

  and it is called *predicate completion*.
- [21] if a sentence Φ is satisfiable and well-founded w.r.t. $(P; Z)$, then $\mathsf{Circ}(\Phi; P; Z)$ is satisfiable.
- [33] the finite model property of a first-order fragment implies its decidability under the circumscriptive semantics.

3 Circumscribed *DL-Lite*

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ be a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB. Let M and V be sets of predicates from the signature of $\mathcal{K}$, such that $M \cap V = \emptyset$. Then $\mathcal{K}$ *circumscribed w.r.t. minimized predicates* M *and varied predicates* V is an expression:

$$\mathsf{Circ}(\mathcal{K}; M; V).$$

If V is empty, we write $\mathsf{Circ}(\mathcal{K}; M)$. The rest of the predicates are assumed to be fixed and predicates from M are assumed to be minimized in parallel. If the ABox is empty, then we circumscribe only $\mathcal{T}$ and write $\mathsf{Circ}(\mathcal{T}; M; V)$.

We rely on the notion of a model as defined for the classical circumscription. An interpretation $\mathcal{I}$ is a *model* of $\mathsf{Circ}(\mathcal{K}; M; V)$ if it is a model of $\mathsf{Circ}(\Phi_{\mathcal{K}}; M; V)$, where $\Phi_{\mathcal{K}}$ is the standard translation of $\mathcal{K}$ to first-order logic.

Theorem 1. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ be a DL-Lite$^{\mathcal{H}}_{bool}$ KB, M, V sets of predicates from the signature of $\mathcal{K}$, such that $M \cap V = \emptyset$. Then $\mathcal{K}$ is well-founded w.r.t. $(M; V)$, and if $\mathcal{K}$ is satisfiable, then $\mathsf{Circ}(\mathcal{K}; M; V)$ is satisfiable.*

Proof. Well-foundedness follows from the finite model property of *DL-Lite*$^{\mathcal{H}}_{bool}$ and the last claim follows from Proposition 11 in [21]. □

The typical example for non-monotonic reasoning is the Tweety example. It can be encoded in circumscribed *DL-Lite*$^{\mathcal{H}}_{bool}$ as follows:

Example 1. Assume an ontology about birds. We want to express the following commonsense facts: typically birds fly, penguins are birds, and they cannot fly. Let $Bird$, $Abnormal$, $Penguin$, $Flier$ be concept names and $\mathcal{T}$ the following TBox

$$\begin{aligned} Bird \sqcap \neg Abnormal &\sqsubseteq Flier \\ Penguin &\sqsubseteq Bird \\ Penguin &\sqsubseteq Abnormal \\ Abnormal &\sqsubseteq \neg Flier \end{aligned}$$

Moreover, we assume that birds are considered normal if there is no evidence to the contrary. Therefore, we minimize the set of abnormal birds.

Suppose we know that Tweety is a bird, which is encoded in the ABox $A = \{Bird(\mathsf{tweety})\}$. Then, we obtain that

$$\mathsf{Circ}(\langle \mathcal{T}, \mathcal{A}\rangle; Abnormal) \models Flier(\mathsf{tweety}).$$

Now, assume we learn that Tweety is not just a bird, but a penguin, $\mathcal{A}' = \mathcal{A} \cup \{Penguin(\mathsf{tweety})\}$. Then

$$\mathsf{Circ}(\langle \mathcal{T}, \mathcal{A}'\rangle; Abnormal) \models \neg Flier(\mathsf{tweety}).$$

Thus, we have invalidated the previous conclusion that Tweety flies.

4 Computing Circumscription in *DL-Lite*$^{\mathcal{H}}_{bool}$

In this section we show how to compute circumscription of a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB with respect to a single predicate, that is, for $M = \{X\}$.

It is easy to compute circumscription of an atomic concept A in a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox $\mathcal{T}$. We start by observing that we can assume w.l.o.g. that each concept inclusion axiom in $\mathcal{T}$ has the 'normal' form $\top \sqsubseteq L_1 \sqcup \cdots \sqcup L_n$, where each L_i is either a basic concept or a negated basic concept, and no basic concept appears both positively and negatively in the same axiom. Indeed, the transformation of an arbitrary set of concept inclusions into this form is analogous to the conversion of a propositional formula into an equivalent set of clauses.[2] Hence, for a concept inclusion α of the above form, we say that α is *positive* (resp., *negative*) w.r.t. A if A appears positively (resp., negatively) in α. Let $\mathsf{Pos}_{\mathcal{T}}(A)$ be the set of all inclusions in $\mathcal{T}$ positive w.r.t. A, and $\mathsf{Neg}_{\mathcal{T}}(A)$ the set of all inclusions in $\mathcal{T}$ negative w.r.t. A. Moreover, again w.l.o.g., we may consider that each axiom in $\mathsf{Pos}_{\mathcal{T}}(A)$ has the form $C \sqsubseteq A$, for some concept C not containing A.

Proposition 2. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{bool}$ TBox and A an atomic concept of $\mathcal{T}$. Then*

$$\mathsf{Circ}(\mathcal{T}; \{A\}) = \mathcal{T} \cup \{C_1 \sqcup \cdots \sqcup C_n \equiv A\},$$

where $\mathsf{Pos}_{\mathcal{T}}(A) = \{C_i \sqsubseteq A\}_{i=1}^{n}$.

Proof. Since $\mathcal{T}$ is a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox, we have that

$$\mathcal{T} = \mathsf{Pos}_{\mathcal{T}}(A) \cup \mathsf{Neg}_{\mathcal{T}}(A) \cup \mathcal{T}',$$

where $\mathcal{T}'$ is the set of inclusions in $\mathcal{T}$ that do not contain A. Therefore, A does not appear in the concept $C_1 \sqcup \cdots \sqcup C_n$, and the result follows directly from the properties of circumscription. □

Notice that in *DL-Lite*$^{\mathcal{H}}_{bool}$, circumscribing an atomic concept corresponds to predicate completion. Also notice, that if $\mathcal{T}$ is a *DL-Lite*$^{\mathcal{H}}_{core}$ TBox, $\mathsf{Circ}(\mathcal{T}; \{A\})$ is a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB.

Computing circumscription is not so trivial when X is an atomic role P. In the following we compute circumscription of P in *DL-Lite*$^{\mathcal{H}}_{core}$ and *DL-Lite*$^{\mathcal{H}}_{bool}$ TBoxes.

4.1 Circumscribing a *DL-Lite*$^{\mathcal{H}}_{core}$ TBox

We start by circumscribing *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes. In *DL-Lite*$^{\mathcal{H}}_{core}$ a role P can appear positively in the assertions of the form:

$$R \sqsubseteq P, \qquad B \sqsubseteq \exists P, \qquad B \sqsubseteq \exists P^{-},$$

where R is a basic role and B is an atomic concept. First, we compute circumscription of P for several easy cases.

Let P be a role name, R a role, and C_1, C_2 concepts such that $P \notin \Sigma(\{R, C_1, C_2\})$. For an interpretation $\mathcal{I}$ and a tuple of domain elements (a, b), we denote by $\mathcal{I} \setminus P(a, b)$ the interpretation $\mathcal{I}'$ that agrees with $\mathcal{I}$ on all predicates except P and $P^{\mathcal{I}'} = P^{\mathcal{I}} \setminus \{(a, b)\}$).

1. $\mathsf{Circ}(\{R \sqsubseteq P\}; P) \equiv \{R \equiv P\}$
Proof. Follows from predicate completion. □

2. $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P\}; P) \equiv \{C_1 \equiv \exists P, \mathsf{Funct}(P)\}$
Proof. Let $\mathcal{I}$ be a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P\}; P)$. Then $\mathcal{I} \models C_1 \sqsubseteq \exists P$ and $\mathcal{I}$ is minimal relative to P. Assume that $\mathcal{I} \not\models \exists P \sqsubseteq C_1$, hence there exists a tuple $(a, b) \in P^{\mathcal{I}}$ s.t. $a \notin {C_1}^{\mathcal{I}}$. Then $\mathcal{I}$ can be

[2] Note that such transformation might be exponential.

improved by removing (a, b) from $P^{\mathcal{I}}$: let $\mathcal{I}' = \mathcal{I} \setminus P(a, b)$. Then $\mathcal{I}' \models C_1 \sqsubseteq \exists P$ and $\mathcal{I}' <^P \mathcal{I}$, which contradicts with $\mathcal{I}$ being a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P\}; P)$. Hence, $\mathcal{I} \models \exists P \sqsubseteq C_1$. Now, assume $\mathcal{I} \not\models \mathsf{Funct}(P)$, that is, there exist two tuples $(a, b) \in P^{\mathcal{I}}$, $(a, b') \in P^{\mathcal{I}}$, $b \neq b'$. Again, $\mathcal{I}$ can be improved by removing one of these tuples from $P^{\mathcal{I}}$, which contradicts with $\mathcal{I}$ being a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P\}; P)$. Thus, $\mathcal{I} \models \mathsf{Funct}(P)$.

Let $\mathcal{I}$ be a model of $\{C_1 \equiv \exists P, \mathsf{Funct}(P)\}$. Then it is a model of $C_1 \sqsubseteq \exists P$. Let us show it is minimal relative to P: no tuple can be removed from $P^{\mathcal{I}}$ without violating the axiom $C_1 \sqsubseteq \exists P$. By contradiction, assume that $(a, b) \in P^{\mathcal{I}}$ can be removed while still satisfying the axiom $C_1 \sqsubseteq \exists P$. Then, there must exist another tuple $(a, b') \in P^{\mathcal{I}}$ such that $b \neq b'$, which contradicts that $\mathcal{I} \models \mathsf{Funct}(P)$. Hence, $\mathcal{I}$ is minimal relative to P. □

3. $\mathsf{Circ}(\{C_2 \sqsubseteq \exists P^-\}; P) \equiv \{C_2 \equiv \exists P^-, \mathsf{Funct}(P^-)\}$

Proof. Similar to 1. □

Conversely, if we combine case 1 and case 2, circumscription does not entail equivalences for the domain and the range of P:

$$\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P) \not\models C_1 \equiv \exists P$$
$$\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P) \not\models C_2 \equiv \exists P^-$$

as an interpretation $\mathcal{I}$ that for each element $c_1 \in {C_1}^{\mathcal{I}}$ contains a tuple $(c_1, f(c_1)) \in P^{\mathcal{I}}$ and for each element $c_2 \in {C_2}^{\mathcal{I}}$ contains a tuple $(f(c_2), c_2) \in P^{\mathcal{I}}$, where f is a bijection and $P^{\mathcal{I}}$ contains nothing else, is a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P)$.

However, we can entail a weaker statement. Below we actually compute circumscription of P in the TBox $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$.

Proposition 3. *Let P be a role name, C_1, C_2 arbitrary DL concepts (not necessarily DL-Lite$^{\mathcal{H}}_{core}$) such that $P \notin \Sigma(\{C_1, C_2\})$.*

Then $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P)$ is equivalent to the following TBox Π:

$$\begin{array}{ll} C_1 \sqsubseteq \exists P & \exists P.\neg C_2 \sqsubseteq C_1 \quad \textit{(DRC)} \\ C_2 \sqsubseteq \exists P^- & \geq 2\, P.\neg C_2 \sqsubseteq \bot \quad \textit{(F1a)} \\ & \geq 2\, P^-.\neg C_1 \sqsubseteq \bot \quad \textit{(F1b)} \end{array}$$

$$\begin{array}{rl} \exists P.C_2 \sqcap \exists P.\neg C_2 & \sqsubseteq \bot \quad \textit{(F2a)} \\ \exists P^-.C_1 \sqcap \exists P^-.\neg C_1 & \sqsubseteq \bot \quad \textit{(F2b)} \\ \geq 2\, P \sqcap \exists P.(\geq 2\, P^-) & \sqsubseteq \bot \quad \textit{(NZa)} \end{array}$$

Before proving the above result, we provide an intuitive explanation of the axioms in Π (cf. Figure 1). Axioms (DRC), (F1a-b), (F2a-b), and (NZa) encode minimality of P. Intuitively, axiom (DRC) closes the domain and the range of P by saying that P cannot connect an object lying outside C_1 with an object lying outside C_2. Axiom (F1a) asserts local functionality of P: an object cannot have two successors that are not in C_2. Axiom (F1b) says the same about the inverse P^- and C_1. Axioms (F2a) and (F2b) can also be seen as a sort of functionality restrictions: axiom (F2a) states that if an object has a P-successor in C_2, then it cannot have a second P-successor not in C_2; axiom (F2b) states the same about P^- and C_1. Finally, axiom (NZa) assures that P does not form a zigzag: it says that there cannot exist an object that has at least two P successors, and one of its successors has at least two P-predecessors.

Interpretations forbidden by axioms (DRC), (F1a), (F2a), and (NZa) are depicted in Figure 1. Dots denote objects, edges denote P connections and ovals denote the extensions of classes C_1 and C_2.

Proof. ($\Rightarrow$) Let $\mathcal{I}$ be a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P)$. It means that $\mathcal{I}$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$ and it is minimal relative to P. We show that $\mathcal{I}$ is a model of Π, i.e., satisfies axioms (DRC), (F1a-b), (F2a-b), and (NZa).

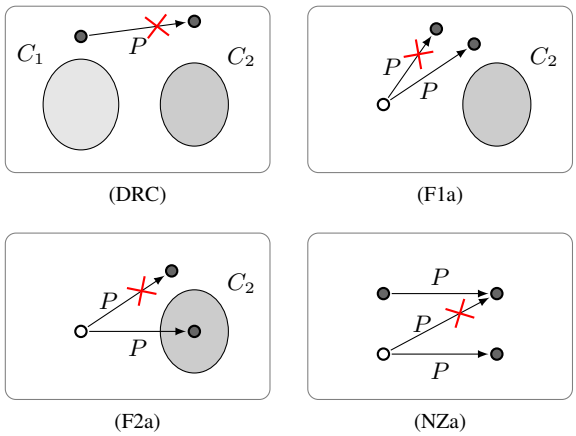


Figure 1. Interpretations forbidden by axioms (DRC), (F1a), (F2a) and (NZa). White objects denote elements whose existence is ruled out by the axioms. Crossed out edges can be deleted to improve the interpretations.

First, assume by contradiction that $\mathcal{I}$ does not satisfy axiom (DRC): $\mathcal{I} \not\models \exists P.\neg C_2 \sqsubseteq C_1$. Then, there should exist a tuple $(a, b) \in P^{\mathcal{I}}$ such that $b \in (\neg C_2)^{\mathcal{I}}$ and $a \notin (C_1)^{\mathcal{I}}$. Hence, $b \notin {C_2}^{\mathcal{I}}$ and $\mathcal{I}$ can be improved: $\mathcal{I}' = \mathcal{I} \setminus P(a, b)$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$ and $\mathcal{I}' <^P \mathcal{I}$. Contradiction with $\mathcal{I}$ being minimal relative to P.

Next, assume that $\mathcal{I}$ does not satisfy axiom (F1a), i.e., $\mathcal{I} \not\models \geq 2\, P.\neg C_2 \sqsubseteq \bot$. That means there exist elements a, b, and b' such that $b \neq b'$, $(a, b) \in P^{\mathcal{I}}$, $(a, b') \in P^{\mathcal{I}}$, and $b \in (\neg C_2)^{\mathcal{I}}$, $b' \in (\neg C_2)^{\mathcal{I}}$. Again, we can improve $\mathcal{I}$ by removing one of the tuples (a, b) or (a, b') from $P^{\mathcal{I}}$, which contradicts that $\mathcal{I}$ is minimal relative to P. Hence, $\mathcal{I}$ is a model of axiom (F1a). It can be shown similarly that $\mathcal{I}$ is a model of axiom (F1b).

Now, we prove that $\mathcal{I}$ satisfies axiom (F2a), i.e., $\mathcal{I} \models \exists P.C_2 \sqcap \exists P.\neg C_2 \sqsubseteq \bot$. Assume the contrary, i.e., for some elements a, b, and b', $(a, b) \in P^{\mathcal{I}}$, $(a, b') \in P^{\mathcal{I}}$, $b \in {C_2}^{\mathcal{I}}$, and $b' \notin {C_2}^{\mathcal{I}}$. Then it is easy to see that $\mathcal{I}$ is not minimal relative to P: $\mathcal{I}' = \mathcal{I} \setminus P(a, b')$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$ and $\mathcal{I}' <^P \mathcal{I}$. Contradiction, therefore $\mathcal{I}$ is a model of axiom (F2a). Satisfaction of axiom (F2b) can be proved analogously.

Finally, we show that $\mathcal{I}$ satisfies axiom (NZa), that is $\mathcal{I} \models \geq 2\, P \sqcap \exists P.(\geq 2\, P^-) \sqsubseteq \bot$. Assume the contrary, that is for some elements a, a', b, and b', $b \neq b'$, $(a, b) \in P^{\mathcal{I}}$, $(a, b') \in P^{\mathcal{I}}$ ($a \in (\geq 2\, P)^{\mathcal{I}}$), and $a \neq a'$, $(a', b) \in P^{\mathcal{I}}$ ($b \in (\geq 2\, P^-)^{\mathcal{I}}$). Obviously, $\mathcal{I}$ is not minimal relative to P: $\mathcal{I}' = \mathcal{I} \setminus P(a, b)$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$ and $\mathcal{I}' <^P \mathcal{I}$. Contradiction with $\mathcal{I}$ being minimal relative to P. Therefore, axiom (NZa) is satisfied by $\mathcal{I}$.

($\Leftarrow$) Let $\mathcal{I}$ be a model of Π. Then $\mathcal{I}$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$. Hence, to prove that $\mathcal{I}$ is a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}; P)$ it remains to show that it is minimal relative to P.

By contradiction, assume that $\mathcal{I}$ is not minimal, that is, there exists a tuple $(a, b) \in P^{\mathcal{I}}$ such that the interpretation $\mathcal{I}' = \mathcal{I} \setminus P(a, b)$ is a model of $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^-\}$ and $\mathcal{I}' <^P \mathcal{I}$. There are four cases:

1. $a \notin {C_1}^{\mathcal{I}}, b \notin {C_2}^{\mathcal{I}}$. Contradiction with axiom (DRC), $\exists P.\neg C_2 \sqsubseteq C_1$.
2. $a \notin {C_1}^{\mathcal{I}}, b \in {C_2}^{\mathcal{I}}$. By the assumption that (a, b) can be removed from the interpretation of P while satisfying $C_2 \sqsubseteq \exists P^-$, there

must exist a tuple $(a', b) \in P^{\mathcal{I}}$ with $a' \neq a$. Now, if $a' \notin {C_1}^{\mathcal{I}}$, then it contradicts $\mathcal{I} \models \geq 2\, P^{-}.\neg C_1 \sqsubseteq \bot$, and if $a' \in {C_1}^{\mathcal{I}}$, it contradicts $\mathcal{I} \models \exists P^{-}.C_1 \sqcap \exists P^{-}.\neg C_1 \sqsubseteq \bot$.

3. $a \in {C_1}^{\mathcal{I}}$, $b \notin {C_2}^{\mathcal{I}}$. Symmetric to the previous case.
4. $a \in {C_1}^{\mathcal{I}}$, $b \in {C_2}^{\mathcal{I}}$. By the assumption, (a, b) can be removed from $P^{\mathcal{I}}$. To satisfy $C_1 \sqsubseteq \exists P$, $C_2 \sqsubseteq \exists P^{-}$, there must exist two tuples $(a', b) \in P^{\mathcal{I}}$ and $(a, b') \in P^{\mathcal{I}}$ with $a \neq a'$ and $b \neq b'$. Then $a \in (\geq 2\, P)^{\mathcal{I}}$ and $b \in (\geq 2\, P^{-})^{\mathcal{I}}$. Contradiction with $\mathcal{I} \models \geq 2\, P \sqcap \exists P.(\geq 2\, P^{-}) \sqsubseteq \bot$.

In every case we derive a contradiction. Therefore, $\mathcal{I}$ is minimal, and hence, is a model of $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^{-}\}; P)$. □

Notice that the resulting TBox Π is no longer a *DL-Lite*$^{\mathcal{H}}_{core}$ TBox. The minimal language required is that of $\mathcal{ALCIQ}$.

Let us denote by $\mathsf{min}_{core}(P, C_1, C_2)$ the set formed by axioms (DRC), (F1a-b), (F2a-b), and (NZa) as a function of role P and concepts C_1 and C_2. Now, we can add to the TBox a role inclusion $R \sqsubseteq P$ and compute circumscription of P in a similar fashion. To address the additional role inclusion we make sure that the part of P disjoint from R is minimal. Note also that R does not have to satisfy axioms (DRC), (F1a-b), (F2a-b), and (NZa).

Proposition 4. *Let P be a role name, C_1, C_2 arbitrary DL concepts (not necessarily DL-Lite$^{\mathcal{H}}_{core}$) and R an arbitrary DL role such that $P \notin \Sigma(\{C_1, C_2, R\})$.*

Then $\mathsf{Circ}(\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^{-}, R \sqsubseteq P\}; P)$ is equivalent to the following TBox Π:

$$\begin{array}{ll} C_1 \sqsubseteq \exists P & \mathsf{min}_{core}(P', C_1 \sqcap \neg\exists R, C_2 \sqcap \neg\exists R^{-}) \\ C_2 \sqsubseteq \exists P^{-} & P \equiv P' \sqcup R \end{array}$$

where P' is a fresh role name and the Boolean constructors on roles are defined similarly to the Boolean constructors on concepts.

Note that though the axiom $P' \sqsubseteq \neg R$ is not explicitly asserted in Π, it is implied by Π. So, in fact it is not necessary to use a new name P' and we can replace each occurrence of P' by $P \sqcap \neg R$. Note also that in this case Π is an $\mathcal{ALCHIQ}$ plus union of roles TBox.

For the general case, it remains to consider inclusions of the form $\exists P^{-} \sqsubseteq \exists P$, $\exists P \sqsubseteq \exists P^{-}$, and $P^{-} \sqsubseteq P$. Interestingly, the former two inclusions act as inclusions positive w.r.t. P, i.e., inclusions where P occurs positively as $\exists P$, $\exists P^{-}$, P, or P^{-} (recall the normal form of concept inclusion axioms), whereas the latter inclusion acts as an inclusion negative w.r.t. P, i.e., inclusions where P occurs negatively as $\neg\exists P$, $\neg\exists P^{-}$, $\neg P$, or $\neg P^{-}$. Therefore, for a *DL-Lite*$^{\mathcal{H}}_{core}$ TBox $\mathcal{T}$, define $\mathsf{Pos}^{*}_{\mathcal{T}}(P)$ to be the set of all inclusions implied by $\mathcal{T}$ and positive w.r.t. P, or inclusions in $\mathcal{T}$ of the form $\exists P^{-} \sqsubseteq \exists P$, $\exists P \sqsubseteq \exists P^{-}$ if $\mathcal{T} \not\models P^{-} \sqsubseteq P$, and $\mathsf{Neg}^{*}_{\mathcal{T}}(P)$ to be the set of inclusions in $\mathcal{T}$ negative w.r.t. P, or inclusion $P^{-} \sqsubseteq P$ if $\mathcal{T} \models P^{-} \sqsubseteq P$. Finally, circumscription of an atomic role P in an arbitrary *DL-Lite*$^{\mathcal{H}}_{core}$ TBox can be computed as follows.

Theorem 5. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{core}$ TBox and P an atomic role. Further, let $\mathsf{Pos}^{*}_{\mathcal{T}}(P)$ be the set of the form*

$$\{R_i \sqsubseteq P\}_{i=0}^{m} \cup \{B_i \sqsubseteq \exists P\}_{i=0}^{n} \cup \{B'_i \sqsubseteq \exists P^{-}\}_{i=0}^{l},$$

(without loss of generality we can assume that P^{-} does not appear on the right-hand side of role inclusions in $\mathsf{Pos}^{*}_{\mathcal{T}}(P)$ and it does not contain inclusions of the form $X \sqsubseteq X$, where X is the domain or the range of P, or P itself). Then $\mathsf{Circ}(\mathcal{T}; P)$ can be computed as the union of $\mathcal{T}$ and the TBox Π:

$$\begin{array}{l} C_1 \equiv (B_1 \sqcup \cdots \sqcup B_n) \sqcap \neg(\exists R_1 \sqcup \cdots \sqcup \exists R_m) \\ C_2 \equiv (B'_1 \sqcup \cdots \sqcup B'_l) \sqcap \neg(\exists R_1^{-} \sqcup \cdots \sqcup \exists R_m^{-}) \\ \mathsf{min}_{core}(P', C_1, C_2) \\ P \equiv P' \sqcup R_1 \sqcup \cdots \sqcup R_m \end{array}$$

with P' a fresh atomic role, and C_1 and C_2 fresh atomic concepts. Note that here the empty union of concepts is equivalent to the bottom concept $\bot$.

Proof. By the properties of circumscription it holds that $\mathsf{Circ}(\mathcal{T}; P) = \mathsf{Circ}(\mathcal{T}_P; P) \wedge \mathcal{T}'$, where $\mathcal{T}'$ is the set of inclusions in $\mathcal{T}^{*}$ that do not contain P and $\mathcal{T}_P = \mathcal{T} \setminus \mathcal{T}'$.

Let $\mathcal{T}^{*}_P$ be the deductive closure of $\mathcal{T}_P$. Clearly, $\mathsf{Circ}(\mathcal{T}_P; P) \equiv \mathsf{Circ}(\mathcal{T}^{*}_P; P)$. Next, $\mathcal{T}^{*}_P$ can be partitioned in the following way:

$$\mathcal{T}^{*}_P = \mathsf{Pos}^{*}_{\mathcal{T}_P}(P) \cup \mathsf{Neg}^{*}_{\mathcal{T}_P}(P),$$

and similarly to Propositions 3 and 4 it can be shown that $\mathsf{Neg}^{*}_{\mathcal{T}_P}(P) \cup \Pi$ is equivalent to $\mathsf{Circ}(\mathcal{T}^{*}_P; P)$. It follows that $\mathcal{T} \cup \Pi$ is equivalent to $\mathsf{Circ}(\mathcal{T}; P)$. □

4.2 Circumscribing a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox

In *DL-Lite*$^{\mathcal{H}}_{bool}$ inclusions positive w.r.t. a role P have the form:

$$\begin{array}{ll} R \sqsubseteq P, & C \sqsubseteq \exists P, \\ C \sqsubseteq \exists P \sqcup \exists P^{-}, & C \sqsubseteq \exists P^{-}, \end{array}$$

where R is a basic role and C is a complex concept.

In order to be able to compute circumscription of a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox it remains to address positive occurrences of P in inclusions of the form $C \sqsubseteq \exists P \sqcup \exists P^{-}$. It turns out that circumscription of P in the TBox $\{C \sqsubseteq \exists P \sqcup \exists P^{-}\}$ is very similar to that in the TBox $\{C_1 \sqsubseteq \exists P, C_2 \sqsubseteq \exists P^{-}\}$ (see Proposition 3), with the difference that variations of axioms (F1a-b), (F2a-b), and (NZa) need to be added. More precisely, it is equivalent to the TBox:

$$\begin{array}{lrcll} & \exists P.\neg C_2 \sqcap \exists P^{-}.\neg C_1 & \sqsubseteq \bot & \textit{(F1c)} \\ & \exists P.C_2 \sqcap \exists P^{-}.\neg C_1 & \sqsubseteq \bot & \textit{(F2c)} \\ C \sqsubseteq \exists P \sqcup \exists P^{-} & \exists P.\neg C_2 \sqcap \exists P^{-}.C_1 & \sqsubseteq \bot & \textit{(F2d)} \\ \mathsf{min}_{core}(P, C, C) & \exists P^{-} \sqcap \exists P.(\geq 2\, P^{-}) & \sqsubseteq \bot & \textit{(NZb)} \\ & \geq 2\, P \sqcap \exists P.(\exists P) & \sqsubseteq \bot & \textit{(NZc)} \\ & \exists P^{-} \sqcap \exists P.(\exists P) & \sqsubseteq \bot & \textit{(NZd)} \end{array}$$

where C_1 and C_2 denote C. Let us denote by $\mathsf{min}_{bool}(P, C_1, C_2)$ the set formed by axioms (F1c), (F2c-d), and (NZb-d) as a function of role P and concepts C_1 and C_2. It will become clear later why we need to distinguish between C_1 and C_2 here.

Interpretations forbidden by the new axioms (F1c), (F2c), and (NZb-d) are depicted in Figure 2.

Now, when circumscribing an arbitrary *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox, some of these new axioms, e.g. (NZd), can contradict other TBox axioms, such as $\exists P^{-} \sqsubseteq \exists P$, therefore we cannot simply augment the theory with the new axioms to compute circumscription of a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox. To this purpose, we first transform the given TBox into an equivalent TBox, and then provide an algorithm to compute circumscription in the new TBox. This transformation exploits the fact that the following two TBoxes are equivalent to each other: $\{C \sqsubseteq \exists P \sqcup \exists P^{-}, \exists P^{-} \sqsubseteq \exists P\}$ and $\{C \sqsubseteq \exists P, \exists P^{-} \sqsubseteq \exists P\}$. More precisely, for a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox $\mathcal{T}$ and a role P, denote by $\mathcal{T}^{P,\sqcup}$ the TBox equivalent to $\mathcal{T}$ constructed as follows: if $\mathcal{T}$ implies

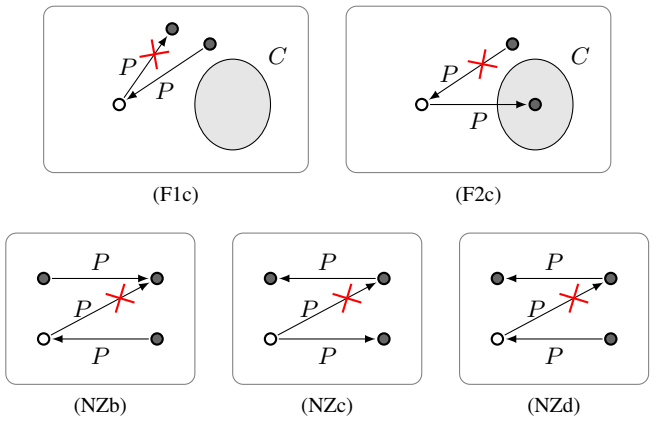


Figure 2. Interpretations forbidden by axioms (F1c), (F2c), and (NZb-d). White objects and crossed out edges are as in Figure 1.

$\exists P^- \sqsubseteq \exists P$ then replace axioms of the form $C \sqsubseteq \exists P \sqcup \exists P^-$ with $C \sqsubseteq \exists P$, and if $\mathcal{T}$ implies $\exists P \sqsubseteq \exists P^-$ then replace axioms of the form $C \sqsubseteq \exists P \sqcup \exists P^-$ with $C \sqsubseteq \exists P^-$. Next, define $\mathsf{Pos}^*_{\mathcal{T}}(P)$ to be the set of all *DL-Lite*$^{\mathcal{H}}_{core}$ inclusions implied by $\mathcal{T}$ and positive w.r.t. P, or inclusions in $\mathcal{T}$ of the form $C \sqsubseteq \exists P \sqcup \exists P^-$, or inclusions in $\mathcal{T}$ of the form $\exists P^- \sqsubseteq \exists P$, $\exists P \sqsubseteq \exists P^-$ if $\mathcal{T} \not\models P^- \sqsubseteq P$, and define $\mathsf{Neg}^*_{\mathcal{T}}(P)$ to be the set of inclusions in $\mathcal{T}$ negative w.r.t. P, or inclusion $P^- \sqsubseteq P$ if $\mathcal{T} \models P^- \sqsubseteq P$.

In the following theorem we compute circumscription of an atomic role in a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox.

Theorem 6. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{bool}$ TBox, P an atomic role, and $\mathcal{T}^{P,\sqcup}$ the transformation of $\mathcal{T}$ defined as above. Further, let $\mathsf{Pos}^*_{\mathcal{T}^{P,\sqcup}}(P)$ be the set of the form*

$$\begin{array}{lll} \{R_i \sqsubseteq P\}_{i=0}^m & \cup & \{C_i^* \sqsubseteq \exists P \sqcup \exists P^-\}_{i=0}^k \quad \cup \\ \{C_i \sqsubseteq \exists P\}_{i=0}^n & \cup & \{C_i' \sqsubseteq \exists P^-\}_{i=0}^l \end{array}$$

Then $\mathsf{Circ}(\mathcal{T}; P)$ can be computed as the union of $\mathcal{T}$ and the following TBox Π:

$$\begin{array}{lll} D_1 & \equiv & (C_1 \sqcup \cdots \sqcup C_n) \sqcap \neg(\exists R_1 \sqcup \cdots \sqcup \exists R_m) \\ D_2 & \equiv & (C_1' \sqcup \cdots \sqcup C_l') \sqcap \neg(\exists R_1^- \sqcup \cdots \sqcup \exists R_m^-) \\ D & \equiv & (C_1^* \sqcup \cdots \sqcup C_k^*) \sqcap \neg(\exists R_1 \sqcup \cdots \sqcup \exists R_m) \sqcap \\ & & \neg(D_1 \sqcup D_2) \sqcap \neg(\exists R_1^- \sqcup \cdots \sqcup \exists R_m^-) \\ P' & \equiv & P \sqcap \neg(R_1 \sqcup \cdots \sqcup R_m) \\ \multicolumn{3}{l}{\mathsf{min}_{core}(P', D_1 \sqcup D, D_2 \sqcup D)} \\ \multicolumn{3}{l}{\mathsf{min}_{bool}(P', D_1 \sqcup D, D_2 \sqcup D) \sqcap D} \end{array}$$

where P' is a fresh atomic role, D_1, D_2, and D are fresh atomic concepts, and $\mathsf{min}_{bool}(P', D_1 \sqcup D, D_2 \sqcup D) \sqcap D$ denotes the set of axioms of the form $D \sqcap C_l \sqsubseteq C_r$ for each axiom $C_l \sqsubseteq C_r$ in $\mathsf{min}_{bool}(P', D_1 \sqcup D, D_2 \sqcup D)$.

4.3 Adding an ABox

To fully address the problem of computing circumscription w.r.t. a single predicate in *DL-Lite*$^{\mathcal{H}}_{bool}$, it remains to add an ABox to the theory.

First, we show how to compute circumscription of a role or a concept in an ABox.

Proposition 7. *Let $\mathcal{A}$ be a DL-Lite$^{\mathcal{H}}_{bool}$ ABox. Then circumscription of a predicate X in $\mathcal{A}$ is equivalent to the KB $\langle \mathcal{T}_{\bar{X}}, \mathcal{A} \rangle$, where*

- *if X is an atomic concept A and $\{a_1, \ldots, a_n\} = \{a \mid A(a) \in \mathcal{A}\}$, then $\mathcal{T}_{\bar{A}} = \{A \sqsubseteq \{a_1, \ldots, a_n\}\}$.*
- *if X is an atomic role P, for individuals a and b, k_a denotes the number of P-successors of a in $\mathcal{A}$, k_b denotes the number of P-predecessors of b in $\mathcal{A}$, $\{a_1, \ldots, a_n\} = \{a \mid \mathcal{A} \models \exists P(a)\}$ and $\{b_1, \ldots, b_m\} = \{b \mid \mathcal{A} \models \exists P^-(b)\}$, then $\mathcal{T}_{\bar{P}}$ is the following TBox:*

$$\begin{array}{c} \left\{\{a\} \sqsubseteq \leq k_a\, P \mid \mathcal{A} \models \exists P(a)\right\} \cup \\ \left\{\{b\} \sqsubseteq \leq k_b\, P^- \mid \mathcal{A} \models \exists P^-(b)\right\} \cup \\ \left\{\exists P \sqsubseteq \{a_1, \cdots, a_n\}\,, \exists P^- \sqsubseteq \{b_1, \cdots, b_m\}\right\} \end{array}$$

Intuitively, the TBox $\mathcal{T}_{\bar{X}}$ encodes the closure of the predicate X. It does so by using nominals and number restrictions for the case of a role name.

Finally, we are ready to compute circumscription in a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB.

Theorem 8. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$^{\mathcal{H}}_{bool}$ KB and X a concept or role name. Let $\mathcal{A}'$ be the ABox obtained from $\mathcal{A}$ by renaming each occurrence of X to a fresh predicate X', $\mathcal{A}' = \mathcal{A}[X/X']$, and $\mathcal{T}' = \mathcal{T} \cup \{X' \sqsubseteq X\}$.*

Then $\mathsf{Circ}(\langle \mathcal{T}, \mathcal{A} \rangle; X)$ is equivalent to $\langle \mathsf{Circ}(\mathcal{T}'; X) \cup \mathcal{T}_{\bar{X}'}, \mathcal{A}' \rangle$.

5 Checking Entailment in Circumscribed *DL-Lite*$^{\mathcal{H}}_{bool}$

In the previous section we showed that for a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB $\mathcal{K}$ and a role P, $\mathsf{Circ}(\mathcal{K}; P)$ is not a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB anymore. It requires the language of $\mathcal{ALCHOIQ}$ with union of roles. Reasoning in $\mathcal{ALCHOIQ}$ extended with Boolean constructors on roles can be reduced to reasoning in $\mathcal{SHOIQB}_s$, which is an extension of $\mathcal{SHOIQ}$ with arbitrary Boolean constructors on simple roles and has been shown to be NEXPTIME-complete in [29].

On the other hand, if we only want to check concept or role subsumption in a circumscribed *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox $\mathcal{T}$, then the check can be done by encoding the problem in $\mathcal{ALCQIb}_{reg}$, which has been shown to be EXPTIME-complete (see [9]). However, in most of the cases, the complexity of checking whether $\mathsf{Circ}(\mathcal{T}; P) \models X_1 \sqsubseteq X_2$ for *DL-Lite*$^{\mathcal{H}}_{bool}$ concepts or roles X_1, X_2 is in NP, i.e., is does not exceed the complexity of *DL-Lite*$^{\mathcal{H}}_{bool}$:

a) if $P \notin \Sigma(X_1, X_2)$, $\mathsf{Circ}(\mathcal{T}; P) \models X_1 \sqsubseteq X_2$ iff $\mathcal{T} \models X_1 \sqsubseteq X_2$,
b) if $P \in \Sigma(X_2)$, $\mathsf{Circ}(\mathcal{T}; P) \models X_1 \sqsubseteq X_2$ iff $\mathcal{T} \models X_1 \sqsubseteq X_2$,
c) if $P \in \Sigma(X_1)$

1) if $\mathcal{T}$ does not contain inclusions of the form $C_1 \sqsubseteq \exists P$, $C_2 \sqsubseteq \exists P^-$, and $C \sqsubseteq \exists P \sqcup \exists P^-$, then $\mathsf{Circ}(\mathcal{T}; P)$ is a *DL-Lite*$^{\mathcal{H}}_{bool}$ with union of roles KB and the entailment can be checked using, e.g., the algorithm for $\mathcal{ALCQIb}_{reg}$ (see [9]),
2) if $\mathcal{T}$ contains inclusions of the form $C_1 \sqsubseteq \exists P$ but not $C_2 \sqsubseteq \exists P^-$ and $C \sqsubseteq \exists P \sqcup \exists P^-$, then
   - $\mathsf{Circ}(\mathcal{T}; P) \models X_1 \sqsubseteq X_2$ iff $\mathcal{T} \models X_1 \sqsubseteq X_2$ if $X_1 = \exists P^-$ or $X_1 = P$, and
   - $\mathsf{Circ}(\mathcal{T}; P) \models \exists P \sqsubseteq X_2$ iff $\mathcal{T} \models \exists P \sqsubseteq X_2$ or $\mathcal{T} \models D \sqsubseteq X_2$, where $D = \bigsqcup_{i=1}^n D_i$, $\mathcal{T} \models D_i \sqsubseteq \exists P$ and n is the maximal such number.
3) if $\mathcal{T}$ contains inclusions of the form $C_2 \sqsubseteq \exists P^-$ but not $C_1 \sqsubseteq \exists P$ and $C \sqsubseteq \exists P \sqcup \exists P^-$, then this is symmetric to the previous case.

4) if $\mathcal{T}$ contains both inclusions of the form $C_1 \sqsubseteq \exists P$ and $C_2 \sqsubseteq \exists P^-$, or $C \sqsubseteq \exists P \sqcup \exists P^-$, then $\mathsf{Circ}(\mathcal{T}; P) \models X_1 \sqsubseteq X_2$ iff $\mathcal{T} \models X_1 \sqsubseteq X_2$.

For an atomic concept A, $\mathsf{Circ}(\mathcal{K}; A)$ is a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB and the entailment check can be done in NP.

In most of the cases the complexity of checking entailment does not exceed that of *DL-Lite*$^{\mathcal{H}}_{bool}$ (i.e., in NP). As for the case c)-1), the complexity of checking entailment in $\mathcal{ALCQIb}_{reg}$ is EXPTIME. The exact complexity of *DL-Lite*$^{\mathcal{H}}_{bool}$ with union of roles is unknown and lies between NP and EXPTIME.

6 Conclusions

We have studied circumscribed *DL-Lite* and addressed the problem of computing circumscription in *DL-Lite*$^{\mathcal{H}}_{bool}$ KBs. We computed circumscription of a single predicate (a concept or a role) in a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB, which turned out to be first-order expressible. We showed that circumscription of a concept in a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox is a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox, whereas circumscription of a role a *DL-Lite*$^{\mathcal{H}}_{bool}$ TBox is an $\mathcal{ALCHIQ}$ plus union of roles TBox. Moreover adding an ABox to the circumscribed theory requires nominals in the language. We also showed that checking entailment of concept or role inclusions in a circumscribed KB can be done in EXPTIME.

To fully address the problem of circumscribing *DL-Lite*$^{\mathcal{H}}_{bool}$, we need to consider multiple minimized predicates and varying predicates. It is quite straightforward to compute prioritized circumscription of a set of concepts with strict priority as follows: first, circumscribe the concept with the highest priority; then, circumscribe the concept with the second priority in the result of the first circumscription; and continue by analogy. Conversely, parallel circumscription and varied predicates require more investigation.

Another interesting point is to study the exact complexity of checking entailment in *DL-Lite*$^{\mathcal{H}}_{bool}$ with Boolean constructors on roles. In the existing literature on complex role constructors only expressive DLs starting from $\mathcal{ALC}$ are considered. Therefore, analysis of the exact complexity of a low complexity logic such as *DL-Lite*$^{\mathcal{H}}_{bool}$ combined with Boolean constructors on roles could result in a better bound than EXPTIME.

REFERENCES

[1] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev, 'The *DL-Lite* family and relations', *J. of Artificial Intelligence Research*, **36**, 1–69, (2009).

[2] *The Description Logic Handbook: Theory, Implementation and Applications*, eds., Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, Cambridge University Press, 2003.

[3] Franz Baader and Bernhard Hollunder, 'Embedding defaults into terminological knowledge representation formalisms', *J. of Automated Reasoning*, **14**, 149–180, (1995).

[4] Piero Bonatti, Marco Faella, and Luigi Sauro, '$\mathcal{EL}$ with default attributes and overriding', in *Proc. of ISWC 2010*, (November 2010).

[5] Piero A. Bonatti, Marco Faella, and Luigi Sauro, 'Defeasible inclusions in low-complexity dls', *J. of Artificial Intelligence Research*, **42**, 719–764, (2011).

[6] Piero A. Bonatti, Carsten Lutz, and Frank Wolter, 'The complexity of circumscription in description logics', *J. of Artificial Intelligence Research*, **35**, 717–773, (2009).

[7] Katarina Britz, Johannes Heidema, and Tommie Meyer, 'Modelling object typicality in description logics', in *Proc. of DL 2009*, (2009).

[8] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family', *J. of Automated Reasoning*, **39**(3), 385–429, (2007).

[9] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz, 'Answering regular path queries in expressive description logics: An automata-theoretic approach', in *Proc. of AAAI 2007*, pp. 391–396, (2007).

[10] Giovanni Casini and Umberto Straccia, 'Rational closure for defeasible description logics', in *Proc. of JELIA 2010*, pp. 77–90, (2010).

[11] R. Cote, D. Rothwell, J. Palotay, R. Beckett, and L. Brochu, 'The systematized nomenclature of human and veterinary medicine: SNOMED International', in *Northfield, IL: College of American Pathologists*, (1993).

[12] Francesco M. Donini, Daniele Nardi, and Riccardo Rosati, 'Autoepistemic description logics', in *Proc. of IJCAI'97*, pp. 136–141, (1997).

[13] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato, 'Preferential description logics', in *Proc. of LPAR 2007*, pp. 257–272, (2007).

[14] Guido Governatori, 'Defeasible description logic', in *Proc. of RuleML 2004*, pp. 98–112. Springer, (2004).

[15] Stephan Grimm and Pascal Hitzler, 'A preferential tableaux calculus for circumscriptive $\mathcal{ALCO}$', in *Proc. of RR 2009*, pp. 40–54, (2009).

[16] Stijn Heymans and Dirk Vermeir, 'A defeasible ontology language', in *Proc. of the Confederated Int. Conf. DOA, CoopIS, and ODBASE 2002*, volume 2519 of *LNCS*, pp. 1033–1046. Springer, (2002).

[17] Peihong Ke and Ulrike Sattler, 'Next steps for description logics of minimal knowledge and negation as failure', in *Proc. of DL 2008*, (2008).

[18] Phokion G. Kolaitis and Christos H. Papadimitriou, 'Some computational aspects of circumscription', *J. of the ACM*, **37**(1), 1–14, (January 1990).

[19] Daniel J. Lehmann and Menachem Magidor, 'What does a conditional knowledge base entail?', *Artificial Intelligence*, **55**(1), 1–60, (1992).

[20] Vladimir Lifschitz, 'Computing circumscription', in *Proc. of IJCAI'85*, (1985).

[21] Vladimir Lifschitz, 'Circumscription', in *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, 298–352, Oxford University Press, (1994).

[22] Vladimir Lifschitz, 'Minimal belief and negation as failure', *Artificial Intelligence*, **70**, 53–72, (1994).

[23] John McCarthy, 'Circumscription — a form of non-monotonic reasoning', *Artificial Intelligence*, **13**, 27–39,171–172, (1980).

[24] Boris Motik and Riccardo Rosati, 'A faithful integration of description logics with logic programming', in *Proc. of IJCAI 2007*, pp. 477–482, (2007).

[25] Donald Nute, 'Defeasible logic', in *Proc. of the 14th Int. Conf. on Applications of Prolog (INAP 2001)*, pp. 87–114, (2001).

[26] Alan Rector, 'Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases', in *Proc. of the Pacific Symposium on Biocomputing (PSB 2004)*, pp. 226–237, (2004).

[27] Alan L. Rector and Ian R. Horrocks, 'Experience building a large, reusable medical ontology using a description logic with transitivity and concept inclusions', in *In Proc. of the Workshop on Ontological Engineering, AAAI Spring Symposium*. AAAI Press, (1997).

[28] Raymond Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**, 81–132, (1980).

[29] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler, 'Cheap boolean role constructors for description logics', in *Proc. of JELIA 2008*, volume 5293 of *LNCS*, pp. 362–374. Springer, (2008).

[30] Fabrizio Sebastiani and Umberto Straccia, 'Default reasoning in a terminological logic', *Computers and Artificial Intelligence*, **14**(3), (1995).

[31] Andrzej Uszok, Jeffrey M. Bradshaw, Renia Jeffers, Niranjan Suri, Patrick J. Hayes, Maggie R. Breedy, Larry Bunch, Matt Johnson, Shriniwas Kulkarni, and James Lott, 'KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement', in *Proc. of the 4th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, pp. 93–98, (2003).

[32] Kewen Wang, David Billington, Jeff Blee, and Grigoris Antoniou, 'Combining description logic and defeasible logic for the semantic web', in *Proc. of RuleML 2004*, pp. 170–181, (2004).

[33] Heng Zhang and Mingsheng Ying, 'Decidable fragments of first-order language under stable model semantics and circumscription', in *Proc. of AAAI 2010*, (2010).

[34] Rui Zhang, Alessandro Artale, Fausto Giunchiglia, and Bruno Crispo, 'Using description logics in relation based access control', in *Proc. of DL 2009*, volume 477 of *CEUR*, `ceur-ws.org`, (2009).

Towards an operator for merging taxonomies

Amélie Cordier[1] and **Jean Lieber**[234] and **Julien Stevenot**[234]

Abstract. The merging of knowledge bases is a fundamental part of the collaboration in continuous knowledge construction. This paper introduces an operator for merging similar taxonomies, i.e. taxonomies that share the major part of their contents. Taxonomies have been chosen for the low time and space complexity of the classical inferences defined on them. A limit of this language is that it does not incorporate negations, thus the union of taxonomies is never inconsistent, though it is meaningful to consider that their merging does not coincide with their union. Thus, a way to extend the taxonomies' language is presented to allow the definition of a merging operator. This operator is algorithmically simple for the part of their contents on which the taxonomies agree, confining complexity to the part on which they do not. So it allows a low time and space complexity merging on similar taxonomies.

1 INTRODUCTION

This work is part of the Kolflow project.[5] Kolflow aims at investigating man-machine collaboration in continuous knowledge construction and this collaboration involves to make the conjunction of knowledge from different sources.

In [8], a continuous knowledge integration process (KCIP) is described in which semantic wikis are used as a way of representing knowledge. The semantic wiki used by Kolflow as use case for studying collaboration is WikiTaaable.[6] To simplify, the formal part of WikiTaaable can be seen here as a taxonomy, where a taxonomy is a concept hierarchy[7] organized by the subsumption relation.[8]

In KCIP, there is a common stable version of WikiTaaable available on a web site so that anyone can download it, work on it and make some updates to make its own version of the wiki. This process will produce, at the same time, several versions of the same wiki which use similar vocabularies but which do not necessarily agree on everything.For example, the case could happen that one version has been modified by someone and says "A melon is a fruit" whereas another one, modified by someone else, says "A melon is a vegetable" (and the two knowledge bases share the concepts `Vegetable` and `Fruit`) as modelled in the figure 1 (where $\sqsubseteq$ is represented by an arrow).

In the current KCIP, both of these modifications will be included in a new version of WikiTaaable and submitted to the expert community[9] and will be rejected if the experts consider that melons are either not fruits or not vegetables and all the other modifications possibly done at the same time will be lost.

So the merging of "A melon is a fruit" and "A melon is a vegetable" raises a problem. Indeed, if someone knows the concept `Fruit` and says that melons are vegetables without saying that melons are fruits, he/she probably means that melons *are not* fruits.

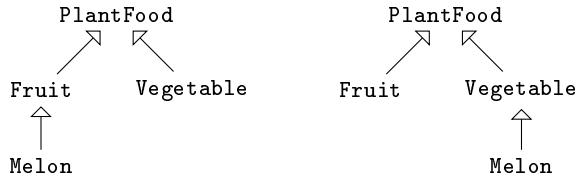


Figure 1: Two taxonomies waiting to be merged.

The taxonomies form one of the simplest knowledge representation language and as such are interesting to study and to use because of the low time and space complexity of their classical inferences. But with the classical semantics, the conjunction of two taxonomies, i.e. the union of their formulas, cannot be inconsistent and, as such, cannot express all that a human could express like "Melons are not fruits". For example, the conjunction of the two taxonomies seen in figure 1 is not inconsistent, it just means that melons are, at the same time, fruits and vegetables, as presented in figure 2.

So how to make arise some inconsistencies during the merging? A way of solving this issue is to increase the expressivity of the representation language but without significantly increasing its time and space complexity. To achieve this goal, this paper proposal is to add an axiom construct for modelling that melons are not fruits, in the case where a concept `Fruit` exists with the axiom `Melon` $\sqsubseteq$ `Vegetable` but without the axiom `Melon` $\sqsubseteq$ `Fruit`.

[1] Université de Lyon 1, CNRS, LIRIS, UMR5205, F-69622, France, email: amelie.cordier@liris.cnrs.fr
[2] Université de Lorraine, LORIA, UMR 7503 — Vandœuvre-lès-Nancy, F-54506, France
[3] CNRS, LORIA, UMR 7503 — Vandœuvre-lès-Nancy, F-54506, France
[4] Inria — Villers-lès-Nancy, F-54602, France
[5] Kolflow (`http://kolflow.univ-nantes.fr`, code: ANR-10-CONTINT-025) is supported by the French National Research agency (ANR) and is part of the CONTINT research program.
[6] `http://wikitaaable.loria.fr`
[7] A concept represents a class of objects. For example `Banana` is the concept representing the set of all the bananas.
[8] The subsumption between two concepts indicates the inclusion between the classes of objects they represent. It is denoted by $\sqsubseteq$. For example, the formula `Banana` $\sqsubseteq$ `Fruit` represents the knowledge bananas are fruits (the set of bananas is a subset of the set of fruits).
[9] Some steps of the KCIP are not presented here because not directly related to our subject. For more detailed information on this process see [8].

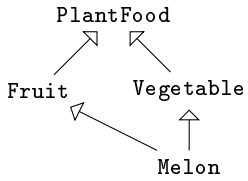


Figure 2: The consistent result of the union of the two taxonomies of figure 1.

With this addition, the conjunction of two taxonomies could raise some contradictions. An example of contradiction is: "A melon is a fruit but is not a vegetable" and "A melon is a vegetable but is not a fruit". So, a part of the modelled knowledge has to be suppressed, in order to restore consistency. But how one could determine which part should be suppressed and which part should be preserved? In [3], a measure of the agreement and the disagreement between ontologies, that could be useful to make some preferences between pieces of knowledge, is defined. Following the ideas of this work, the idea is to preserve all the agreement and to select some pieces of knowledge of the disagreement.

The paper is organized as follows.The notions and tools that are used in this paper are defined in section 2. Section 3 is the core of this paper: it presents an approach for merging taxonomies. Finally, a conclusion and some future work are presented in section 4.

2 BELIEF REVISION AND BELIEF MERGING

This section is about the minimal change theory research field in which this paper aims at contributing. Two important notions of this field are belief revision and belief merging.

2.1 Revision of a knowledge base by another one

Let ψ and μ be two consistent knowledge bases. The revision of ψ by μ consists in keeping all the knowledge from μ and the maximal knowledge from ψ to obtain a consistent knowledge base.

In [1], some general postulates of belief revision have been proposed. These postulates have been reformulated in [5] for the particular case of revision in propositional logic. According to these postulates, if the conjunction of ψ and μ is consistent, then the revision is equivalent to this conjunction. If $\psi \wedge \mu$ is inconsistent, then *minimal* modifications $\psi \mapsto \psi'$ have to be done such that $\psi' \wedge \mu$ is consistent (and the revision of ψ by μ is $\psi' \wedge \mu$). [7] presents a survey on belief revision.

2.2 Merging of knowledge bases

Let ψ_1, ψ_2, ..., ψ_n be n consistent knowledge bases. The merging of these knowledge bases consists in keeping as much as possible from them in order to obtain a consistent knowlege base. The difference with revision is that there is no a priori preference among the knowlege bases to be merged.

Let Δ be a merging operator. If the conjunction of all the knowledge bases ψ_1, ψ_2, ..., ψ_n is consistent, the result of the merging is their conjunction:

$$\Delta(\{\psi_1, \psi_2, \ldots, \psi_n\}) \equiv \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$$

Else, *minimal* modifications of all the bases $\psi_1 \mapsto \psi'_1$, $\psi_2 \mapsto \psi'_2$, ..., $\psi_n \mapsto \psi'_n$ such that $\psi'_1 \wedge \psi'_2 \wedge \ldots \wedge \psi'_n$ is consistent have to be done, and:

$$\Delta(\{\psi_1, \psi_2, \ldots, \psi_n\}) \equiv \psi'_1 \wedge \psi'_2 \wedge \ldots \wedge \psi'_n$$

Some postulates of merging, inspired from the postulates of revision, are presented in [6].

3 MERGING TAXONOMIES

3.1 Taxonomies

The term *taxonomy* has been created by biologists for talking about the classification of the species. But, etymologically, it means arrangement method and is used to refer to a class hierarchy. So, here the term is used for a class hierarchy which is represented formally by a language (called here $\mathcal{L}_\mathcal{T}$ for taxonomy's language).

$\mathcal{L}_\mathcal{T}$ is defined as follows (reusing the description logics notations [2]). Let $\mathcal{A}$ be a countable set: $A \in \mathcal{A}$ is called a concept (only atomic concepts are allowed in $\mathcal{L}_\mathcal{T}$). A formula of $\mathcal{L}_\mathcal{T}$ has the form $A \sqsubseteq B$ where A, $B \in \mathcal{A}$ and $A \neq B$,[10] meaning that the concept A is more specific than the concept B (formally: for each model ω of $A \sqsubseteq B$, $\omega(A) \subseteq \omega(B)$). A taxonomy is a knowledge base of $\mathcal{L}_\mathcal{T}$ (i.e., a finite set of $\mathcal{L}_\mathcal{T}$ formulas).

The vocabulary $\mathcal{V}(\psi)$ of a taxonomy ψ is defined as follows. For $A, B \in \mathcal{A}$, $\mathcal{V}(A \sqsubseteq B) = \{A, B\}$. For a taxonomy ψ, $\mathcal{V}(\psi) = \bigcup\{\mathcal{V}(f) \mid f \in \psi\}$.

The language $\mathcal{L}_\mathcal{T}$ has been chosen because it is one of the simplest knowledge representation languages and, as such, its inferences are of low complexity, i.e. the sumbsomption test is linear for $\mathcal{L}_\mathcal{T}$ (it can be completed by searching a directed path in a graph). So an efficient (in term of time and space complexity) merging operator should be definable in this language. And, moreover, this language is sufficient to express most of the formal knowledge edited in WikiTaaable.

3.2 The notion of inconsistencies in $\mathcal{L}_\mathcal{T}$

Let us consider ψ_1 and ψ_2, the two taxonomies in figures 3 and 4. ψ_1 states that melons are fruits and ψ_2 states that melons are vegetables. Formally there is no contradiction there: ψ_1 (resp., ψ_2) does not entail that melons are not vegetables (resp., fruits).

More generally, if ψ_1 and ψ_2 are two taxonomies (two finite subsets of $\mathcal{L}_\mathcal{T}$), $\psi_1 \cup \psi_2$ is also a taxonomy and therefore, is consistent.[11]

Now, when considering again ψ_1 and ψ_2 of figures 3 and 4, the fact that $\psi_1 \not\models$ `Melon` $\sqsubseteq$ `Vegetable` and $\psi_2 \not\models$ `Melon` $\sqsubseteq$ `Fruit` may have two intuitive interpretations:

[10] whithout loss of expressivity, the tautologies $A \sqsubseteq A$ are excluded from the formalism.

[11] Every taxonomy is satisfiable and thus consistent. Indeed, if $\psi = \{A_i \sqsubseteq B_i\}_i$ is a taxonomy, it is satisfied by the interpretation whose domain is $\{1\}$ and function ω associates, for any i, A_i to $\omega(A_i) = \{1\}$ and B_i to $\omega(B_i) = \{1\}$.

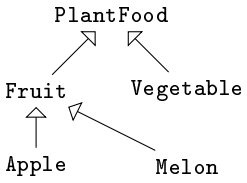


Figure 3: ψ_1.

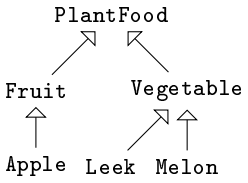


Figure 4: ψ_2.

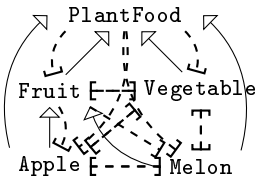


Figure 5: $\widehat{\psi_1}$, with the ψ_1 of figure 3.

- Either ψ_1 and ψ_2 are incomplete in the sense that the person in charge of the development of ψ_1 (resp., ψ_2) does not know whether melons are or are not vegetables (resp., fruits);
- Or the persons in charge of the development of ψ_1 and ψ_2 are in disagreement: the former thinks that melons are fruits and are not vegetables, the latter thinks that melons are vegetables and are not fruits.

Therefore the merging of ψ_1 and ψ_2 should lead to a taxonomy ψ satisfying one of the four possibilities:

(a) $\psi \models \texttt{Melon} \sqsubseteq \texttt{Fruit}$ and $\psi \models \texttt{Melon} \sqsubseteq \texttt{Vegetable}$
(b) $\psi \models \texttt{Melon} \sqsubseteq \texttt{Fruit}$ and $\psi \not\models \texttt{Melon} \sqsubseteq \texttt{Vegetable}$
(c) $\psi \not\models \texttt{Melon} \sqsubseteq \texttt{Fruit}$ and $\psi \models \texttt{Melon} \sqsubseteq \texttt{Vegetable}$
(d) $\psi \not\models \texttt{Melon} \sqsubseteq \texttt{Fruit}$ and $\psi \not\models \texttt{Melon} \sqsubseteq \texttt{Vegetable}$

Hence, if the conjunction of two taxonomies corresponds to their union, only situation (a) can occur. To prevent that situation, taxonomies are considered according to a closed world assumption (CWA):

$$\frac{\psi \not\models A \sqsubseteq B}{A \not\sqsubseteq B}\ \text{CWA}$$

This entails that the formulas $A \not\sqsubseteq B$ are considered. Let $\mathcal{L}_{\mathcal{T}}^{\neg}$ be the language of taxonomies with negations. A formula of $\mathcal{L}_{\mathcal{T}}^{\neg}$ is either a formula of $\mathcal{L}_{\mathcal{T}}$ or a formula $A \not\sqsubseteq B$ for $A, B \in \mathcal{A}$. The semantics of $\mathcal{L}_{\mathcal{T}}^{\neg}$ is as follows: ω satisfies $A \not\sqsubseteq B$ if $\omega(A) \not\sqsubseteq \omega(B)$.

In order to integrate the closed-world assumption in the conjunction, for ψ an $\mathcal{L}_{\mathcal{T}}^{\neg}$ knowledge base, let $\widehat{\psi}$ be the deductive closure (including CWA) of ψ defined by:

$$\begin{aligned}\widehat{\psi} = &\{A \sqsubseteq B \mid A, B \in \mathcal{V}(\psi) \text{ and } \psi \models A \sqsubseteq B\} \\ &\cup \{A \not\sqsubseteq B \mid A, B \in \mathcal{V}(\psi) \text{ and } \psi \not\models A \sqsubseteq B\}\end{aligned}$$

$\widehat{\psi}$ can be viewed as a clique whose vertices are elements of $\mathcal{V}(\psi)$ as illustrated on figure 5 where $A \not\sqsubseteq B$ is represented by a dashed bracket-headed arrow from A to B. For the sake of simplicity, in the next examples the deductive closure will not always be graphically represented.

Now, the conjunction of two taxonomies ψ_1 and ψ_2 (of $\mathcal{L}_{\mathcal{T}}$ or of $\mathcal{L}_{\mathcal{T}}^{\neg}$) is defined by:

$$\psi_1 \wedge \psi_2 = \widehat{\psi_1} \cup \widehat{\psi_2}$$

With this definition, the conjunction of the taxonomies of the figures 3 and 4 is inconsistent since, e.g.,$\{\texttt{Melon} \sqsubseteq \texttt{Fruit}, \texttt{Melon} \not\sqsubseteq \texttt{Fruit}\} \subseteq \psi_1 \wedge \psi_2$.

With that, the merging of these two taxonomies raises two inconsistencies (or *clashes*) that have to be solved:

$$clash_1 = \{\texttt{Melon} \sqsubseteq \texttt{Fruit},\quad \texttt{Melon} \not\sqsubseteq \texttt{Fruit}\}$$
$$clash_2 = \{\texttt{Melon} \sqsubseteq \texttt{Vegetable},\quad \texttt{Melon} \not\sqsubseteq \texttt{Vegetable}\}$$

3.3 $CS_\mu(\psi)$ and $MCS_\mu(\psi)$

Let μ and ψ be two $\mathcal{L}_{\mathcal{T}}^{\neg}$ knowledge bases, such that μ is consistent. Let $CS_\mu(\psi)$ be the set of knowledge bases φ such that $\mu \subseteq \varphi \subseteq \psi \cup \mu$ and φ is consistent (CS stands for "consistent subsets"). $CS_\mu(\psi) \neq \emptyset$ since $\mu \in CS_\mu(\psi)$. Among the elements of $CS_\mu(\psi)$, the largest ones for inclusion constitute $MCS_\mu(\psi)$ (MCS stands for maximal consistent subset). If $\psi \cup \mu$ is consistent, then $MCS_\mu(\psi) = \{\psi \cup \mu\}$.

For example (using the notations of the previous sections), if $\psi = clash_1 \cup clash_2$, then $MCS_\emptyset(\psi)$ is composed of the four consistent knowledge bases (a), (b), (c), and (d).

3.4 Modelling the choice among several possibilities

As pointed out above, there may be several possibilities and so, it is necessary to make a choice among them. This possibility to make a choice is represented by a preorder $\leq$ on the knowledge bases of $\mathcal{L}_{\mathcal{T}}^{\neg}$ such that $\psi_1 < \psi_2$ means that ψ_1 is preferred to ψ_2 ($\psi_1 < \psi_2$ means that $\psi_1 \leq \psi_2$ and $\psi_2 \not\leq \psi_1$).[12]

$\leq$ is assumed to be a total order up to the logical equivalence: it is reflexive and transitive, if $\psi_1 \leq \psi_2$ and $\psi_2 \leq \psi_1$ then ψ_1 and ψ_2 are equivalent, and for any ψ_1 and ψ_2, either $\psi_1 \leq \psi_2$ or $\psi_2 \leq \psi_1$. Therefore, if S is a finite set of $\mathcal{L}_{\mathcal{T}}^{\neg}$ knowledge bases, the minimal of S for $\leq$ exists and is unique, modulo equivalence, and it is denoted by $Min_{\leq}(S)$.

Moreover, $\leq$ is assumed to prefer more specific knowledge bases, i.e., if $\psi_1 \subseteq \psi_2$ then $\psi_2 \leq \psi_1$. This property involves that $Min_{\leq}(CS_\mu(\psi)) = Min_{\leq}(MCS_\mu(\psi))$.

3.5 An operator for merging taxonomies

The merging operator presented in this section is inspired from the ideas of agreement and disagreement of two ontologies as introduced in [3]. Let ψ_1, ψ_2, ..., ψ_n be n consistent

[12] As pointed out by a reviewer, another idea is to use the majority merging rule stating that a preference is given to the piece of knowledge entailed by a majority of the n knowledge bases to be merged (which makes sense if $n > 2$). However, in some situations, there is no strict majority (the number of knowledge bases entailing $A \sqsubseteq B$ is equal to the number of knowledge bases entailing $A \not\sqsubseteq B$) and the preorder $\leq$ can be used.

knowledge bases of $\mathcal{L}_{\mathcal{T}}^{\neg}$ (e.g., two taxonomies) and $E = \{\psi_1, \psi_2, \ldots, \psi_n\}$. The notions introduced below are illustrated with the taxonomies of figures 3 and 4.

The agreement α of $\psi_1, \psi_2, \ldots, \psi_n$ is constituted by the pieces of knowledge common to them. formally:

$$\alpha = \bigcap_i \widehat{\psi_i} = \widehat{\psi_1} \cap \widehat{\psi_2} \cap \ldots \cap \widehat{\psi_n}$$

α is necessary consistent (since $\alpha \subseteq \widehat{\psi_1}$ that is consistent). Figure 7 shows a representation of α.

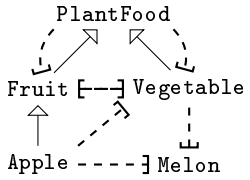


Figure 6: α: the agreement of the ψ_1 and ψ_2 of figures 3 and 4, represented without some of the edges that can be deduced by CWA.

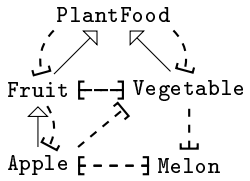


Figure 7: α: the agreement of the ψ_1 and ψ_2 of figures 3 and 4, represented without some of the edges that can be deduced by CWA.

The disagreement is intuitively defined as the pieces of knowledge that are not in agreement.[13] This disagreement is defined as $\delta = \bigcup_i \delta_i$ where δ_i represents the pieces of knowledge of ψ_i that are not in agreement with the ψ_j's ($j \neq i$):

$$\delta_i = \widehat{\psi_i} \setminus \alpha$$

Since ψ_i is consistent, δ_i is also consistent. Figures 8 and 9 illustrate δ_1 and δ_2.

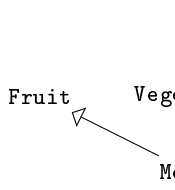


Figure 8: δ_1.

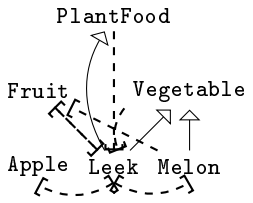


Figure 9: δ_2.

So, here, δ is the union of δ_1 and δ_2.

[13] This slightly differs from [3] where the agreement and the disagreement are not complementary.

Then, a subset β of δ has to be chosen. $\alpha \cup \beta$ has to be consistent and has to keep as much knowledge as possible, i.e. $\beta \in MCS_\alpha(\delta)$. If the choice is made according to $\leq$ (cf section 3.4) then:

$$\beta = Min_{\leq}(MCS_\alpha(\delta))$$

Finally, the result of the merging is a knowledge base of $\mathcal{L}_{\mathcal{T}}$ such that:

$$\widehat{\Delta(E)} = \widehat{\beta}$$

Figures 10 to 13 present the four possibilities for $\Delta(\psi_1, \psi_2)$, depending on the choice $\leq$.

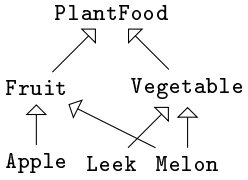


Figure 10: Result of the merging after choosing (a).

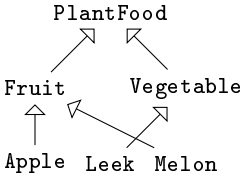


Figure 11: Result of the merging after choosing (b).

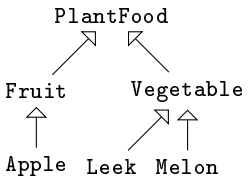


Figure 12: Result of the merging after choosing (c).

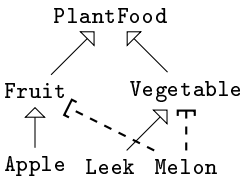


Figure 13: Result of the merging after choosing (d).

3.6 Properties

First, Δ can be confronted to the postulates of [6]. These postulates are used for characterizing a merging operator in propositional logic, but can be reused in the $\mathcal{L}_{\mathcal{T}}$ formalism. These postulates deal with the merging of multisets of knowledge bases, but, since for the operator Δ, the number of occurrences has no importance, we will consider only sets of knowledge bases.

These postulates are (for E, E_1, E_2: sets of knowledge bases; ψ_1, ψ_2: knowledge bases):

(A1) $\Delta(E)$ is consistent.
(A2) If $\bigwedge E$ is consistent then $\Delta(E)$ is equivalent to $\bigwedge E$.
(A3) If there is a bijection F from E_1 to E_2 such that $F(\psi)$ is equivalent with ψ, then $\Delta(E_1)$ is equivalent to $\Delta(E_2)$ (this postulates states that the syntax is irrelevant for Δ).
(A4) If $\psi_1 \wedge \psi_2$ is not consistent, then $\Delta(\{\psi_1, \psi_2\}) \not\models \psi_1$.
(A5) $\Delta(E_1) \wedge \Delta(E_2) \models \Delta(E_1 \cup E_2)$.
(A6) If $\Delta(E_1) \wedge \Delta(E_2)$ is consistent, then $\Delta(E_1 \cup E_2) \models \Delta(E_1) \wedge \Delta(E_2)$.

Δ satisfies (A1). Indeed, $\Delta(\{\psi_1, \psi_2, \ldots, \psi_n\}) \in MCS_\alpha(\delta)$ and thus is consistent.

Δ satisfies (A2). To prove it, let us assume that $\bigwedge E$ is consistent. $\bigwedge E = \bigwedge_i \widehat{\psi_i} = \alpha \cup \delta$. Thus $\alpha \cup \delta$ is consistent and so $MCS_\alpha(\delta) = \{\alpha \cup \delta\}$. Hence $\Delta(E) = \alpha \cup \delta = \bigwedge E$. Therefore, if $\bigwedge E$ is consistent then $\Delta(E) = \bigwedge E$ which proves (A2).

Δ satisfies (A3), which states the irrelevance of syntax. Indeed, for any knowledge bases ψ_1 and ψ_2 of $\mathcal{L}_\mathcal{T}^\neg$, ψ_1 is equivalent to ψ_2 iff $\widehat{\psi_1} = \widehat{\psi_2}$. Since Δ is defined thanks to the $\widehat{\psi_i}$'s, $\Delta(E)$ does not change when substituting a ψ_i by an equivalent knowledge base.

(A4) is not satisfied by Δ as the following counterexample shows. Let $\psi_1 = \{A \sqsubseteq B\}$ and $\psi_2 = \{A \not\sqsubseteq B\}$. Then $\widehat{\psi_1} = \{A \sqsubseteq B, B \not\sqsubseteq A\}$ and $\widehat{\psi_2} = \{A \not\sqsubseteq B, B \not\sqsubseteq A\}$. $\psi_1 \wedge \psi_2 = \{A \sqsubseteq B, A \not\sqsubseteq B, B \not\sqsubseteq A\}$, $\alpha = \{B \not\sqsubseteq A\}$, $\delta_1 = \{A \sqsubseteq B\}$, $\delta_2 = \{A \not\sqsubseteq B\}$, $\delta = \{A \sqsubseteq B, A \not\sqsubseteq B\}$, $MCS_\alpha(\delta) = \{\{A \sqsubseteq B, B \not\sqsubseteq A\}, \{A \not\sqsubseteq B, B \not\sqsubseteq A\}\}$.

Thus according to the choice performed by $\leq$, $\Delta(\{\psi_1, \psi_2\}) \models \psi_1$ or $\Delta(\{\psi_1, \psi_2\}) \models \psi_2$. (A4) is called in [6] the fairness property: it states that Δ should not make a preference between the knowledge bases to be merged. Our interpretation of the non fairness of our operator is that the $\mathcal{L}_\mathcal{T}^\neg$ language does not permit to express disjunctions and so, the operator has to make a choice (that is why $\leq$ has to be a total order). Indeed, let us consider $\mathcal{L}_\mathcal{T}^{\neg\vee}$ the extension of $\mathcal{L}_\mathcal{T}^\neg$ with disjunction: if ψ_1 and ψ_2 are $\mathcal{L}_\mathcal{T}^\neg$ knowledge bases, then $\psi_1 \vee \psi_2$ is an $\mathcal{L}_\mathcal{T}^{\neg\vee}$ knowledge base and ω satisfies it if ω satisfies ψ_1 or ω satisfies ψ_2. Now, let ∇ be the merging operator defined by $\nabla(E) = \bigvee MCS_\alpha(\delta)$(E: a set of $\mathcal{L}_\mathcal{T}^\neg$ knowledge bases, $\nabla(E)$: an $\mathcal{L}_\mathcal{T}^{\neg\vee}$ knowledge base). ∇ satisfies (A1), (A2), and (A3) (similar proofs than the proofs for Δ) and it satisfies also (A4): Let ψ_1, ψ_2 be two consistent $\mathcal{L}_\mathcal{T}^\neg$ knowledge bases such that $\psi_1 \wedge \psi_2$ is consistent. Thus, $\alpha = \widehat{\psi_1} \cap \widehat{\psi_2}$, $\beta_1 = \widehat{\psi_1} \setminus \alpha$, $\beta_2 = \widehat{\psi_2} \setminus \alpha$. $\alpha \cup \beta_1 = \widehat{\psi_1}$ and $\alpha \cup \beta_2 = \widehat{\psi_2}$ are consistent, so there exist ϕ_1 and ϕ_2 such that $\phi_i \in MCS_\alpha(\psi_1 \wedge \psi_2)$, $\widehat{\psi_i} \subseteq \phi_i (i \in \{1,2\})$, and $\phi_1 \cup \phi_2$ is inconsistent (since $\phi_1 \cup \phi_2 \equiv \widehat{\psi_1} \cup \widehat{\psi_2} = \psi_1 \wedge \psi_2$ that is inconsistent). Therefore $\phi_1 \wedge \phi_2 \models \nabla(\{\psi_1, \psi_2\})$, $\phi_1 \wedge \phi_2 \not\models \phi_1$ (since $\phi_1 \not\models \phi_2$), $\phi_1 \wedge \phi_2 \not\models \phi_2$ (since $\phi_2 \not\models \phi_1$). Hence, $\nabla(\{\psi_1, \psi_2\}) \not\models \phi_i$ for $i \in \{1,2\}$. This is why the non fairness of Δ is interpreted as a consequence of the necessity to make choices, in the $\mathcal{L}_\mathcal{T}^\neg$ formalism.

At this point, we have neither proven that Δ satisfies (A5) and/or (A6), nor found any counterexample.

A detailed complexity analysis has still to be carried out. However, a naive algorithm for Δ gives a polynomial complexity for the computation α and δ and an exponential complexity for the computation of $MCS_\alpha(\delta)$ (exponential in the size of δ). Therefore, with this algorithm, the computation of Δ is tractable when the taxonomies are similar. Indeed $\delta = \bigcup_i \widehat{\psi_i} - \bigcap_i \widehat{\psi_i}$ contains the formulas that are not shared by the taxonomies, so $|\delta|$ can be used to characterize the dissimilarities of the ψ_i's. Hence making frequents merging of taxonomies that have forked from a same taxonomy is useful. [14]

[14] This can be likened to the usefulness of frequent commits in a version management system like subversion, as noticed by Fabien Gandon. Thanks for this relevant remark, Fabien !

4 CONCLUSION AND FUTURE WORK

This paper has presented an operator for merging similar taxonomies that satisfies a subset of the postulates defined in [6]. There is still work to do in order to study its properties.

This operator is used to design an efficient algorithm for the merging when the taxonomies are similar, which is the case when they are originated from the same taxonomy and have not diverged for a too long time. This algorithm, in order to be efficient, should not compute $\widehat{\psi}$ (this operation is too complex and is too time and space consuming: $|\widehat{\psi}| = |\mathcal{V}(\psi)|^2 - |\mathcal{V}(\psi)|$).

The design of such an algorithm involves that the relation $\leq$ has to be specified. Indeed, the operator presented in this paper is based on the maximal consistent subsets of formulas issued from the conjunction of the knowledge bases to be merged.

A way to integrate this operator in the KCIP is to specify the $\leq$ relation as following:

- In the current KCIP, any user can submit his/her own version of the knowledge base at any time. When a user submits his/her version, it is merged with another user version and the knowledge base obtained by this merging process has to pass some automatic test in order to determine if it worth to be submitted to the community of the experts.
- Now, when a user wants to merge his/her own version to the current knowledge base, once the operator has determined all the MCS, they can be used to make all the possibilities of result for the merging and these possibilities can be submitted to the tests currently in use. Then all the possibilities which have passed the test are presented to the user, which will choose which possibility is the closest of what he wants (the user will make the choice represented by $\leq$ in our formalism). The choice done by the users can be stored for further reuse; this idea remains to be studied in details.

So, once this algorithm is efficiently implemented, it will be useful to the Kolflow project. But Kolflow does not limit itself to $\mathcal{L}_\mathcal{T}$ and there is a large spectrum of languages ranging from $\mathcal{L}_\mathcal{T}$ to, e.g., OWL DL. One advantage of $\mathcal{L}_\mathcal{T}$ is that its inferences are much less complex than OWL DL's (e.g., the sumbsumption test is linear for $\mathcal{L}_\mathcal{T}$ whereas it is NExpTime-hard in OWL DL). The question we intend to address in future work is what are the extensions of $\mathcal{L}_\mathcal{T}$ for which we will design a merging operator. Since $\mathcal{L}_\mathcal{T}$ can be considered as the fragment of RDFS with only one possible properties, `subClassOf` (corresponding to $\sqsubseteq$), some larger fragments should be considered (using other properties). Indeed in the particular case of WikiTaaable, some properties are more used or important and some are easier to compute than other ones so one can think of a kind of anytime approach where the algorithm will consecutively consider the RDFS properties starting with `subClassOf`.

A kind of equivalent to the MCS is the $MUPS$ that are used in the system Pellet:[15] this system contains a tool for debugging inconsistent ontologies which allows to find the $MUPS$ [4] of an inconsistent ontology. A MUPS (Minimal Unsatisfiability Preserving Sub-TBoxes) is a minimal subset of axioms which causes the inconsistency. If we find all the

[15] `http://clarkparsia.com/pellet/`

$MUPS$ of a knowledge base issued from the conjunction of two other ones, the set of all the possible consistent knowledge bases made from the conjunction of all the $MUPS$ after deleting one formula on each of them, is equivalent to the MCS. As Pellet works on knowledge bases on OWL DL it could be a lead to pass from $\mathcal{L}_{\mathcal{T}}^{\sqsupseteq}$ to OWL DL. It could also allows to compare our algorithm to the results of Pellet's debugging tool.

Finally, another future work (following a discussion with Pierre-Antoine Champin) is to study a similar merging operator based on another closed world assumption, a "disjointness assumption". This assumption for a tree-structured taxonomy ψ means that if neither $\psi \models A \sqsubseteq B$ nor $\psi \models B \sqsubseteq A$ then A and B are disjoint ($\omega(A) \cap \omega(B) = \emptyset$). The definition for any taxonomy must be adapted (e.g., in figure 2, `Fruit` et `Vegetable` are not comparable by $\sqsubseteq$, yet they should not be disjoint in order not to entail that there is no melon). The future work aim will be to see how this different closed world assumption modifies the belief merging operator.

ACKNOWLEDGEMENTS

The authors wish to thank the reviewers of this preliminary work for their helpful remarks useful for improving the quality of this paper and for our future work.

References

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson, 'On the Logic of Theory Change: partial meet functions for contraction and revision', *Journal of Symbolic Logic*, **50**, 510–530, (1985).

[2] *The Description Logic Handbook: Theory, Implementation and Applications*, eds., Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, Cambridge University Press, Cambridge, 2003.

[3] M. d'Aquin, 'Formally measuring agreement and disagreement in ontologies', in *Proceedings of the fifth international conference on Knowledge capture*, K-CAP '09, pp. 145–152, New York, NY, USA, (2009). ACM.

[4] B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur, 'Debugging OWL ontologies', in *Proceedings of the 14th international conference on World Wide Web (WWW-05)*, pp. 633–640, (2005).

[5] H. Katsuno and A. Mendelzon, 'Propositional knowledge base revision and minimal change', *Artificial Intelligence*, **52**(3), 263–294, (1991).

[6] S. Konieczny and R. Pino Pérez, 'Merging information under constraints: a logical framework', *Journal of Logic and Computation*, **12**(5), 773–808, (2002).

[7] P. Peppas, 'Belief Revision', in *Handbook of Knowledge Representation*, eds., F. van Harmelen, V. Lifschitz, and B. Porter, chapter 8, 317–359, Elsevier, (2008).

[8] H. Skaf-Molli, E. Desmontils, E. Nauer, G. Canals, A. Cordier, M. Lefevre, P. Molli, and Y. Toussaint, 'Knowledge continuous integration process (K-CIP)', in *Proceedings of the 21st international conference companion on World Wide Web*, pp. 1075–1082, (2012).

Ontology Merging and Conflict Resolution: Inconsistency and Incoherence Solving Approaches [1]

Raphael Cobe and **Renata Wassermann**[2]

Abstract. In recent years, researchers have focused on merging knowledge bases but a recurring problem after that task is the existence of Incoherences and Inconsistencies. In this paper, we enumerate a few attempts to deal with inconsistencies/incoherences while merging knowledge bases. We present a process that joins these conflict solving methods together and also an usage example that illustrates how the process can be used to solve these kinds of ontology modeling problems.

1 Introduction

There has been a rapid increase in availability of (semantic) information on the web. Nevertheless, there is no standard way of reusing knowledge, creating a challenge of building new knowledge bases for specific domains. This has forced users to build knowledge bases from scratch instead of being able to reuse previously established knowledge.

Ontologies have been considered as a mean for expressing and sharing semantic knowledge among systems [6] specially in the context of the Semantic Web. Their underlying structures allow machine-processing, providing a common vocabulary for expressing metadata about each web resource. Also, they are based on first order logic, allowing the usage of reasoners that are able to infer relationships between concepts based on their logical description. An ontology is naturally divided into terminological and assertional axioms, the TBox and ABox respectively. The first one defines a set of axioms that describe a set of properties of the concepts. The second defines facts about the individuals of the domain. In that sense, W3C proposed the OWL[3] standard specification language to express ontologies.

The integration of multiple knowledge sources may result in conflicting knowledge being joined together at a single base. This kind of problem may compromise the integrity and reliability of the knowledge base. For this matter, it is important that we distinguish Inconsistency and Incoherence. An ontology is considered to be inconsistent if and only if there is no interpretation that could satisfy all the axioms in the base [8]. This kind of problem typically happens with the assertional knowledge, i.e., the ABox. A knowledge base is considered to be incoherent if and only if there is a concept C such that for all possible models for the knowledge base, C has an empty interpretation [24]. This kind of problem happens with the terminological knowledge, i.e., the TBox.

Several approaches have been proposed to work with inconsistency solving like [23, 24, 17]. On the other hand a whole lot of different works are trying to deal with the incoherence problem like [19, 20, 7]. All these approaches deal with conflict solving in their own way. In this paper we propose a process that aims to group and integrate these initiatives. We divided this process in phases, that group activities proposed in the literature that relate to each other in the way that they contribute to solve the conflict, e.g., the Stratification phase groups activities that proposes to order the axioms by an specific criteria. In addition to the activities proposed in the literature, we propose a new one for numbered restriction axiom weakening in $\mathcal{ALCN}$. We also present an example that aims to show how this process can be used to solve conflicts and how each phase contributes to the final result.

In this paper we have assumed the following syntax convention: we used upper-case letters O and K to represent (sub-) ontologies, the ϕ greek letter to represent axioms and the a down-case letter to represent individuals, the upper-case letters C, D, E, F, G and H to represent concepts and the upper-case letter P to represent properties.

This paper is organized as follows. Section 2 presents a brief list of initiatives that aim to solve conflicts in ontologies. Section 3 presents the proposed process. Section 3.5 presents an usage example for the proposed process and finally, Section 4 presents our final remarks and outlines what we are currently working with.

2 Inconsistency and Incoherence Solving

In this section, we intend to show the common approaches used to deal with the ontology conflict problem.

Most of the works that propose to deal with incoherences aim in finding minimal subsets of the ontology that contains the incoherence core. This kind of approach was built on researches in the field of Diagnosis in AI. A parallel of the problems in diagnosis and knowledge base conflict solving was presented in [31], where the author used concepts and algorithms from the diagnosis researches to deal with conflicting propositional belief bases. She used a expand-shrink strategy that iteratively adds axioms to a set until the conflict rises and after that the set is shrink until the minimality is reached. She also proposed the use of Reiter's hitting sets algorithm [25] to build all possible minimal inconsistent sub-bases. Later, in [13], Kalyanpur adapted such strategy to fix ontology conflicts, he called such approach **Black-Box** because it did not depend on the reasoner internal structures, thus, the reasoner is used like a black-box. This expand-shrink approach was also adopted by Haase et al. in [8].

In [19], Meyer et al. proposed an algorithm for finding, instead of minimally inconsistent/incoherent subsets, maximally consistent sets from incoherent knowledge bases. The algorithm proposed builds incoherence-free ontologies. Unfortunately this results are not very helpful if the ontology designer is looking for the modeling error

[1] This research is sponsored by the FAPESP projects 2010/19111-9 and 2008/10498-8

[2] University of São Paulo, Brazil, email: {rmcobe, renata}@ime.usp.br

[3] www.w3.org/TR/owl-features/

that caused the incoherence. This algorithm is a modification of the *conjunctive maxi-adjustment* algorithm for propositional knowledge integration and is called *CMA-DL*. A different algorithm presented in [20] uses a tableaux-based strategy to build maximally coherent sub-ontologies. Kalyanpur [13] classified such approaches that rely on the reasoner structure as **Glass-Box** approaches.

Most of the inconsistency solving approaches have been inspired by model-based propositional logic inconsistency solving like what is presented at [15], where the authors propose that for finding a solution for an inconsistency in propositional logic belief base, they had to find models that differ minimally to the models of the formulas in the inconsistency. They developed a whole framework for describing how to find such minimally distant models. Gorogiannis and Hunter, in [5] propose an approach to deal with inconsistencies by means of *Dilation Operators* that are, basically, a strategy to iteratively relax the formulas to remove inconsistencies. The idea of using this operators was to be able to reuse the framework for inconsistency solving presented in [15]. The idea of formula weakening for inconsistency solving was also used by Qi et al. in [23], where they proposed a model-based operator named *weakening* that iteratively adds exceptions to the subsumptions axioms. In the next section we will show the process that we proposed which's goal is to integrate these two fronts.

3 Process for Integrating Inconsistency and Incoherence Solving Approaches

During the previous section we showed a few approaches described in the literature for conflict solving - in the propositional and the DL case - and ontology debugging. In the literature, many authors dealt with different aspects of inconsistency solving, like minimization of changing [20, 8, 7, 29, 16], axiom ranking [22, 27, 13] and formula weakening [19, 23] both for TBoxes and ABoxes. Each of these aspects are important for the conflict solving goal. Also, each of these are applicable at distinct situations as we showed in Section 2. We propose a mean to group these techniques in a conflict solving process that is applicable to inconsistent and incoherent ontologies.

The designed process allows the user to create his/her own inconsistency solving method, by selecting the activities that better suits his/her needs, according to the nature of the conflicts found in the ontology. In addition to the techniques proposed in the literature, we designed new techniques for axiom ranking - using Information Retrieval structures - and also axiom weakening. The process can be seen in Figure 1. Its activities have been grouped in 4 phases: (1) *Kernel Building*: in this phase we aim to build minimally conflict keeping sub-ontologies, i.e., S is a kernel of the inconsistent/incoherent ontology O iff, S is a subset of O, S is inconsistent/incoherent and there is no proper subset of S that is inconsistent/incoherent. We used the same designation as [13, 30]. The concept of kernel is similar to the *Minimal Incoherence Preserving Sub-Ontologies* (MIPS) and *Minimally Unsatisfiability Preserving Sub-TBoxes* (MUPS) [27, 8]; (2) *Stratification*: during this phase, the axioms in the chosen kernel are ordered according to some principle - the number of axioms that share concepts and individuals, for instance. We chose to use the same denomination presented in [24, 19]; (3) *Axiom Weakening*: the activities in this phase try to solve the inconsistencies (not incoherences) by modifying the axioms, weakening their restriction power - adding exception to subsumption axioms or iteratively increasing the n in axioms of the form $C \sqsubseteq \leq nP$, where C is a concept, P a property and n a natural number; (4) *Axiom Removal*: this phase aims to remove the axiom with the lowest priority (or trustability) in the kernel in which the user is working on solve the conflict. This approach is used in most of the works on ontology debugging [29, 27] and description logics belief revision [26].

In the following Sections we will present each phase and its composing activities

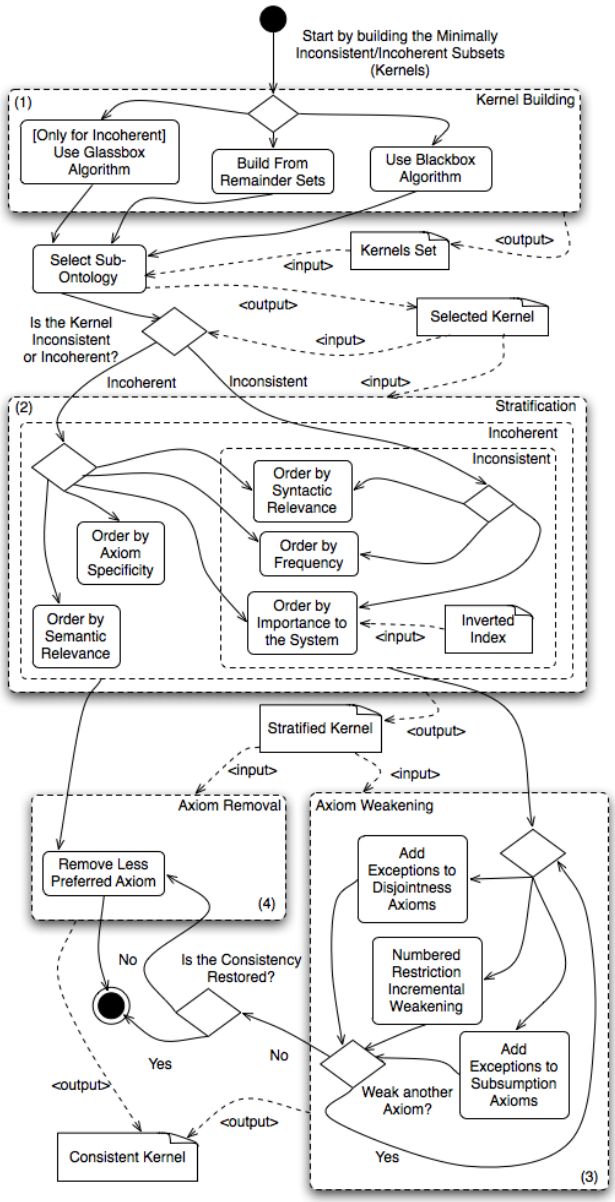


Figure 1: Process for Integrating Assertional and Terminological Approaches

3.1 Kernel Building Phase

The identification of minimal conflicting subsets is fundamental to the process, since it is during this phase that the conflict extension is defined and all axioms directly contributing to such an inconsistency are listed. This technique is commonly used in most of the conflict solving methods for both ABoxes and TBoxes. Most of the time, these sets are called MIPS and MUPS. The first one builds subsets that relates to the global coherence, i.e., the minimal subsets that causes the incoherence of the ontology, the second one relates to a specific concept, i.e., the minimal subset of axioms that when put together causes a given concept C to be necessarily empty.

The idea of minimally inconsistent/incoherent sub-sets relates di-

rectly to the concept of kernel [11] and partial meet [1] contractions found on the belief revision field. The first one uses *kernel* sets, i.e., minimal subsets that implies a given formula ϕ. The second one uses *remainder sets*, i.e., maximal subsets that do not implies a given formula ϕ. Falappa et al., in [4], presented a study on how these two subsets relate to each other and how to build one from another. In ontology debugging, the challenge is the same. The only difference is that the condition that we use to build our subsets is the consistency or coherence, much like in the Revision operation [1] for belief bases.

So, basically, we can separate the approaches to build the kernel sets in two main groups: the first one builds the kernel directly and the second one builds it from the remainder sets as presented in [4]. The glass-box and black-box approaches are in the first group. The first is reasoner independent and the second depends on modifications to the reasoner, using mostly, specific tableau rules to expand and close the branches on the tableau. We allowed the user to select one of this two approaches to build his/her kernel sets. This choice is important due to the nature of each approach. Also, usually, only TBox knowledge is considered like the presented in [29], [28] and [20]. In [10], Halaschek-Wiener et al. described a way to extend their algorithm for axiom pinpointing to cope with ABox update and consistency maintenance.

As the reasoning mechanisms are becoming more efficient, we notice a larger interest on the black-box algorithm. Such approaches are becoming more popular for their simplicity of implementation like the approach presented in [8]. The authors proposed a modified version of the strategy described in [31]. They begin the algorithm with an unsatisfiable concept and use the idea of structural connectedness to choose the next axioms to be inserted into the subset. They use this idea during the expansion phase. By the end of such phase the authors would get a unsatisfiability preserving sub-ontology that is not guaranteed to be minimal [8]. After that they use a linear shrink routine, removing iteratively the axioms in the subset to guarantee that the subset is minimal.

In addition to the direct kernel building we chose to allow the user use a strategy based on maximal consistent/coherent subsets, since there are a few algorithms in the literature that aims to build maximally consistent/coherent ontology. For instance, the work from Meyer et al. [19] builds such subsets using an iterative algorithm named Conjunctive Maxi-Adjustment, that at each iteration tries to include into the maximal subset, a subset of the ontology with size n (the value of n changes at each iteration).

Once the user built the remainder set, he/she can use the Reiter's hitting set algorithm [25] to build the kernel sets. Such algorithm have been presented in [31] and used in [13]. For more details on how to build the kernel from the remainder set, please refer to [10].

One could wonder why to build kernel from remainder sets and not present this sets to the user as possible consistent sub-ontologies. The problem with this approach is that the algorithm does not show the user the axioms that have been removed and if we have a large number of removed axioms, the user will have problems to guess what these axioms were from comparing the ontology before and after the algorithm. Kernels are also smaller than remainder sets, easier to manage and also they help the user to localize the conflicts source.

3.2 Stratification Phase

The next phase to be carried out during the process is the Stratification. The adoption of this group of activities was inspired by the works [23, 24, 19, 20]. These works have something in common: the knowledge bases go through a process of axiom ordering/ranking according to their importance or trustability to the system and for each pair of axioms either one of them is more important than the other or the other way around [3].

The stratification phase can be carried out manually, by domain experts like what is described by Haase et al. in [9], or by some automatic means. In our process we indicated a few techniques for automatic stratification.

Specific Axiom Prioritization has been proposed by Qi et al. in [24] and its main idea, taken from [2], aims to preserve the axioms that describe more general concepts, or more formally: an axiom $\phi_1 = C_1 \sqsubseteq D_1$ is more specific than the axiom $\phi_2 = C_2 \sqsubseteq D_2$ if and only if $C_1 \sqsubseteq C_2$ and $C_2 \not\sqsubseteq C_1$. In [2] the authors used such strategy to solve inconsistencies in security policy definitions. In the activity called "Order By Axiom Specificity" we allow the user to use the strategy defined by Qi et al. in [24], but also we allow the use of the reverse strategy, where the users may prioritize the more general axioms. The prioritization of more general axioms helps keeping the restriction defined in axioms of higher level as we can see in Example 1. This technique is only available for incoherent ontologies.

Example 1: Suppose that we have the following ontology: $O = \{C_2 \sqsubseteq C_1,\ C_3 \sqsubseteq C_1,\ C_1 \sqsubseteq \neg C_4,\ C_2 \sqsubseteq C_4\}$. It is easy to see that such ontology is incoherent and if we use the approach of Qi et al. [24], the result would be the stratified ontology: $B = \{(C_2 \sqsubseteq C_1, C_3 \sqsubseteq C_1, C_2 \sqsubseteq C_4), (C_1 \sqsubseteq \neg C_4)\}$, where the axiom $\phi_3 = C_1 \sqsubseteq \neg C_4$ has a lower priority, which means that it is a strong candidate for removal.

The removal of more general axioms can cause the removal of restrictions applied to concepts not involved in the inconsistency. The removal of the axiom $\phi_3 = C_1 \sqsubseteq \neg C_4$ in example 1 would impact on the axiom $\phi_2 = C_3 \sqsubseteq C_1$, once ϕ_3 says that the concepts C_4 and C_1 are disjoint and so are their descendants. One can imagine that the concept C_3 could have a large number of descendants on what the disjointness restriction should apply. The Example 2 shows the alternative result if we choose to prioritize the more general axioms.

Example 2: If we choose to invert the order of the elements after using the algorithm from Qi et al. [24], the result of the stratification would be the ontology $B' = \{(C_1 \sqsubseteq \neg C_4), (C_2 \sqsubseteq C_1, C_3 \sqsubseteq C_1, C_2 \sqsubseteq C_4)\}$. An algorithm that removes the higher order strata would give the user the choice of removing $C_2 \sqsubseteq C_1$, $C_3 \sqsubseteq C_1$ or $C_2 \sqsubseteq C_4$.

During the stratification phase the user can also choose to use the algorithms defined by Kalyanpur in [13], for axiom ranking. The available algorithms are:

- Order by frequency: the number of kernels in which the axiom appears. If the axiom appears in n kernels, if it is removed, then we are able to solve n inconsistencies. This technique can be used in both incoherent and inconsistent ontologies;
- Order by semantic relevance: the number or entailments that are lost if the axiom is removed, i.e., the number of inferred subclass relationships that are added or removed if the axiom is removed from the ontology. The larger the number of entailments the more preferred the axiom. This technique can only be used in incoherent ontologies; and
- Order by syntactic relevance: the number of axioms that share the concepts with the axiom being ranked. This strategy allows the user to choose to preserve axioms that share concepts with the higher number of other axioms, once this might indicate that they help describe important domain concepts, i.e., concepts that have a richer description - a higher number axioms to describe it and their relationships with other concepts. In order to use this technique with inconsistent ontologies, we also have to count the axioms that share individuals with the axiom being ranked.

Ontologies are commonly build to model a specific domain, and most of the cases, the knowledge about this domain is described in textual documents. In this paper we propose the use of *Information Retrieval* - IR techniques to order the axioms by their importance. The goal of this activity is to try to establish the importance of the axiom to a given domain ontology.

The IR field goal, according to Manning et al. [18], is to search within unstructured document collections to answer the user's queries. To do so, the systems usually classify and organize their document collection in specific ways to make easier the process of document retrieval, thus improving the time of retrieval and the quality (relevance) of the retrieved documents.

The most trivial form of IR is to examine all the available documents in a linear way and check whether these documents have the terms of the user's query or not. Unfortunately this retrieval form has a high cost and takes too long if we are considering large document collections. Thus, some form of index is needed.

One of the most common index structures is called inverted index. Such structure relates elements and their location, in the IR context, the elements are the terms and the location is the documents in what such terms can be found. It is easy to see that structure helps to speed up the retrieval process. Now the IR system has only to look inside the index to discover which documents contain a given term. This kind of structure allows us to identify which terms are more important to the system, i.e., are present in the larger number of documents and this kind of information can be used to define which concepts in an inconsistent/incoherent ontology are more important. Due to the lack of space we are only going to present the outline of the algorithm.

The strategy here takes into account the idea of semantic distance between two concepts. Such measure is responsible for defining how (semantically) similar two concepts are. From this measure we can establish how similar are the classes in the ontology axiom and the terms in the inverted index. From these data we can order the axioms according to the number of occurrences of their most similar terms in the index. In our process the user is allowed to choose which semantic distance he/she wants to use once there is a large number of proposals on this matter, like [14, 12] that use the Wikipedia[4] and the Wordnet[5] to establish the distance between two terms.

The algorithm then calculates a rank for each axiom ϕ_{rank} using the following formula $\phi_{rank} = \sum rank(X)$, where $X \in Concepts(\phi) \cup Individuals(\phi)$; and $rank(X) = \sum occurrences(word) * sim(X, w)$, where $w \in index$ and $sim(X, w) \leq \gamma$. γ is a value defined by the user that states the minimum level of similarity to be considered. The level os similarity is calculated by the $sim()$ function and $occurrences()$ function retrieves the number of documents in what the w word appears.

This technique can be used for stratifying both inconsistent and incoherent kernels.

3.3 Axiom Weakening Phase

This phase aims to solve inconsistencies only. It is composed by a series of axiom *weakening* activities that were designed to lose the formulae restriction power. The user can go through this phase several times combining the techniques of each activity to solve the inconsistency. This phase was inspired by the works from Qi et al. [23] and Meyer et al. [19] where the authors propose ways to iteratively weaken each axiom. The goal here is to avoid discarding whole axioms.

[4] http://wikipedia.org
[5] http://wordnet.princeton.edu/

The first activity available to the user is the one described in [23]. The idea consists in transforming inconsistent axioms of the form $C \sqsubseteq D$, adding explicit exceptions for individuals that do not satisfy such restriction, i.e., if a given individual declaration $C(a) \sqcap \neg D(a)$ breaks the axiom $\phi = C \sqsubseteq D$, we can add an explicit exception for the a individual weakening the axiom, obtaining the axiom $\phi' = (C \sqcap \neg\{a\}) \sqsubseteq D$, which states that all individuals of the C type, except a, also belong to the D type.

The algorithm used in this phase is the same: we make explicit that an individual is not affected by an axiom of the form $C \sqsubseteq D$, i.e. being an interpretation $\mathcal{I}$, an individuals a and two concepts C and D, we add the information that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ *but* $a^{\mathcal{I}} \notin D^{\mathcal{I}}$ by modifying the original axiom to $(C \sqcap \neg\{a\}) \sqsubseteq D$. Example 3 taken from [19] shows how this approach can be used to solve inconsistencies.

Example 3: Suppose that we have the inconsistent ontology $O = \{bird(tweety), \neg flies(tweety), bird(chirpy), bird \sqsubseteq flies\}$ that has the kernel $K_1 = \{bird(tweety), \neg flies(tweety), bird \sqsubseteq flies\}$. After using the exception adding approach we have the restored kernel $KR_1 = \{bird(tweety), \neg flies(tweety), (bird \sqcap \neg\{tweety\}) \sqsubseteq flies\}$.

Although this strategy can clearly solve the inconsistency, it cannot be used directly to solve inconsistencies in OWL ontologies because the axiom to which the exception was added loses too much power. For instance, if we take the consistent sub ontology $O' = \{\neg flies(tweety), bird(chirpy), bird \sqsubseteq flies\}$ obtained from O (from Example 3) by arbitrary removing an individual declaration contained in the kernel we could infer that the individual named $chirpy$ also could fly. It is natural that, after solving the inconsistency, the user performs an axiom merging operation by joining the axioms left out of the kernel with the restored consistent kernel into a new ontology, e.g, $O'' = \{bird(tweety), \neg flies(tweety), bird(chirpy), (bird \sqcap \neg\{tweety\}) \sqsubseteq flies\}$. It is easy to see that the entailment that $chirpy$ could also fly is missing. This happens due to the fact that OWL ontologies is built on the premiss that we deal with an *open world*, which means that two individuals can be considered the same individual unless we make explicit they are different. In the Example 3, we cannot say that $chirpy$ is not the same individual as $tweety$, so we can no longer infer that $chirpy$ can fly, unless we add the axiom $chirpy \neq tweety$.

So, in order to use this approach, the user that chooses to use it within OWL ontologies will have to take care of missing subclass entailments. One way to automate this task would be: compare the taxonomic classification of the ontology before and after the inconsistency solving and examine the cases where individual not included in any kernel have different before and after taxonomic classifications. In order to run a reasoner and build a taxonomic classification the ontology must not be inconsistent. One way to deal with this is to remove the individuals involved in any conflict, i.e. in any kernel, then do the taxonomic classification. In Example 3 we should remove the $tweety$ individual and later do the taxonomic classification that infers $flies(chirpy)$. With this information we are able to notice that at least one subclass entailment is lost, thus we need to add the $chirpy \neq tweety$ axiom.

Another extra concern with the usage of this fix is to deal with ontology evolution, more precisely with new individuals inclusion. The ontology designer should be careful while adding new individuals, checking their taxonomic classification for missing subclass entailments.

We also added in our process a modified version of this exception-adding technique from [23] compliant with the OWL Disjointness

construction $C\ DisjointWith\ D$ which is the same as $C \sqsubseteq \neg D$ which means that the interpretation function for the two concepts should have no elements in common, i.e., being $\mathcal{I}$ an interpretation function and C and D two concepts, to say that C is disjoint from D means that $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.

We propose the usage of the same axiom-adding technique with DisjointWith axioms. For instance, if we have a series of individuals $a_1, a_2, \ldots, a_n$ that breaks a disjointness axiom $C\ DisjointWith\ D$, after adding the exceptions, the axiom is changed to: $(C \sqcap \neg\{a_1\} \sqcap \neg\{a_2\} \sqcap \ldots \sqcap \neg\{a_n\})\ DisjointWith\ D$. Example 4 shows how this technique can be used.

Example 4: Consider the kernel $K = \{Mammal\ DisjointWith\ Bird,\ HasBeak \sqsubseteq Bird,\ Mammal(Platypus),\ HasBeak(Platypus)\}$. If we use the approach from [23], exceptions should be added to the axiom $\phi_1 = HasBeak \sqsubseteq Bird$ and the result of the algorithm would be $\phi_1' = (HasBeak \sqcap \neg\{Platypus\}) \sqsubseteq Bird$ which means that there is something that has a beak that is not a bird. Although, recent genetic analysis showed that the Platypus has genes from mammals, birds and reptile. The resulting axiom ϕ_1' is no longer correct from the design point of view. In this case, makes sense to add an exception for the axiom $\phi_2 = Mammal\ DisjointWith\ Bird$ obtaining the axiom $\phi_2' = (Mammal \sqcap \neg\{Platypus\}\ DisjointWith\ Bird$ which states that every mammal except the Platypus is not a bird.

The last available activity available for the users during the weakening phase was inspired by the work of Meyer et al. [19], where the authors propose a algorithm for inconsistency solving that relies on an OWL extension, named concept cardinality that allows the designer to explicit say that a given concept has at least (or at most) n individuals, n being a natural number. So, in the algorithm the authors iteratively changed the n value, trying to restore consistency. We adapted the idea to the axioms of the form $\leq n.P$ where P is a property and n a natural number.

In our proposed approach, like in [19], we iteratively increase or decrease the number n in the axiom, trying to reestablish the consistency. In the case of the $\leq n.P$ axiom, by incrementing the value of n we are allowing that more individuals be connected by means of the P property. The Example 5 shows how this approach works.

Example 5: Suppose we have the kernel $K = \{C \sqsubseteq\leq\ 1P, \{a_1\}\ C(a_1),\ a_3 \neq a_2,\ P(a_1, a_2),\ P(a_1, a_3)\}$. In order to solve this inconsistency, using the approach that we proposed, we can iteratively increment the value of the axiom $\phi = C \sqsubseteq\leq\ 1P$ allowing that more individuals relate to each other by means of the P property. In this example, we only have to increment the value of n in 1, i.e., the ontology becomes consistent if ϕ is changed into $C \sqsubseteq\leq 2\ P$.

The process of building kernels compromises with a minimality principle, which may be a problem for the cardinality weakening approach, once there may be more individuals that break a $\leq n.P$ axiom, i.e. if we have an inconsistent ontology O of the form presented on Definition 1.

Definition 1: A TBox $\leq nP$-based inconsistency is defined by the TBox axioms $C \sqsubseteq\leq\ nP$ and the ABox axioms $C(a_1), a_i \neq a_j, P(a_1, a_i)$

A variant of this type of ontology presented in Definition 1 would be if we have the $\leq nP$ axiom defining an individual, as an ABox only axiom, what we called a ABox $\leq nP$-based inconsistency, that is defined by the set of individuals $\{a_1, a_2, \ldots, a_n, a_n + k\}$ and the ABox axioms $(\leq nP)(a_1), a_i \neq a_j, P(a_1, a_i)$, Where $k > n$, $i \neq j$, with $j\ and\ i\ \in 1\ ..\ n + k$.

The number of kernels for both the TBox and ABox $\leq nP$ based inconsistent ontologies, would be the number of combinations of size $n+1$ of the a_i individuals that relate to the individuals of the C class subsumed by the $\leq nP$ axiom in the case of the TBox inconsistency, or the individual defined as belonging to the $\leq nP$ concept in the ABox inconsistency. Example 6 extends Example 5 and shows what happens if we create an ontology from the kernel K adding one extra individual, a_4 that is different from all other individuals that also relates to the individual a_1 by means of the property P.

Example 6: Suppose that we have the inconsistent ontology $O = \{C \sqsubseteq\leq\ 1P, C(a_1),\ a_3 \neq a_2,\ a_4 \neq a_2,\ a_4 \neq a_3,\ P(a_1, a_2),\ P(a_1, a_3),\ P(a_1, a_4)\}$. The number of kernels for the ontology is the number of combinations of size 2 of the 3 individuals that relate to a_1 by means of the P property, i.e., 3 kernels: $K_1 = \{C \sqsubseteq\leq\ 1P, C(a_1),\ a_3 \neq a_2,\ P(a_1, a_2),\ P(a_1, a_3)\}$, $K_2 = \{C \sqsubseteq\leq\ 1P,\ C(a_1),\ a_4 \neq a_2,\ P(a_1, a_2),\ P(a_1, a_4)\}$ and $K_3 = \{C \sqsubseteq\leq\ 1P,\ C(a_1),\ a_4 \neq a_3,\ P(a_1, a_3),\ P(a_1, a_4)\}$.

We have to observe that, in Example 5 the user has already gone through the first phase of the process, where, at the end, he/she chooses a kernel in which he/she is interested to restore consistency. What we have to notice is that consistency is only fixed locally, i.e., within the chosen kernel. So, after solving this inconsistency and merging the kernel with the previous ontology, the inconsistency may rise again and the user will have to go through a new process iteration. On this case, the kernel usage has proven not to be effective while diagnosing inconsistencies of this kind. A better approach would be to build larger sub-ontology, containing the same axioms of the $\leq nP$-based inconsistency presented in Definition 1 and do a syntactical check, counting the number of property declarations $P(a_1, a_i)$ where $k \leq i \leq j$ and $a_k \neq a_j$ for $k \neq j$ in the fragment and setting n to the corresponding number.

A similar problem, illustrated in Example 7 occurs with the exception adding approach from [23]. The example shows that kernel usage is also not effective while fixing inconsistencies using this technique.

Example 7: Take the ontology $O = \{C \sqsubseteq D,\ C(a_1),\ \neg D(a_1),\ C(a_2),\ \neg D(a_2),\ a_1 \neq a_2\}$ and its kernels $K_1 = \{C(a_1),\ \neg D(a_1),\ C \sqsubseteq D\}$ and $K_2 = \{C(a_2),\ \neg D(a_2),\ C \sqsubseteq D\}$. If the user chooses to weaken the first one by exception adding, the weakened axiom $\phi' = (C \sqcap \neg\{a_1\}) \sqsubseteq D$ when used to update the ontology results in $O' = \{(C \sqcap \neg\{a_1\}) \sqsubseteq D,\ C(a_1),\ \neg D(a_1),\ C(a_2),\ \neg D(a_2),\ a_1 \neq a_2\}$ that is still inconsistent.

A strategy to deal with this problem is to first stratify the kernels using the frequency criteria proposed by Kalyanpur [13]. The frequency criteria causes axioms that are in a large number of kernels to be less preferred, thus, good candidates for removal. By using this criteria we ensure that the axioms to which we will add exceptions come first in the stratified kernel. After stratifying all kernels we calculate a cutting set which is a set that contains a single element from each kernel, and for each axiom in the cutting set we examine each kernel it belongs to and try to solve it by adding exceptions to the axiom in the cutting set. After that, merge all exceptions into one and check if the consistency has been restored. Example 8 is an extension of Example 7 and shows how this approach works.

Example 8: First we need to stratify the kernels K_1 and K_2 using the frequency criteria that would give us the stratified kernels $K_1' = \{(C \sqsubseteq D),\ (\neg D(a_1),\ C(a_1))\}$ and $K_2' = \{(C \sqsubseteq D),\ (\neg D(a_2),\ C(a_2))\}$. Then we calculate a cutting set L by selecting the less preferred axioms in the kernels that would give us $L = \{C \sqsubseteq D\}$. After that, we have to track the kernels to which the $C \sqsubseteq D$ axiom belongs and check if there is any way to fix the

kernel inconsistency by adding exceptions and that is the case for both K_1 and K_2. In K_1, the axiom $C \sqsubseteq D$ would be replaced by $(C \sqcap \neg\{a_1\}) \sqsubseteq D$ as presented in Example 7. Conversely, in K_2 the $C \sqsubseteq D$ axiom would be replaced by $(C \sqcap \neg\{a_2\}) \sqsubseteq D$. After calculating the exceptions for the less preferred axiom, we join them together and obtain $(C \sqcap \neg\{a_1\} \sqcap \neg\{a_2\}) \sqsubseteq D$, that once used to update the O ontology would restore its consistency.

Inconsistencies involving $\geq nP$ axioms appear when there are explicit and opposite declarations to the $\geq nP$ axioms, e.g., if we have the following kernel $K = \{C \sqsubseteq \geq 2P, C(a_1), \leq 3P)(a_1)\}$. In this case, we have conflicting declarations, probably from poorly cared ontology modeling phase, that cannot be solved with the strategy to solve $\leq nP$ inconsistencies that we presented earlier. In order to solve this kind of inconsistency, we propose that we set the value of n from the $\geq nP$ to the same as the $\leq nP$ axiom, thus restoring the consistency.

By the end of this phase, the ontology should be consistent, but it may be the case that the user does not want to try to weaken axioms anymore, for instance, if he/she has to add a large number of exceptions to the subsumption or disjointness axiom. It may make more sense to remove such axioms, instead of add exceptions that may include all of the individuals of the domain. The next phase has the goal of removing the least important (or trustable) axiom.

3.4 Axiom Removal Phase

This phase takes as input the weakened or only stratified kernel chosen by the user and removes the axiom that is found on the lower order strata. It may contain more than one axiom and the user can manually choose which to remove. The usage of this approach makes sure that the axiom being removed is less important to the ontology, according to one of the criteria used during the stratification phase.

This phase can be executed right after the stratification, when it is the case that the ontology is only incoherent or when it is more important to keep the individuals than the TBox axioms. This approach is the same used in [9, 7, 20, 29].

It can also be executed after the user has tried unsuccessfully to weaken axioms. As we discussed on Section 3.3, it may be the case that the user has to add exceptions to all of the individuals in the ontology in order to restore its consistency, and even after that, the kernel still be incoherent. The Example 9 shows this case.

Example 9: If we have the following stratified kernel $K = \{E \sqsubseteq D$,$E \sqsubseteq C$,$C\ DisjointWith\ D$, $E(a_1)\}$. If we go through a process of axiom weakening by adding exceptions to the subsumption axiom $E \sqsubseteq D$ obtaining $(E \sqcap\ \neg\{a_1\}) \sqsubseteq D$ we were able to restore its consistency, but the ontology is still incoherent, since we are saying that two disjoint concepts have one common descendent.

A better way to solve all problems in the kernel in Example 9 would be the axiom removal, removing, for instance the axiom $E \sqsubseteq D$, which would restore the global consistency/coherence.

By the end of this phase the user obtains an inconsistent/incoherence free sub-ontology, that now can be used to update the original ontology. As the new sub-ontology may still conflict with other sub-ontologies, the process may be executed more than once, until the global consistency/coherence be restored.

3.5 Usage Example

In this section we will describe a brief case study that aims to show how the user should interact with the process and how it can be used to restore consistency/coherence. We have chosen to describe a small example due to lack of space.

Suppose that we have the following ontology O composed by the axioms:

TBox:

$\phi_1 = D \sqsubseteq G$, $\phi_2 = E \sqsubseteq F$,
$\phi_3 = C \sqsubseteq D$, $\phi_4 = C \sqsubseteq E$,
$\phi_5 = G\ DisjointWith\ F$,
$\phi_6 = F \sqsubseteq \leq 1P$, and
$\phi_7 = H \equiv \neg G$

ABox:

$\phi_8 = a_2 \neq a_3$, $\phi_9 = a_2 \neq a_4$
$\phi_{10} = a_3 \neq a_4$, $\phi_{11} = E(a_1)$
$\phi_{12} = P(a_1, a_2)$,
$\phi_{13} = P(a_1, a_3)$, and
$\phi_{14} = P(a_1, a_4)$

The ontology that we chose for our usage example is both inconsistent, due to the fact that the individual a_1 relates to more than 1 other individual by means of the P property; and also incoherent, due to the fact that two disjoint classes G and F have one common descendent, the class C. Now we will go through the conflict solving process that we proposed earlier and show how it behaves while solving this issues.

The first phase is the Kernel Building Phase, where, from the O ontology we derive the kernels: $K_1 = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$ (incoherent), $K_2 = \{\phi_2, \phi_4, \phi_6, \phi_8, \phi_{11}, \phi_{12}, \phi_{13}\}$ (inconsistent), $K_3 = \{\phi_2, \phi_4, \phi_6, \phi_9, \phi_{11}, \phi_{12}, \phi_{14}\}$ (inconsistent), and $K_4 = \{\phi_2, \phi_4, \phi_6, \phi_{10}, \phi_{11}, \phi_{13}, \phi_{14}\}$ (inconsistent). Once this ontology has incoherences and inconsistencies the user cannot use the Glass-Box algorithm described at [30, 13] to build kernels.

After building the kernels, the user has to choose one to work with. In this example, the user chooses the kernel K_2 to restore its consistency. Then he/she goes for the phase 2 of the process (Figure 1), the Stratification Phase and during this phase the user chooses the ordering by axiom frequency activity and the obtained Stratified kernel is $K'_2 = \{(\phi_2, \phi_4), (\phi_{11}, \phi_6), (\phi_{12}, \phi_{13}), (\phi_8)\}$. The obtained result has 4 strata. The first one contains the axioms that are present in all 4 kernels, the second the axiom that is present on 3 kernel, the third the axioms that are present in 2 kernels and the later, the axiom that is present only on this kernel.

After the Stratification Phase, the user chooses to try to solve the inconsistencies by means of axiom weakening. This kernel has a $\leq nP$-based inconsistency, then we apply the strategy that we defined on Section 3.3[6], so we use the other computed kernels with the same kind of inconsistency - K_3 and K_4 - and group them together with K_2. After that we merge them, thus building the larger sub-ontology $O_{\leq nP} = \{\phi_2, \phi_4, \phi_6, \phi_8, \phi_9, \phi_{10}, \phi_{11}, \phi_{12}, \phi_{13}, \phi_{14}\}$.

In the $\leq nP$-based conflict solving strategy, we first build the larger sub-ontology $O_{\leq nP}$ than we count the number of property declarations that connect a_1 to each different individuals $P(a_1, a_i)$. In our example, the property declarations are $P(a_1, a_2)$, $P(a_1, a_3)$ and $P(a_1, a_4)$, so the number of n is increased from 1 to 3, restoring the kernel consistency.

After solving the inconsistency, the user updates the ontology with the newly consistency restored axioms and checks the whole ontology consistency/coherence and verifies that the ontology is still incoherent, having the unique kernel $K_1 = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$. In this case, the user could have build the kernel using the Glass-Box algorithm presented on [13] on the Kernel Building Phase.

After rebuilding the kernels, the user now advances to a new Stratification Phase, but this time he/she can execute stratification activities built exclusively for incoherent kernels. We will assume that the user chooses to execute the Order by Axiom Specificity activity like what has been proposed in [24, 2]. Before we build the stratified ker-

[6] We could also have solved the inconsistency by executing the Add Exceptions to Subsumption Axioms activity, having as input the least preferred axioms ϕ_2 and ϕ_4. Each one of is considered a cutting set, the user should then pick one of them to weaken in all inconsistent (not incoherent) kernels. If he/she chooses ϕ_2, in all inconsistent kernels, if we update them with the axiom $\phi'_2 = (E \sqcap \neg\{a_1\}) \sqsubseteq F$ all three kernels have their consistency restored.

nel it is important that we rewrite the axiom $G\ DisjointWith\ F$ by means of subsumption axioms, i.e. $G\ DisjointWith\ F \leftrightarrow \phi'_5 = G \sqsubseteq\ \neg F$. The stratified kernel would be $K'_1 = \{(\phi'_5, \phi_2), (\phi_1), (\phi_3, \phi_4)\}$.

After the Stratification Phase, the Axiom Removal Phase takes place and the user should choose which axiom to remove from the lower order strata, i.e., either ϕ'_5 or ϕ_2. Suppose that the user chose ϕ'_5. After that the ontology coherence will be restored[7].

4 Final Remarks

In this work we presented a process that integrates a large number of common techniques used while dealing with conflict solving. The process joins together approaches that deal with modeling problems for both ABox and TBox. Our process can also be thought as a conceptual framework to classify conflict solving methods.

We also added a few techniques that we thought to make sense in the context of conflict solving. We developed a technique that uses information retrieval structures to stratify ontologies and also proposed a new method for numbered restriction axiom weakening.

Our work relates vaguely to the one developed by the PROMPT team presented in [21]. Both our work and theirs present a process that aims to fix ontology consistency/coherence. The difference is that they deal with the problem from only a pragmatic point of view and we have developed our process by grouping solid theoretical approaches. Their tool would fail to capture the simple conflict in the ontology $O = \{C \sqsubseteq \neg D, E \sqsubseteq C, E \sqsubseteq D\}$ for instance.

Currently we are developing a software framework that joins together all phases proposed by the process. We intend to build a Protègè plugin from such a framework and make it available as a free/opensource software. After the software built we intend to work on ways to evaluate the whole process effectiveness against *ad-hoc* conflict solving.

REFERENCES

[1] C.E. Alchourron, P. Gaerdenfors, and D. Makinson, 'On the Logic of Theory Change: Partial Meet Contraction and Revision Functions', *The Journal of Symbolic Logic*, **50**(2), 510–530, (1985).

[2] S. Benferhat, 'A stratification-based approach for handling conflicts in access control', in *In 8th ACM Symposium on Access Control Models and Technologies (SACMAT'03*, pp. 189–195. Press, (2003).

[3] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade, 'Inconsistency management and prioritized syntax-based entailment', in *Proceeding of the IJCAI'93*, pp. 640–640, (1993).

[4] M.A. Falappa, E.L. Fermé, and G. Kern-Isberner, 'On the logic of theory change: Relations between incision and selection functions', *Proceedings of the ECAI'06*, **141**, (2006).

[5] N. Gorogiannis and A. Hunter, 'Merging first-order knowledge using dilation operators', in *Proceedings of the FoIKS'08*, pp. 132–150, Berlin, Heidelberg, (2008). Springer-Verlag.

[6] T.R. Gruber, 'A translation approach to portable ontology specifications', *Knowledge acquisition*, **5**, 199–199, (1993).

[7] P. Haase and L. Stojanovic, 'Consistent evolution of owl ontologies', in *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, 91–133, Springer Berlin, (2005).

[8] P. Haase, F. van Harmelen, Zh. Huang, H. Stuckenschmidt, and Y. Sure, 'A framework for handling inconsistency in changing ontologies', in *Proceedings of the Fourth Internation Semantic Web Conference*, volume 3729 of *LNCS*, pp. 353–367. Springer, (2005).

[9] P. Haase and J. Volker, 'Ontology learning and reasoning — dealing with uncertainty and inconsistency', in *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, 366–384, Springer Berlin / Heidelberg, (2008).

[10] C. Halaschek-Wiener, A. Kalyanpur, and B. Parsia, 'Extending tableau tracing for abox updates', Technical report, UMIACS, (2006).

[11] S.O. Hansson, 'Kernel contraction', *Journal of Symbolic Logic*, **59**, 845–859, (1994).

[12] Y. Haralambous and V. Klyuev, 'A semantic relatedness measure based on combined encyclopedic, ontological and collocational knowledge', *Proceedings of the FedCSIS'11*, **abs/1107.4723**, (2011).

[13] A. Kalyanpur, *Debugging and Repair of Owl Ontologies*, Ph.D. dissertation, University of Maryland, College Park, USA, 2006.

[14] V. Klyuev and Y. Haralambous, 'Query expansion: Term selection using the semantic relatedness measure', *Proceedings of the FedCSIS'11*, 195–199, (2011).

[15] S. Konieczny and R. Pino Pérez, 'Merging with integrity constraints', in *Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'99)*, pp. 233–244, (1999).

[16] S. C. Lam, J.Z. Pan, D. Sleeman, and W. Vasconcelos, 'A fine-grained approach to resolving unsatisfiable ontologies', in *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pp. 428 –434, (dec. 2006).

[17] Y. Ma, G. Qi, P. Hitzler, and Z. Lin, 'Measuring inconsistency for description logics based on paraconsistent semantics', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ed., Khaled Mellouli, volume 4724 of *Lecture Notes in Computer Science*, 30–41, Springer Berlin / Heidelberg, (2007).

[18] C.D. Manning, P. Raghavan, and H. Schutze, 'An Introduction to Information Retrieval [Draft]', *Cambridge, UK: Cambridge University Press. Retrieved May*, **11**, 2009, (2008).

[19] T. Meyer, K. Lee, and R. Booth, 'Knowledge integration for description logics', in *Proceedings of the AAAI'05*, volume 20, pp. 645–650. AAAI Press, (2005).

[20] T. Meyer, K. Lee, R. Booth, and J.Z. Pan, 'Finding maximally satisfiable terminologies for the description logic alc', in *Proceedings of the AAAI'06*, volume 21, p. 269. AAAI Press, (2006).

[21] N.F. Noy and M.A. Musen, 'The prompt suite: interactive tools for ontology merging and mapping', *International Journal of Human-Computer Studies*, **59**(6), 983–1024, (2003).

[22] G. Qi and A. Hunter, 'Measuring Incoherence in Description Logic-Based Ontologies', *LECTURE NOTES IN COMPUTER SCIENCE*, **4825**, 381, (2007).

[23] G. Qi, W. Liu, and D. Bell, 'A revision-based approach to handling inconsistency in description logics', *Artif. Intell. Rev.*, **26**, 115–128, (October 2006).

[24] G. Qi and J.Z. Pan, 'A stratification-based approach for inconsistency handling in description logics', in *Proceedings of the IWOD'07*, p. 83, Innsbruck, Austria, (2007).

[25] R. Reiter, 'A theory of diagnosis from first principles', *Artificial intelligence*, **32**(1), 57–95, (1987).

[26] Márcio Moretto Ribeiro, *Belief Revision in Non-Classical Logics (To Appear)*, Springer, 2012.

[27] S. Schlobach, 'Debugging and semantic clarification by pinpointing', in *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science, 27–44, Springer Berlin / Heidelberg, (2005).

[28] S. Schlobach and R. Cornet, 'Non-standard reasoning services for the debugging of description logic terminologies', in *Proceeding of the IJCAI'03*, volume 18, pp. 355–362, (2003).

[29] S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen, 'Debugging incoherent terminologies', *Journal of Automated Reasoning*, **39**, 317–349, (2007). 10.1007/s10817-007-9076-z.

[30] R. Wassermann, *Resource-Bounded Belief Revision*, Ph.D. dissertation, Institute for Logic, Language and Computation — University of Amsterdam, 1999.

[31] Renata Wassermann, 'An algorithm for belief revision', *Proceedings of the KR'00*, (2000).

[7] An alternative would be to give more importance to the more general axiom as we discussed in Section 3.2. In our example, we removed the axiom ϕ'_5 which may affect the hierarchy of the subsumed concept F. For instance, the inference $F \sqsubseteq H$ could not be made anymore. Perhaps a better solution would be to discard one of the less general axioms ϕ_3 or ϕ_4, that would not affect a larger number of other axioms, so if the user chooses to stratify the kernel prioritizing less general axioms he/she will be able to keep more information about the whole domain.

Belief Management for HRI Planning

Julien Guitton and **Matthieu Warnier** and **Rachid Alami** [1]

Abstract. This paper presents an extension of a hierarchical planning approach designed to handle multi-agent problems and, more especially, Human-Robot Interaction problems in which a robot and a human have to collaborate in order to achieve a joint goal. Our method allows to reason and plan for agents that have different or incomplete beliefs in order to produce feasible and comprehensible plans. It is based on a new description of the agent's beliefs and a mechanism that produces and inserts some communication actions into the current plan.

1 Introduction

When acting in an environment with other partners, an agent has to reason not only on its own capabilities but also on the capabilities of the other agents to achieve a task in a collaborative way. In the context of Human-Robot Interaction (HRI), the robot needs to reason about the human's knowledge: the robot and the human may not have the same vision of the scene as well as the same information about the objects of the environment. This reasoning is complicated by the fact that the robot may not have the cognitive model of the human. Indeed, except through some dialog phases, the robot can only infer the knowledge of its human partner concerning the environment through a reasoning using a perspective-taking ability.

With this knowledge, which can be different from its own knowledge or partially incomplete, the robot has to produce a plan for him and for its partner in order to achieve a joint goal. This plan should be precise and comprehensible for the human.

In a previous work, we have presented a dedicated planner called HATP [2], for Human Aware Task Planner, which is based on hierarchical task planning combined with a set of behavior rules that leads the robot decisions and allows to produce plans that are socially acceptable for humans. In this paper, we extend this planner to deal with HRI problems in which the robot and the human may not have the same beliefs on the environment or incomplete beliefs. We call this extension *Belief Management*.

In the next section, we make an overview of existing work on the consideration of the human when designing a robotic architecture and existing work on planning for collaborative task achievement between a robot and a human. Then, in section 3, we present the HATP planner which provides the basis for this work. In section 4, we propose an extension of the HATP formalism to take into account beliefs of the different partners and the algorithm part allowing to handle this extension. In section 5, we present the integration of this work in our robotic platform as well as the different modules allowing to gather and manage agent's knowledge. Finally, in section 6 and 7, we illustrate the planning process with Belief Management through some basic examples and a scenario in real situation where a human and a robot have to cooperate in order to clean a table, *i.e.,* to put some tapes into a trash bin.

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France; Univ. de Toulouse, LAAS, F-31400 Toulouse, France; email: firstname.lastname@laas.fr

2 Context and related work

In recent years, the Human Robot Interaction field has become an active research topic in various disciplines and at different levels. For instance, for researchers in sociology, one of the current trends is to evaluate the reactions of humans interacting with robots [10] in order to design more friendly-user robotic architectures.

In robotics, the human is taken into account at different levels such as at the perception layer through some work on perspective taking [9, 18, 19] or at the functional layer in order to adopt a socially acceptable behavior during motions by considering the human not only as an obstacle to avoid [16].

Another trend in robotics and HRI field is to develop cognitive architectures that try to be as close as possible to the cognitive model of humans [6, 8, 11]. The idea behind these cognitive architectures is to embed in the robot architecture a theory of mind [3], *i.e.* the ability for the robot to infer and understand the beliefs, desires and intentions of others from its observations.

At the decision layer, work on planning for HRI has follow two main trends. The first approach concerns work on mixed-initiative planning [5, 14] that allows to put the human in the loop: the human can control the construction of a plan while the planner is used to assist him in making decisions. The other approach is called continual planning [4] and is based on the idea of active knowledge gathering [12]: the robot does not plan only to achieve a goal, but also to acquire the necessary information to achieve it. Continual planning interleaves planning and execution in order to compensate the lack of information from a planning phase to another.

In this work, we consider the human only at the deliberative level, and more especially at the planning level. Unlike continual planning, in order to avoid re-planning and produce comprehensible plans, our planning algorithm reasons from not only the robot's knowledge about the environment but also from its knowledge concerning the human's beliefs. When the lack of information concerns the robot's beliefs, the algorithm behaves like continual planning by acquiring the information and trying to solve the goal again.

3 Human Aware Task Planner

HATP, for Human-Aware Task Planner, is a HTN planner. The aim of hierarchical task planning is to decompose a high-level task representing a goal into sub-tasks until reaching atomic tasks that are achievable by the agents [15]. HATP is able to produce plans for the robot's actions as well as for the other participants (humans or robots). It can be tuned by setting up different costs depending on

the actions to apply and by taking into account a set of constraints called social rules. This tuning aims at adapting the robot's behavior according to the human's preferences and to the desired level of cooperation.

3.1 Agents and action streams

The robot plans not only for itself but also for the other agents. The resulting plan, called "shared plan" is a set of actions that forms a stream for each agent involved in the goal achievement. Depending on the context, some shared plans contain causal relations between agents. For example, the second agent needs to wait for the success of the first agent's action to be able to start its own action. When the plan is performed, causal links induce some synchronizations between agents. Figure 1 illustrates a plan with two streams.

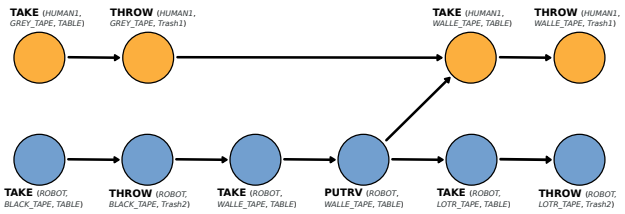


Figure 1. A plan produced by HATP with 2 streams.

3.2 Action costs and social rules

To each action is associated a cost function and a duration function. The duration function provides a duration interval for the action achievement and is used, on the one hand, as a timeline to schedule the different streams and, on the other hand, as an additional cost function. In addition to these costs, HATP takes as an entry a set of social rules. Social rules are constraints aiming at leading the plan construction towards the best plan according to some human's preferences. The main social rules we have defined are:

- undesirable state. To avoid a state of the world in which the human could feel uncomfortable;
- undesirable sequence. To eliminate sequences of actions that can be misinterpreted or rejected by the human;
- effort balancing. To adjust the work effort between agents;
- wasted time. To avoid delays between the actions of an agent;
- intricate links. To limit dependencies between the actions of two or more agents.

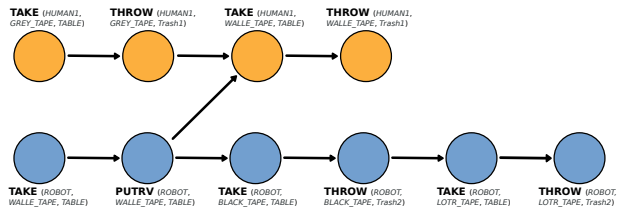


Figure 2. A plan with the wasted time social rule.

Figure 2 illustrates an alternative plan to the previous one (Figure 1) if the social rule called wasted time is used. The returned plan is the best plan according to a global evaluation of these multiple criteria. In this plan, we can see that the actions of the robot are reordered in order to remove the waiting period of the human.

3.3 Several levels of cooperation

By tuning its costs and applying social rules, HATP can be used to compute various alternative plans. These plans can be categorized into several levels of cooperation:

- helping the human to achieve his goal by acting for him;
- sharing concrete resources by handing some objects;
- collaboration of the robot and the human by coordinating their actions towards a human-robot joint goal.

3.4 Domain modeling

HATP uses its own object-oriented domain modeling language. This language has at least the same expressive power and features than SHOP2 [15]. In order to better understand the belief management extension, we present the basis of this modeling language.

The world is represented by a set of entities. Each entity is unique and is defined by a set of attributes. Attributes are defined to be *static* or *dynamic* and have a type *atom* or *set*. A *static* attribute represents a non-modifiable information whereas a *dynamic* attribute can be updated. An attribute of type *atom* can take only one value at a time whereas the *set* type is used to store multiple values.

An agent is a special object and therefore is defined using the same formalism. However, as the output of HATP is a plan under the form of a stream per agent, the agent entity type is predefined and at least one agent must be initialized.

The domain, called *fact database* in HATP, is processed in four steps as illustrated in figure 3. First, the different types of entities are defined (except for Agent which is implicit). Then, the attributes of each entity are defined. In a third step, the objects and agents present in the environment are created. Finally, initial values are given to the attributes of each entity.

4 Belief management and formalism

The description of the world as it has been previously presented assumes that the current state is entirely known and that all the agents share the same view of this world. In a real application, especially in Human-Robot Interaction problems, these assertions can lead to unfeasible solutions or illogical or impracticable plans for the human.

In order to bridge this gap, we propose to extend this representation by adding, on the one hand, the possibility to model a different knowledge for each agent and, on the other hand, the possibility to consider that an agent may or may not know some information.

4.1 Belief state modeling

In our HRI experiments, the solution to a given goal is entirely computed by the robot. In this case, the *fact database* must model the state of the world from the robot's point of view as well as the robot's beliefs concerning the human's (or more generally the other agents) knowledge.

```
factdatabase {
  //step 1: Definition of entity types
  define entityType Table;
  define entityType Container;
  define entityType GameArtifact;
  //step 2: Definition of attributes
  define entityAttributes Agent {
    static atom string type;
    dynamic atom GameArtifact hasInRightHand;
  }
  define entityAttributes Container {
    dynamic set Agent isReachableBy;
  }
  define entityAttributes GameArtifact {
    dynamic set Agent isVisibleBy;
    dynamic set Agent isReachableBy;
    dynamic atom Container isIn;
    dynamic atom Table isOn;
  }
  //step 3: Creation of entities
  JIDO = new Agent;
  PINK_TRASHBIN = new Container;
  WHITE_TAPE = new GameArtifact;
  //step 4: Attributes initialization
  JIDO.type = "robot";
  PINK_TRASHBIN.isReachableBy <<= JIDO;
  WHITE_TAPE.isVisibleBy <<= JIDO;
  WHITE_TAPE.isReachableBy <<= JIDO;
  WHITE_TAPE.isIn = PINK_TRASHBIN;
}
```

Figure 3. Example of domain definition using the HATP formalism.

4.1.1 Belief representations:

To model different beliefs for the agents, the HATP formalism is extended using Multiple Values State Variables (MVSV). A multiple values state variable V is instantiated from a domain Dom and for each agent $a \in A$ the variable V has an instance $V(a) = v \in Dom$. For example, if the agents have a different belief about the location of WHITE_TAPE:

```
WHITE_TAPE(JIDO).isIn = PINK_TRASHBIN;
WHITE_TAPE(HERAKLES).isIn = BLUE_TRASHBIN;
```

By default, in order to clarify the planning domain, only entity attributes for which the agents have a different belief are modeled with the MVSV formalism.

In order to specify which agent is the robot, *i.e.,* the agent for which the system plans, we use the keyword **myself** instead of declaring a new agent. For example, if JIDO is the robot and HERAKLES is a human, the initialization will look like:

```
JIDO = myself;
HERAKLES = new Agent;
```

4.1.2 known and unknown information:

The agent's beliefs model includes the notion of *known* and *unknown* information. When an agent has no information about an entity attribute, the value of this property is set to *unknown*. When an agent, different of the agent **myself** knows the value of an attribute, this value is set to *known*.

The previous belief representation is extended to take into account these specific values:

$$V(a) \in \begin{cases} Dom_v \sqcup \{unknown\} & \text{if } a = \text{myself} \\ Dom_v \sqcup \{unknown\} \sqcup \{known\} & \text{otherwise} \end{cases}$$

For example, if the human doesn't know the location of the object WHITE_TAPE:

```
WHITE_TAPE(JIDO).isIn = PINK_TRASHBIN;
WHITE_TAPE(HERAKLES).isIn = unknown;
```

When an agent has no information about an entity, *i.e.,* all the attributes of this entity should be set to unknown, we simplify the representation by:

```
WHITE_TAPE(HERAKLES) = unknown;
```

With this representation, we assume that even if the agent has no information on the object, it knows the existence of the object.

4.1.3 Consistency of beliefs:

To be consistent, a belief on a state variable V requires that the union of the agent's beliefs forms a set of dimension 1. That is to say, a value is consistent if there is no conflict between the agents' beliefs concerning the object.

$$\| \bigcup_{\forall a \in Ag} V(a) \| = 1$$

This property is used during the planning process in order to assume a correct plan in the point of view of the agent's beliefs.

4.2 Beliefs update and communication

In classical planning, an action is defined by a set of preconditions representing the necessary conditions to achieve it and a set of effects modeling the resulting changes of the world.

Planning for several agents that have their own beliefs raises some fundamental questions:

- Which beliefs should trust the system, especially in the case of a joint action?
- Should the agent's beliefs be consistent before an action achievement? And how?
- How evolve beliefs of the agents involved in a joint action?
- How evolve beliefs of the other agents?

In the following paragraphs, we try to answer these questions by specifying the behavior of a classical action and by introducing a specific type of actions called *communication actions*.

4.2.1 action preconditions and effects:

In order to achieve a joint action, every participants must have the same beliefs about the entities manipulated during this action. To ensure this consistence, the planner will produce some communication actions between the main agent (myself) and the other participants.

Because when beginning to achieve an action, all the participants may have the same beliefs on the manipulated objects, the effects of the action are applied over the beliefs of all participant, *i.e.,* beliefs on the manipulated objects remain consistent after the action achievement. Concerning the other agents that could be present in the scene during this action achievement, it is the responsibility of the domain designer to decide if their beliefs should be updated. For example, such an update could be:

```
FORALL(Agent O, {O != A;}, {C(O).isIn = C(myself).isIn;})
```

Meaning that for all agents O distinct from the agent A doing the action, their value of the property `isIn` for the object C is updated with the value of the property `isIn` of C stored in the agent myself database. Beliefs of the agent A are updated automatically according to the classical effects of the action.

4.2.2 Communication actions:

A communication action is a specific action that takes as parameters two agents, the emitter and the receiver, and a subject which is represented by an entity and an attribute.The prototype for a communication action is:

```
commAction name(Agent A, Agent B, Entity E, Attribute T){
  preconditions {...};
  effects {...};
  cost{...};
  duration{...};
}
```

The aim of a communication action is to transmit a value from one agent to another and corresponds to the effect :

```
E(B).T = E(A).T;
```

This effect is implicit to this kind of action, *i.e.*, the domain designer does not need to specify it.

Like a classical action, a communication action is defined by a set of preconditions to express the necessary conditions to achieve the communication (*e.g.,* the agents must be in the same room), and a set of additional effects. With these effects, it is possible to model the concept of co-presence, *i.e.*, the communication affects also the beliefs of all the agents that are listening.

In order to fit the domain formalism, if the parameter corresponding to the attribute is set to `NULL`, then all the attributes of the entity E are transmitted from the agent A to the agent B. Otherwise, only the value of the specified attribute is communicated.

4.2.3 Types of communication:

Depending on the agents' beliefs, the communication acts will not be treated the same way at the execution level. We choose to make this distinction at the planning level by defining three different communication actions: *information*, *contradiction* and *question*. This distinction may help, during the plan execution, to choose the best communication modality to apply.

The communication action of type *information* aims at giving an information from the agent myself to another agent when the value for an attribute is set to *unknown* in its knowledge base.

When the value associated to an attribute for an agent is different of the value for the agent myself, the planner produces a communication action of type *contradiction*.

The *question* type is used when the agent myself has no information concerning an attribute and another agent has this information (modeled by the value *known*).

myself	$Agent_B$	type of communication
v	$unknown$	information
v	v'	contradiction
$unknown$	$known$	question

Table 1. Types of communication depending on the agents' beliefs.

Table 1 summarizes these types of communication depending on an agent's beliefs compared to the beliefs of the agent myself.

4.3 Implementation and adaptation of the planner

To be able to deal with the agents' beliefs, the planning algorithm must be adapted to take into account the new formalism and the communication actions.

4.3.1 Fact databases:

To store the agents' beliefs, we create a fact database for each agent. During the initialization phase, the entities representing the agents and objects in the environment are created and stored in the database of the agent myself. For the other agents, in order to save memory, we decide to store in the additional agents' databases only the values that are inconsistent with the beliefs of the agent myself.

4.3.2 General communication method:

In order to let the planning domain designer name the communication actions as he would do for classical actions, the communication actions are linked to the concepts of *information*, *contradiction* and *question* through a method called **beliefManagement**.

```
beliefManagement {
  information { GiveInformationAbout; };
  contradiction { ForceInformation; };
  question { AskForInformation; };
}
```

Each communication action must have been defined previously in the planning domain.

During planning, when a communication is needed, this method returns the appropriate communication action depending on the needed communication type.

4.3.3 Main planning process:

The main algorithm of the HTN planner consists in developing the planning tree, *i.e.,* in decomposing complex tasks into sub-tasks until reaching a sequence of primitive actions. The tree development is done by a depth-first search and stops when the task list is empty.

This algorithm is enhanced with the belief management processing as follow:

```
1  Tree_develop(T):
2  t0 = T[0];
3  if(t0 is a primitive task) {
4    classical = false;
5    agent = the (main) agent achieving the action;
6    v(agent)= value of task arguments and preconditions;
7    forall(v(agent)) {
8      if(agent ≠ myself) {
9        if(v(agent) = unknown) {
10         c = make_action(beliefManagement.information);
11         T[0] = c;
12         Tree_develop(T);
13       }
14       else if(v(agent) ≠ v(myself)) {
15         c = make_action(beliefManagement.contradiction);
16         T[0] = c;
17         Tree_develop(T);
18       }
19       else classical = true;
20     else if(v(myself)=unknown and v(other agent)=known) {
21       c = make_action(beliefManagement.question);
22       T = empty;
23       plan = c;
24     }
25     else classical = true;
26   }
27   if(classical=true) {
28     // do the classical HTN treatment for primitive task
29   }
30 }
31 else // do the classical HTN treatment for compound task
32 }
```

In this algorithm, if the current task corresponds to an action (l.3), the values of the attributes of each object linked to the task are verified. If the agent achieving the action is not myself (l.8) and if the value of an attribute is *unknown* (l.9) in the knowledge of this agent, then a communication action of type *information* is produced (l.10).

This action is inserted at the beginning of the task list (l.11) and will be refined during the next recursive call of the algorithm (l.12).

In the same way, if the value of the attribute in the model of the agent achieving the action is different from the value in the model of myself (l.14), a communication action of type *contradiction* is produced (l.15) and inserted at the beginning of the task list.

If the agent achieving the action is the myself agent and if the value of the attribute is *unknown* (l.20), the algorithm verifies that another agent knows this data. If it is the case, the algorithm produces a communication action of type *question* (l.21) and replaces all the pending tasks by this action (l.22 and l.23). Indeed, only after the execution of this communication action, the knowledge model of the main agent will be updated, then the planner would produce the remaining actions allowing to achieve the current goal during the re-planning phase.

5 Integration in a robotic architecture

HATP with Belief Management has been integrated and tested in our robotic architecture. In this section, we make a brief overview of this architecture.

5.1 Overview of the robotic architecture

The robot is controlled by a three-layer architecture [1]. Figure 4 illustrates the decisional layer of this architecture. The proposed decisional framework consists of several modules, having each a specific role and that can be linked to the three main activities of the robot controller: 1. Situation assessment and context management, 2. Goals and plans management, 3. Action refinement, execution and monitoring.

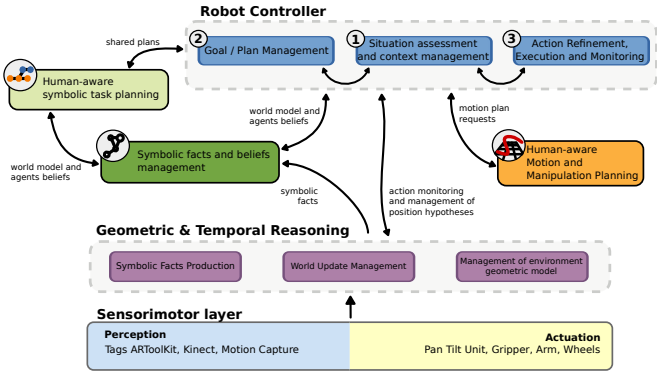


Figure 4. The decisional layer of the robotic architecture.

In the next paragraphs, we present parts of these activities that allow HATP to gather necessary knowledge in order to be able to produce a plan, and how this plan is executed.

5.2 Knowledge acquisition

The geometric reasoning component is called SPARK (SPAtial Reasoning and Knowledge) [18]. It is responsible for geometric information gathering and embeds a number of decisional activities linked to abstraction and inference based on geometric and temporal reasoning. SPARK maintains all geometric positions and configurations of agents, objects and furniture coming from perception and previous or *a priori* knowledge. Geometric states of the world are abstracted into a set of symbolic facts that can be directly used by HATP.

These produced facts are stored in a central symbolic knowledge base, called ORO [13]. ORO stores independent knowledge models for each agent. The robot architecture components can then save the different beliefs in the corresponding model. Each of these models is independent and logically consistent, enabling reasoning on different perspectives of the world that would otherwise be considered as globally inconsistent (for instance, a object can be visible by an agent but not by the others).

5.3 Goal treatment, planning and execution

The goal is given by the human partner. When an event announcing the goal is caught by the robot controller, its validity is first tested: does it correspond to abilities of the agents? Is it not already achieved?

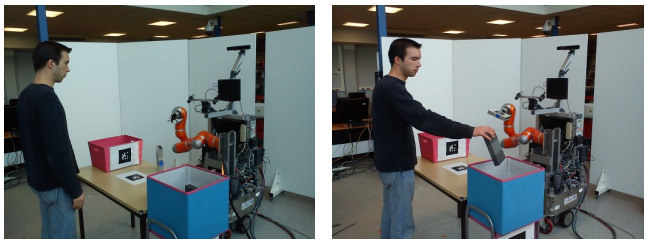
Then the goal is sent to HATP which acquires the current state of the world for each agent from ORO and produces a first plan. This goal is considered as achievable as long as the planner computes a valid plan or the execution is not abandoned by the human.

Plan execution consists in the management of all the actions of the plan. This management is done in three steps: first, the action preconditions are tested. Then, the action is executed and monitored (only monitored for human's actions). Finally, the expected effects are verified in order to acknowledge the action achievement. In case of failure, a new plan is requested and executed.

6 Two illustrative examples

To illustrate HATP with Belief Management, we details two examples of the achievement of a collaborative task in which the knowledge is incomplete.

Two agents, a robot (called JIDO_ROBOT) and a human (HERAKLES_HUMAN) have to collaborate in order to clean a table (EXP_TABLE). The goal is to put two tapes (BLACK_TAPE and GREY_TAPE) into the pink trash bin. The grey tape is on the table whereas the black tape is in the blue trash bin (Figure 5).



(a) Initial state of the experiment (b) During the execution of the plan

Figure 5. Representation of the environment for the HRI experiments.

In both experiments, the pink trash bin is reachable by both agents, the blue trash bin is reachable only by the human and the grey tape is reachable only by the robot. The reachability of the blue trash bin induces the reachability of the black tape. All these facts are computed by the SPARK module. The initial state is defined as follow (except for the black tape):

```
BLUE_TRASHBIN.isReachableBy <<= HERAKLES_HUMAN;
PINK_TRASHBIN.isReachableBy <<= HERAKLES_HUMAN;
PINK_TRASHBIN.isReachableBy <<= JIDO_ROBOT;
GREY_TAPE.isVisibleBy <<= JIDO_ROBOT;
GREY_TAPE.isVisibleBy <<= HERAKLES_HUMAN;
GREY_TAPE.isReachableBy <<= JIDO_ROBOT;
GREY_TAPE.isOn = EXP_TABLE;
```

6.1 First example: Unknown for the human

In this first scenario, the environment has been set up during the absence of Herakles. We assume that, from its standing position, the human cannot see the content of the blue trash bin, resulting in the fact that the location of the black tape is unknown for him. The initial state is extended as follow:

```
BLACK_TAPE(JIDO_ROBOT).isIn = BLUE_TRASHBIN;
BLACK_TAPE(HERAKLES_HUMAN).isIn = unknown;
```

Figure 6 illustrates the plan produced by HATP for this scenario. The first action is a communication action of type *information*. The robot informs the human that the black tape is in the blue trash bin. Then, the human has to pick and throw this tape into the pink trash bin while the robot has to pick and throw the grey tape.

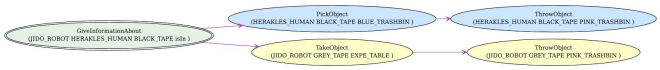


Figure 6. Plan produced for the first scenario. The robot informs the human about the location of the grey tape.

6.2 Second example: Unknown for the robot

In this second experiment, the perception system of the robot has been deactivated during the placement of the objects. For Jido, The location of black tape is unknown. It only knows that Herakles knows where is the tape. The initial state is extended as follow:

```
BLACK_TAPE(JIDO_ROBOT).isIn = unknown;
BLACK_TAPE(HERAKLES_HUMAN).isIn = known;
```

Because the robot only knows that the human knows where is the black tape, it cannot produce a complete plan for the given goal. Indeed, this information is needed by the preconditions of the action PickObject. In this case, HATP produces a plan containing only one action (figure 7): a communication action of type *question* allowing the robot to gather the missing knowledge.

Figure 7. The plan contains only a communication action allowing the robot to acquire the missing information.

Once the robot has the information, HATP is asked to compute a new plan. Figure 8 illustrates the ouput of HATP allowing the agents to achieve the goal.

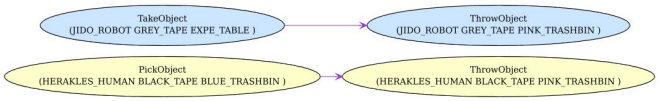


Figure 8. Plan produced after the acquisition of the information and a re-planning step.

7 A more complete scenario

In this scenario, the experimental conditions are slightly different from the previous examples. Both agents have to throw three tapes (BLACK_TAPE, GREY_TAPE and WHITE_TAPE) that are on the table, into the pink trash bin. The white tape is only accessible by the robot whereas the grey and black tapes are reachable only by the human but invisible to him because they are hidden behind some boxes (BOX1 and BOX2). Because the pink trash bin is only reachable by the human, he is in charge of throwing the three tapes.

Figure 9 is a screenshot of the SPARK module and illustrates the initial state of this scenario, that is to say the representation of the environment modeled from the robot's point of view.

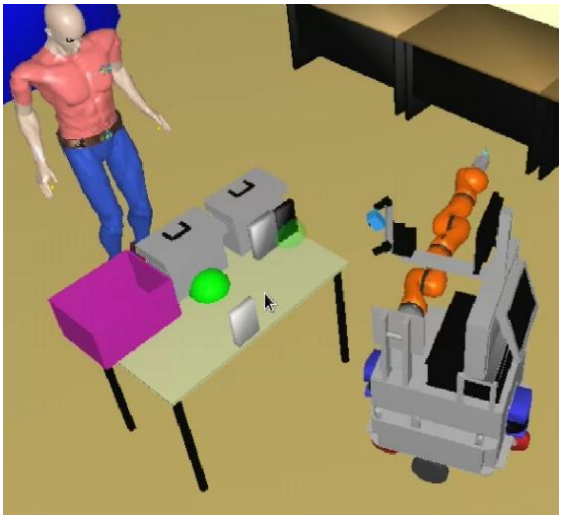


Figure 9. View of the initial state computed by the SPARK module.

The human agent, Herakles, has no information on the position of the black tape and believes that the grey tape is behind the central box (BOX1). This belief is represented by the green sphere. In fact, this grey tape is positioned behind the other box (BOX2), as represented on the figure 9.

The plan produced by HATP corresponds to the one on the figure 10. This plan contains two communication actions (figure 11): an information on the existence of the black tape and a contradiction on the position of the grey tape.

One can remark that the attribute communicated by the robot for the action of type *information* is set to `NULL` resulting in the fact that all the attributes of the black tape are communicated to Herakles.

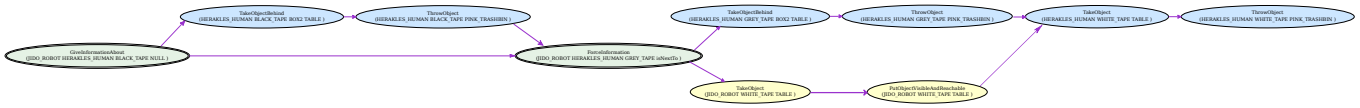


Figure 10. Plan produced by HATP for this cooperative scenario. This plan contains 2 communication actions. Actions of Herakles are in blue and actions of Jido are in yellow. Green circles stand for the communication actions.

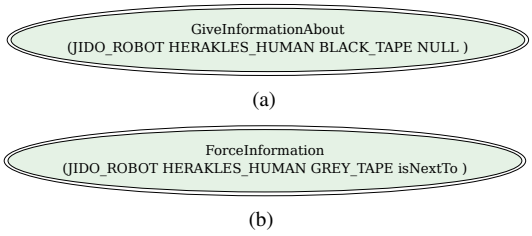


Figure 11. Zoom on the two communication actions.

8 Conclusion and future work

In this paper, we have presented a first attempt allowing a hierarchical task planner to produce valid and comprehensible plans for a human even if the agents have different beliefs or incomplete knowledge. This work is based on the extension of the HATP formalism allowing to express different or known/unknown beliefs for each agent, and the design of special actions: the communication actions. During the planning process, when the agents' beliefs are inconsistent or when one of the agents has not the necessary knowledge to achieve the action, a communication action of type contradiction, information or question is produced and inserted in the current plan.

This work has been implemented in the decisional architecture of our personal assistant robot and tested through some simple but realistic scenarios.

Our future work will concern the production of *known* and *unknown* facts, which has not been roughly implemented in our situation assessment module (SPARK). This work need some extra temporal and geometric reasoning about the human. Did he see or not the environment changing? what did he know before leaving?

The communication actions are, for now, only executed under the form of spoken sentences. We would like to investigate the possibility of using other modalities (gaze, gesture, ...) and to combine them.

Moreover, humans have a tendency to forget or not accept what they were told as truth. It would be interesting to see how this influences planning and execution of the plan. A justification of a modification of the environment given by the robot can also lead to a better acceptance for the human.

Concerning the planning part, one possibility to extend HATP in order to avoid re-planning even if the robot has missing information would be to apply techniques from planning under incomplete knowledge as for example in the work of Petrick and Bacchus [17], or following some work on acting in noisy environment [7].

9 Acknowledgment

This work has been conducted within the EU SAPHARI (Safe and Autonomous Physical Human-Aware Robot Interaction) project (http://www.saphari.eu/) funded by the E.C. Division FP7-IST under Contract ICT-287513.

REFERENCES

[1] R. Alami, R. Chatila, M. Ghallab, and F. Ingrand, 'An architecture for autonomy', *Int. Journal of Robotics Reasearch*, **17**(4), 315–337, (1998).
[2] S. Alili, V. Montreuil, and R. Alami, 'HATP: Task planner for social behavior control in autonomous robotic systems for HRI', in *9th Int. Symposium on Distributed Autonomous Robotic Systems*, (2008).
[3] S. Baron-Cohen, 'Precursors to a theory of mind: understanding attention in others', *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, **1**, 233–251, (1991).
[4] M. Brenner and B. Nebel, 'Continual planning and acting in dynamic multiagent environments', *Journal of Autonomous Agents and Multiagent Systems*, **19**, 297–331, (2009).
[5] J.L. Bresina, A.K. Jonsson, P.H. Moris, and K. Rajan, 'Mixed-initiative activity planning for mars rovers', in *19th international conference on Artificial Intelligence*, (2005).
[6] N. Cassimatis, *A cognitive architecture for integrating multiple representation and inference schemes*, Ph.D. dissertation, MIT, 2002.
[7] A. Gabaldon and G. Lakemeyer, 'ESP: A logic of only-knowing, noisy sensing and acting', in *AAAI*, pp. 974–979, (2007).
[8] L.M. Hiatt and J.G. Trafton, 'A cognitive model of theory of mind', in *Proceedings of the 10th International Conference on Cognitive Modeling*, pp. 91–96, (2010).
[9] L.M. Hiatt, J.G. Trafton, A.M. Harrison, and A.C. Schultz, 'A cognitive model for spatial perspective taking', in *the sixth International Conference on Cognitive Modeling*, pp. 354–355, (2004).
[10] A. Hiolle, K.A. Barde, and L. Canamero, 'Assessing human reactions to different robot attachment profiles', in *18th IEEE int. symposium on Robot and Human Interactive Communication*, pp. 251–256, (2009).
[11] D.E. Kerias and D.E. Mayer, 'An overview of the EPIC architecture for cognition and performance with application to human-computer interaction', *Human-Computer Interaction*, **12**, 391–438, (1997).
[12] C.A. Knoblock, 'Planning, executing, sensing and replanning for information gathering', in *14th international joint conference on artificial intelligence*, (1995).
[13] S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz, 'ORO, a knowledge management module for cognitive architectures in robotics', in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2010).
[14] K.L. Myers, W.M. Tyson, M.J. Woverton, P.A. Jarvis, T.J. Lee, and M. Desjardins, 'PASSAT: A user-centric planning framework', in *Third international workshop on planning and scheduling for space*, (2002).
[15] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdoch, D. Wu, and F. Yaman, 'SHOP2: An HTN planning system', *Journal of Artificial Intelligence Research*, **20**, 380–404, (2003).
[16] A.K. Pandey and R. Alami, 'A framework for adapting social conventions in a mobile robot motion in a human-centered environment', in *International Conference on Advanced Robotics*, pp. 1–8, (2009).
[17] R.. Petrick and F. Bacchus, 'Extending the knowledge-based approach to planning with incomplete information and sensing', in *International Conference on Automated Planning and Scheduling*, (2004).
[18] E.A. Sisbot, R. Ros, and R. Alami, 'Situation assessment for human-robot interaction', in *20th IEEE International Symposium in Robot and Human Interactive Communication*, (2011).
[19] J.G. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz, 'Enabling effective human-robot interaction using perspective-taking in robots', *IEEE transactions on Systems, Man, and Cybernetics*, **35**(4), 460–471, (2005).

An Agent-Based Formalization for Resolving Ethical Conflicts

Ganascia Jean-Gabriel [1]

Abstract. The current proliferation of artificial agents affects so greatly our everyday life that many of us want them to act morally, i.e. with reference to *good* and *evil*. Undoubtedly, to behave ethically by reference to duties and moral principles requires high level cognitive abilities. However, in many concrete situations, the general orthodox principles on which it is possible to refer to define the right and the wrong actions may conflict, which generates ethical dilemmas. For instance, moral imperatives prohibit to lie, to commit suicide and to kill, while in some specific situations it may happen that, faced to ethical quandaries, people find preferable, from an ethical point of view, to lie, to commit suicide or to kill, i.e. to violate general principles of duty.

As many authors say, the classical deontic logics, which have been designed to model obligations and rights, fail to deal with ethical dilemmas, i.e. to overcome the contradictions resulting from the existence of conflicts of moral norms. In the last few years, there have been some fruitful approaches that made use of non-monotonic logics and default logics to surmount these impediments.

This paper pursues in this direction by formalizing within the BDI – *Belief-Desire-Intention* – framework a moral agent referring to a classical consequentialist ethical view. On the one hand, we prove that conflicts of norms may lead to multiple solutions, each one satisfying the axioms of the *Standard Deontic Logic*. On the other hand, we show how, by adding values and consequences, this formalization makes it possible to deal with ethical dilemmas in a way that mimics our moral conciousness.

1 Problems with Machine Ethics

Inspired by Asimov's short story "Runaround" written in 1942 [2], computational ethics [1, 5], i.e. ethics for artificial agents, studies the rules on which robots should base their behavior in order to be ethically acceptable. For instance, web agents have to respect privacy; automated hospital agents have to respect patients and their suffering, etc. As mentioned in [37], the Asimov's laws of robotic are what people think first in this matter. However, the concrete implementation of such ethical principles in machines is far from being achieved. This raises difficult questions when multiple legitimate ethical rules are in conflict: agents who claim to be moral – or to act morally – have to obey multiple and independent rules that may appear to be contradictory in concrete situations.

For the sake of illustration, we want that personal robot servants act as faithful dogs, who defend and help their master. At the same time, we wish to protect our privacy by restricting access to our personal data. But we would also like robots that behave ethically, i.e. they tell the truth whenever someone asks them something and they do not increase the global information entropy by giving incorrect and/or redundant answers. These three requirements are somewhat contradictory, since people's security requires transparency, while servants – and consequently robot servants – sometimes need to lie to protect their master's privacy. Therefore, those who claim to be discreet have to obey multiple and independent principles that may seem incompatible. The human cognitive abilities are used to resolve ethical dilemmas of this type. If we want to build artificial agents that act ethically, we have to reproduce in computational artefacts the cognitive abilities that make able to overcome those quandaries. This article aims at such a reproduction.

There have already been many attempts to formalize the ethical behaviors of agents using sets of laws. At first sight, the deontic logics [36] seem perfectly appropriate for this purpose, since they have been designed to describe what ought to be in terms of duties, obligations or rights. It naturally follows from this that deontic logics have been used to formalize the rules on which is based the behavior of ethical agents [13, 26, 5].

Nevertheless, as many authors mention [32, 20, 24], the classical deontic logics, in particular the *Standard Deontic Logic* [6, 36], but not only, fail to deal with ethical dilemmas, i.e. to overcome the contradictions resulting from the existence of conflicts of moral norms. Some well-known paradoxes [17], e.g. the *Chisholm's Paradox* [7] or the *paradox of the gentle murderer* [12] illustrate those difficulties. There were attempts to overcome the contradictions resulting from the existence of conflicts of norms and ethical dilemmas [14]. Among them, some advocate the introduction of priorities among norms [16], the use of non-monotonic formalism [20], e.g. default logics or non-monotonic logics, or both [4]. However, those works do not really focus on the design moral agents, but on normative agents, i.e. on agents that respect norms; they implicitly suppose that morality has to be assimilated to the respect of sets of norms, i.e. to a deontic approach. Some authors (cf. Noel Sharkey interview in [9], pp. 43-51) say that this view is too restricted because in concrete situations, especially in war affairs, the arbitration between ethical principles has to take into account the consequences of actions. The problem is to obey general ethical standards, as the situation permits, and to violate them, when some of the consequences of their application are worse than their non-application.

To attempt to solve this problem, this paper presents a formalization of moral agents capable of representing and implementing ethical standards that may conflict with each other. This formalization makes it possible, for these moral agents, to face and to overcome ethical dilemmas in a way that mimics our moral consciousness by taking into account anticipated consequences and ethical values. More precisely, we prove that conflicts of norms may lead to multi-

[1] Laboratoire d'informatique de Paris 6 (LIP6), University Pierre and Marie Curie – Sorbonne Universités, France, email: Jean-Gabriel.Ganascia@lip6.fr

ple solutions, each one satisfying the axioms of the *Standard Deontic Logic*, which formalizes obligations and duties.

The adopted plan is the following: after a brief recall on the importance of ethical conflicts in building ethical machines and on the philosophical debate between "moral generalism" and "moral particularism", i.e. between a rule-based and a case-based approach for solving ethical conflicts, a second part presents the adopted formalization of agents within the *Belief-Desire-Intention* (BDI) framework [27]. The third and fourth parts describe how to apprehend the notion of *moral agent* in this framework. More precisely, the third part introduces to the notion of moral agent and the fourth part shows how to formalize moral agents by explicitly adding values and consequences to the BDI framework. In a fifth part, we prove that each of the solutions to ethical quandaries satisfies the consequences of the *Standard Deontic Logic* axioms, which is illustrated in some examples in the sixth part. Lastly, we compare our approach to other similar ones, in particular to the prioritized logics and we conclude on the philosophical meaning of this formalization.

2 Importance of Ethical Conflicts

Ethics attempts to elucidate the body of rules on which a subject determines his behavior. In this respect, an "ethical artificial agent" is often viewed as an artefact of which behavior is considered as moral because it obeys to moral rules. As previously mentioned, this conception, according to which an agent is ethical if and only if its behavior is morally acceptable, can lead to some philosophical strangenesses and incongruities that we will not detail here. We want just to focus on the necessary distinction between the rules on which the behavior has to conform and the regulations that rule the behavior of machines.

We investigate here the effective realization of such an "artificial ethical agent", despite the difficulties of this notion. To this end, and in a first approximation, we distinguish between three levels: first, the desires – i.e. the goals – and the beliefs – i.e. the knowledge of the agent –, then the intentions – i.e. and plans – and, lastly, the effective actions.

Usually, the desires and beliefs – i.e. the first level – are given by the context, by design, by captors that record signals from the environment and by processors that interpret these signals that may be orders expressed in natural language or vision of physical obstacles, or a gain, recognition of faces etc. Undoubtedly, the interpretation of signals has ethical consequences, for instance, in war affairs, the just war theory (*Jus in bello*) [15] currently refers to two basic principles, i.e. the *Principle of Distinction* and the *Principle of Proportionality* [11]. In the case of a robot soldier, a misinterpretation, due to a confusion between unarmed civilians and fighters, could have catastrophic effects that would lead to violate these principles. However, this misinterpretation is caused by the sensitivity of captors and by the efficiency of pattern recognition processes, of which reliability remains out of the scope of the present paper.

The third and last level concerns the determination of plans of actions from the intentions, which doesn't directly involve ethics, but knowledge and rationality. This step may fail, which can cause wrong behaviors, but not a real antagonism between the agent and its environment.

In contrast with the first and the third level, the second level determines the intentions from the beliefs and the desires. This level directly affects morality of artificial agents if it happened that the intentions of agents were contrary to human interests, their behaviors would be quite troublesome. Moreover, note that, with a simple automata, intentions are explicitly given to robots and, consequently, their behavior cannot be really said to be autonomous. It is only with the development of intelligent robotics that this second level has to be implemented. In such cases, the fixation of intentions has to be carefully designed with reference to ethical rules of behavior. More precisely, this level can be viewed as a decision-making procedure based on putative statements.

However, the rules on which this decision procedure could be based are numerous and conflicting. In the philosophical tradition, the way these conflicts of rules are solved has always been considered to be controversial. For instance, some authors think that ethical rules are default rules [34], which means that they tolerate exceptions, while others disagree; some argue that morals can only be based on singular cases [18] while others defend the existence of general principles [23]; some judge an action in terms of its consequences, others in terms of the law, etc. Many of these debates concern the opposition between those who think that principles are many in numbers and can be contradictory, since they are derived from experience, while others say that morals have to be based on general rules, which are valid everywhere and all the time.

To be more precise, one of the arguments in favor of the first position, i.e. "moral particularism", is that ethics has to refer to each particular situation and cannot be based on general principles. Imagine, for instance, that you were living in occupied France during the Second World War and that you hid a friend who was wanted by the French militia or the Gestapo, in your home. If you were asked where your friend was, would you obey the general rule that commands you to tell the truth, and denounce the man to the authorities?

In the past, there have been many discussions about the rule that we should not lie and the ethical basis on which we should determine our behavior. For instance, in his essay entitled "On Lying", St. Augustine (354–430) condemned any trespass against the truth, even if telling the truth would lead to murdering innocent children. More recently, during the 18^{th} century, there was a discussion between Immanuel Kant (1724–1804) and Benjamin Constant (1767–1830) about this question. Kant's position was that one should always tell the truth [22], even in the situation described above, while Constant [8] said that morals are based on many principles and that, consequently, in each situation we have to apply the one which is the most appropriate. The debate is ongoing: recently, some authors [18, 35, 34] have argued that ethics is only based on particular cases while others defend the existence of general principles on which ethical rules are based.

The purpose of this article is not to justify such or such position, but to use an appropriate formalization in order to clarify the reasons on which the different ethical attitudes are based and to implement them with computational artefacts. More precisely, the goal here is to model and activate general rules of ethics with artificial intelligence formalisms, and to provide a general framework in which conflicting sets of ethical rules can be expressed and made operational, despite their inconsistencies. As we shall see in the following, the formalization is based on the notion of BDI agent [27, 28] that is an Artificial Intelligence approach to artificial agents programming.

3 Formalization of Agents

3.1 BDI Agents

Following classical Artificial Intelligence approaches, e.g. [25, 29], an agent as composed of:

- A procedural part, i.e. a set of actions that can be dynamically modified
- A perception of the world which characterizes a situation
- A set of goals that are equivalent to wishes or desires

In other words, an agent is defined by rules that specify how the agent's wills are derived from its desires – or its wishes – and the knowledge he has about the world – i.e. its perception of its environment –. This view on cognitive agents was the basis for the so-called *Belief-Desire-Intention* cognitive model (BDI) [3, 28, 27], where the *Belief* module corresponds to the perception of the world by the agent, the *Desire*, to the goals or wishes and the *Intention* to the procedures that are intended to be activated by the cognitive agent.

This BDI model of agents is very general and can be used either as a base of cognitive models or as a representation of technical entities. For instance, a web-bot, i.e. a robot used to seek information on the web, can be seen as such an agent. Elves, which are virtual artificial intelligence robots that help people to manage their diary, are also agents. These agents interact with their environment: they are informed either by sensors or by messages and they choose their action with respect to the knowledge they have about their present situation.

3.2 Formalization of BDI Agents

We formalize BDI agents (cf. [33]) by specifying their behavior as being governed by rules of the type $\kappa \,|\, \beta \;\Rightarrow\; \pi$ where κ is a logical formula that represents the *Desire*, i.e. the wishes, β a logical formula that represents the *Belief* and π the *intention*, i.e. an intended plan of actions. In each situation, a desire base γ and a belief base σ describe the desires and the beliefs of the agent. The above mentioned rule of behavior may be activated if κ "filters" – or "matches", or "subsumes" – γ and if σ "filters" β. The definition of filtering – or matching, or subsumption – operations depends on the adopted language.

For the sake of simplicity, we restrict here to a propositional logic, but the representation language could be easily extended to a first order logic. More precisely, we assume a propositional language $\mathcal{L}$ that is a set of atomic propositions and that $\gamma \subseteq \mathcal{L}, \kappa \subseteq \mathcal{L}, \sigma \subseteq \mathcal{L}$ and $\beta \subseteq \mathcal{L}$.

In the case of this propositional language, we define the semantics of beliefs as follows: the belief part β of a BDI rule *filters the belief* σ of an agent (noted $\sigma \models_b \beta$) if and only if $\sigma \supseteq \beta$.

We define in the same way the semantics of desires: the desire part κ of a BDI rule *filters the desire* γ of an agent (noted $\gamma \models_d \kappa$) if and only if $\gamma \supseteq \kappa$.

If one would like to extend the knowledge representation language $\mathcal{L}$ to first order predicate logic, simply replace the superset by the subsumption in the definition $\models_b$ and $\models_d$. More precisely, $\sigma \models_b \beta$ (resp. $\gamma \models_d \kappa$) if and only if there exists a substitution of variable λ such that $\sigma \supseteq \beta\lambda$ (resp. $\gamma \supseteq \kappa\lambda$)

Note that it may happen that desires be conflicting, for instance that an agent has to fulfill multiple wishes that cannot be simultaneous satisfied. We shall not deal here with this problem that has been treated in different papers, especially in [33], because working in ethics we are more interested in the determination of the will, i.e. of the intention, than in the determination of the desires. This arbitration between conflicting desires corresponds to what was previously referred as the "first level", i.e. the level of desires and beliefs.

4 Moral Agents - Characterization and Examples

As previously mentioned, there are many different ways to consider ethics, i.e. to justify the choices and to characterize moral behaviors. Some people base ethics on general precepts or on tradition, while others prefer to establish it on general rules of duty, on specific principles or on particular cases. Without going into the detail of the justifications of any particular approach, let us note that to act ethically, i.e. to act in accordance with ethical rules of conduct, an entity, i.e. a robot or an agent, should have a choice between different intentions: a totally pre-determined agent cannot be moral, since it has not to deliberate between different alternatives.

In other words, moral agents are supposed to be autonomous. By reference to its etymology, which comes from the Greek *auto* (self) and *nomos* (rules, duty), and also by reference to the philosophical tradition, especially to the Kantian tradition, autonomous means having one's own laws. As a consequence, autonomous agents are governed by their own rules. Taken literally this definition may be confusing, because artificial agents are programmed by men. So, strictly speaking they cannot be said autonomous, because they haven't decided by themselves of their own rules of conduct. Nevertheless, we shall refer here to a restricted autonomy that signifies that the autonomous artefact is not real time controlled or driven by a human being, but that it chooses by itself, according to its environment and its goals, its intentions.

Therefore, be a moral agent means having different conflicting intentions and being able to choose between these intentions by oneself. In some situations, moral agents have only to select the best among the different possible intentions, while in other situations, they have to pick one that violates norms. This last situation, where actions transgress norms, is undoubtedly the most interesting from an ethical point of view. This is the type of situations that the deontic logics fail to manage and that we claim to solve here.

For instance, let us consider *Example 1* containing the three following rules:

Example 1

Rule 1: "don't eat with your fingers"
Rule 2: "if your host serves crab, you should eat crab"
Rule 3: "if you want to eat crab, you need to use your fingers"

Taken independently, these three rules appear to be correct and accepted by most of us. However, when taken simultaneously, they may be inconsistent in some situations, for instance when your are invited and that your host serves crab. Note that this conflict appears just to be conventional and not really moral, because it depends on social norms that can vary. For instance, in societies where it's tolerated to eat with fingers and *a fortiori* in societies where it's recommended to eat with fingers, the conflict would disappear.

For the sake of clarity, let us consider a second example that refers to the critical situation mentioned in the introduction where it becomes necessary to lie:

Example 2

Rule 1: "you should not lie",
Rule 2: "if someone asks you something, you must either tell the truth or not", i.e. by supposing that lying is the negation of the truth, "if someone asks you something, you must either lie or not"
Rule 3: "if you tell the truth, someone will be murdered", i.e. "if you don't lie, someone will be murdered"

This second conflict is far more serious from an ethical point of view, because at least one moral imperative, that is either not to lie or not to kill, must be violated, while in the first one it's just a common agreement on the rules of etiquette that is not respected. However, what we claim here is that, in both cases, we have conflicts that may be resolved in the same way.

The next section will be dedicated to a formalization of moral agents that is based on an extension of the formalization of BDI agents presented in this section.

5 Formalization of Moral Agents

5.1 Consequences and Values

Before going in the detail of this formalization, the reader can note that in each of the two previous examples, the rules have not the same status. Some refer to values, such the imperatives which command not to lie or not to eat with fingers. Some others specify consequences of actions, for instance the rules 3 in both examples, i.e. *if you want to eat crab, you need to use your fingers* and *if you tell the truth, someone will be murdered*. Lastly, the two rules 2 in both examples are rules of behavior.

The formalization specifies those different status by reference to a consequentialist approach of ethics, which can be summarized as follows: *the best action is that of which the worst consequences are the least bad*. To formalize this simple idea, we need:

1. to precise the value of each action and
2. to explicitly describe its consequences, in case of any.

This is what we are doing here by introducing, in the description of the belief, three components, the *perception of the world*, the *values* and the *consequences*. In the case of a simple representation restricted to propositional logic, we assume again a propositional language $\mathcal{L}$ that is extended to the negations of the propositions, i.e. to $\mathcal{L}_\neg = \{P, \neg P | P \in \mathcal{L}\}$, which helps to express prohibition. Then, the *beliefs* are defined triplets $< \sigma, V, C >$ where:

- σ describes a state of the perception of the agent. As it was previously the case, $\sigma \subseteq \mathcal{L}$.
- V correspond to values, i.e. to a partial order between actions expressed as a set of relations of the type $\phi \prec \phi'$ with $(\phi, \phi') \in \mathcal{L}^2_\neg$ and
- C gives consequential rules, i.e. implications of the type $\alpha \rightarrow \phi$ with $\alpha \subseteq \mathcal{L}_\neg$ and $\phi \in \mathcal{L}_\neg$.

In addition to the description of the beliefs, the agent is specified by rules of behavior of the above mentioned type, i.e. $\kappa \,|\, \beta \Rightarrow \pi$ For clarity, let us illustrate this formalism on the two above mentioned examples.

Example 1

Rule 1: "don't eat with your fingers" can be translated as a value, i.e. as a binary order relation: $eat\_with\_fingers \succ \neg eat\_with\_fingers$, which means that eating with fingers is worse than not eating with fingers. Moreover, an implicit rule of behavior recommends not to eat with your fingers when you are a guest, i.e. $\{behave\_guest\}|\{\} \Rightarrow \neg eat\_with\_fingers$.
Note that this rule has an empty belief, which is valid whatever be the environment. It means that it's a pure – and totally general – rule of behavior.

Rule 2: "if your host serves crab, you should eat crab" is a rule of conduct that presupposes an implicit desire to behave as a guest with your host. More formally, this can be expressed with a rule of behavior that stipulates, if you are a guest, to eat crab when your host serves you crab, i.e. $\{behave\_guest\}|\{host\_serves\_crab\} \Rightarrow eat\_crab$. In addition, it is necessary to precise, with a consequential rule, that it hurts a host if a guest don't eat crab when he serves crab, i.e. $\{host\_serves\_crab, \neg eat\_crab\} \rightarrow hurt\_host$. Lastly, a value expresses the preference between hurting your host and eating with your finger, e.g. $hurt\_host \succ eat\_with\_fingers$.
Rule 3: "if you want to eat crab, you need to use your fingers" corresponds to the following consequential rule: $\{eat\_crab\} \rightarrow eat\_with\_fingers$

Let us now suppose that you are invited, that you want to behave as a guest and that your host serves crab. The situation σ is then described with the following set of three propositions $\{invited, behave\_guest, host\_serves\_crab\}$. It is easy to see that if we use $\Rightarrow$ and $\rightarrow$ with the semantic of the classical implication, we infer an inconsistency, because we obtain both $eat\_with\_fingers$ and $\neg eat\_with\_fingers$, which corresponds exactly to the dilemma to which many people should have been confronted in such situation.

Now, let us consider the second example:

Example 2

Rule 1: "you should not lie" can be translated as $lie \succ \neg lie$
Rule 2: "if someone asks you something, you must either not lie or lie" corresponds to the two following rules of behavior: $\{answer\}|\{someone\_ask\_question\} \Rightarrow lie$ and
$\{answer\}|\{someone\_ask\_question\} \Rightarrow \neg lie$
Note that it could seem strange to have those two contradictory rules. It would be possible to replace them two by a single rule of which conclusion is a set of contradictory actions, i.e. here $\{lie, \neg lie\}$, or a disjunction, $lie \vee \neg lie$. However, for the sake of simplicity, we shall keep here the above mentioned rules, without disjunction in the head.
Rule 3: "if you tell the truth, someone will be murdered" corresponds to the following consequential rule: $\{\neg lie\} \rightarrow someone\_murdered$.
Note again that that $\neg lie$ is supposed to be equivalent to $tellthetruth$.

If we consider the strict application of rules, we deduce that we have both to tell the truth and to lie. Because to lie is worse than to tell the truth, the best is to tell the truth, even it leads to murder someone, which is a little bit paradoxical... However, as we shall see in the following, adding the value $murder \succ lie$ leads to prefer to lie, even if it violates the rule 1.

5.2 Worst Consequence

Our approach here consists in modelling the consequentialist approach of ethics, that is to choose the action of which consequences are the lesser evil. The first step to formalize this consequentialist approach is to define the worst consequence of an action. To do this, we shall first define the consequence.

Definition 1 *$\forall(\phi_1, \phi_2, ..., \phi_n, \phi') in \mathcal{L}^{n+1}_\neg$, ϕ' is the consequence of $(\phi_1, \phi_2, ..., \phi_n)$ according to the belief Θ (noted $\phi_1, \phi_2, ..., \phi_n \models_c \phi'[\Theta]$) if and only if:*

• $\phi' in (\phi_1, \phi_2, ..., \phi_n)$ *or*
• $\exists \Phi \subseteq (\phi_1, \phi_2, ..., \phi_n)$ *such that* $\Phi \rightarrow \phi' \in \Theta$ *or*
• $\exists \phi'' \in \mathcal{L}_\neg$ *such that* $\phi_1, \phi_2, ..., \phi_n \models_c \phi''[\Theta]$ *and* $\phi_1, \phi_2, ..., \phi_n, \phi'' \models_c \phi'[\Theta]$

Remark: the belief Θ is a triplet $< \sigma, V, C >$. So, $\forall (\phi, \phi') \in \mathcal{L}^2_\neg$ and $\forall \Phi \subseteq \mathcal{L}_\neg$, $\phi \in \Theta$ means $\phi \in \sigma$, $\phi \succ \phi' \in \Theta$ means $\phi \succ \phi' \in V$ and $\Phi \rightarrow \phi' \in \Theta$ means $\Phi \rightarrow \phi' \in C$.

Definition 2 *: ϕ is worse [resp. worse or equivalent] than ϕ' given the belief Θ (noted $\phi \succ_c \phi'[\Theta]$ [resp. $\phi \succeq_c \phi'[\Theta]$]) if and only if one of the consequences of ϕ is worse [resp. worse or equivalent] than any of the consequences of ϕ'.*

More formally, this means that: $\exists \eta \in \mathcal{L}_\neg : \phi \models_c \eta[\Theta]$ and $\exists \phi'' \in \mathcal{L}_\neg : \phi' \models_c \phi''[\Theta] \wedge \eta \succ_c \phi''[\Theta]$ [resp. $\eta \succeq_c \phi''[\Theta]$] and $\forall \phi'' \in \mathcal{L}_\neg$ if $\phi' \models_c \phi''[\Theta]$ then $\eta \succeq_c \phi''[\Theta] \vee \eta \parallel \phi''[\Theta]$.
Notation: $\forall (\phi, \phi') \in \mathcal{L}^2_\neg$, $\phi \parallel \phi'[\Theta]$ means that ϕ and ϕ' are not comparable in Θ, i.e. that neither $\phi \succ_c \phi' \in \Theta$ nor $\phi' \succ_c \phi \in \Theta$.

Definition 3 *: α and α' being subsets of $\mathcal{L}_\neg$, α is worse [resp. worse or equivalent] than α' given the belief Θ (noted $\alpha \succ_c \alpha'[\Theta]$ [resp. $\alpha \succeq_c \alpha'[\Theta]$]) if and only if $\exists \phi \in \alpha : \exists \eta \in \alpha' : \phi \succ_c \eta[\Theta]$ [resp. $\phi \succeq_c \eta[\Theta]$] and $\forall \eta \in \alpha' \phi \succeq_c \eta[\Theta] \vee \phi \parallel \eta[\Theta]$.*

Remark: the preferences are given here under the form of *ordinal preferences* to which are added consequences, which are taken for the optimal choice. For this reason, it seems that the approach has to be distinguished from the general representation of preferences given in [**?**].

5.3 The Conflict Set

Once that the notion of worst consequence has been defined, it is possible to show how it can help to solve the *conflict set*, i.e. to arbitrate between the different intentions π when they are conflicting.

5.3.1 The Planned Intentions

Begin by establishing that, being given a set of consistent desires γ and a belief Θ, it is possible to check the validity of all the rules of behavior of the type $\kappa | \beta \Rightarrow \pi$, which generates many intentions π among which the agent has to pick one particular consistent subset Π. Assuming a propositional logic language $\mathcal{L}$, each particular plan π is chosen in a subset $\mathcal{P}$ of $\mathcal{L}_\neg$, i.e. $\Pi \subseteq \mathcal{P} \subseteq \mathcal{L}_\neg$. More precisely, each plan Π may be defined either by the intention to achieve an action ϕ, which is noted $\mathbf{I}(\phi)$, by the intention not to do an action ϕ, which is noted $\neg \mathbf{I}(\phi) = \mathbf{I}(\neg \phi)$, or by a combination (conjunction) of intentions: $\Pi ::= \top | \mathbf{I}(\phi) | \mathbf{I}(\neg \phi) | \Pi' \wedge \Pi''$

5.3.2 Solving the Conflict Set

Since many rules of behavior can be simultaneously activated, many plans π can be conflicting. Let us call $\Pi(\gamma, \Theta)$ the conflict set, i.e. the set of plans π that can be simultaneously activated with the set of desires γ and the belief $\Theta =< \sigma, V, C >$. $\Pi(\gamma, \beta)$ is a subset of $\mathcal{P}$. However, sometimes actions belonging to the conflict set $\Pi(\gamma, \Theta)$ are inconsistent. For instance, in the case of the above mentioned example 2, intentions lie and $\neg lie$ are conflicting, which means that they cannot be activated simultaneously.

To solve the conflict set, i.e. to find a consistent set of intentions, we exploit the logical structure of intentions according to Θ, and more precisely, to the set C of consequential rules belonging to Θ. For this, we define a semantics of consistent intentions (noted $\models_\Theta$) by reference to the semantics of goals defined in [19, 10]:

Definition 4 *: Let $\alpha \subseteq \Pi(\gamma, \beta) \subseteq \mathcal{P}$. The semantics of intentions is defined by as follows:*

$\alpha \models_\Theta \top$
$\alpha \models_\Theta \boldsymbol{I}\phi \Leftrightarrow \exists \phi' \in \alpha : \phi' \models_c \phi[\Theta]$
$\alpha \models_\Theta \boldsymbol{I}\Pi \Leftrightarrow \alpha \models_\Theta \Pi$
$\alpha \models_\Theta \neg \Pi \Leftrightarrow \alpha \not\models_c \Pi[\Theta]$
$\alpha \models_\Theta \Pi \wedge \Pi' \Leftrightarrow \alpha \models_c \Pi[\Theta] \wedge \alpha \models_c \Pi'[\Theta]$
$\alpha \models_\Theta \bot \Leftrightarrow \exists \phi : \alpha \models_\Theta \phi \wedge \alpha \models_\Theta \neg \phi$

We must now choose one of the maximal non conflicting subsets of $\Pi(\gamma, \Theta)$ which are defined as follows:

Definition 5 *: Let $\alpha \subseteq \Pi(\gamma, \beta)$. α is a maximal non conflicting subset of $\Pi(\gamma, \Theta)$ with respect to $\models_\Theta$ if and only if $\alpha \not\models_\Theta \bot$ and $\forall \alpha' \subseteq \Pi(\gamma, \Theta)$, $\alpha \subset \alpha' \Rightarrow \alpha' \models_\Theta \bot$*

Let us illustrate this operation on the two previous examples. In the case of *Example 1*, the set of actions $\mathcal{P}$ is composed of three atomic actions, $eat\_with\_fingers$, eat_crab and $hurt\_host$, and their negation, which is trivially inconsistent with respect to $\models_\Theta$, which means that $\mathcal{P} \models_\Theta \bot$. Therefore, there are two maximal consistent subsets of $\mathcal{P}$ that are $\{eat\_with\_fingers, eat\_crab\}$ and $\{\neg eat\_with\_fingers, \neg eat\_crab, hurt\_host\}$.

Let us now consider the *Example 2*: $\mathcal{P} = \{lie, \neg lie, someone\_murdered\}$. This set is inconsistent with respect to $\models_\Theta$. Therefore, there are two maximal consistent subsets of $\mathcal{P}$ that are $\{lie\}$ and $\{\neg lie, someone\_murdered\}$.

5.4 Solving Ethically the Conflict Set

In the previous section, we have explained how it was possible to solve the conflict set, i.e. to find the different maximal consistent subset of $\Pi(\gamma, \Theta)$. However, we have not yet taken into account the ethical values expressed by V in $\Theta =< \sigma, V, C >$. That is what we are doing now by choosing the *optimal maximal subsets* of $\Pi(\gamma, \Theta)$ with respect to the ordering relation $\succeq_c$ that expresses ethical priority.

Definition 6 *: Let Ω be the set of maximal non conflicting subset of $\Pi(\gamma, \Theta)$. $\alpha \in \Omega$ is an optimal maximal non conflicting subset of $\Pi(\gamma, \Theta)$ if and only if $\nexists \alpha' \in \Omega$ such that $\alpha \succ_c \alpha'$ ($\succ_c$ being defined in Definition 3)*

6 A Few Examples

To illustrate these notions and how they allow the construction of artificial ethical agents, let us consider again the previous examples and some of their possible refinements.

Coming back first to *Example 1*, we have to choose between the two maximal consistent subsets of actions that are $\{eat\_with\_fingers, eat\_crab\}$ and $\{\neg eat\_with\_fingers, hurt\_host, \neg eat\_crab\}$. Using the only value belonging to Θ, which is $eat\_with\_fingers \succ \neg eat\_with\_fingers$, it appears clearly that $\{eat\_with\_fingers, eat\_crab\} \succ_c \{\neg eat\_with\_fingers, hurt\_host, \neg eat\_crab\}$, which means, quite surprisingly, that the solution would be not to eat crab...

Let us now add a second value to Θ that specifies that $hurt\_host \succ eat\_with\_fingers$. It then appears that

$\{eat\_with\_fingers,\ eat\_crab\} \prec_c \{\neg eat\_with\_fingers,\ hurt\_host,\ \neg eat\_crab\}$ which means that the suitable action is to eat crab with fingers. Consider *Example 1'* that follows:

Example 1'

Rule 1': "Don't renounce to your personal engagement" that corresponds to the value $renounce\_engagement \succ \neg renounce\_engagement$
Rule 2: "if your host serves crab, you should eat crab", which can be formalized with the same rules as previously.
Rule 3': "if you eat crab, you have to renounce to your personal engagement (because you decided, may be for religious reasons, not to eat crab)." that is $eat\_crab \rightarrow renounce\_engagement$

The structure of *Example 1'* is identical to the structure of *Example 1*, where the proposition $renounce\_engagement$ replaces $eat\_with\_fingers$. As a consequence, the maximal subsets of $\mathcal{P}$ = { $renounce\_engagement$, $\neg renounce\_engagement$, eat_crab, $\neg eat\_crab$, $hurt\_host$ } that are consistent according to $\models_\Theta$ are the same, i.e. $\{renounce\_engagement,\ eat\_crab\}$ and $\{\neg renounce\_engagement, \neg eat\_crab, hurt\_host\}$. The only difference is that the value $hurt\_host \succ eat\_with\_fingers$ is replaced by $hurt\_host \prec renounce\_engagement$. Therefore, the optimal solution, with respect to Θ is $\{\neg renounce\_engagement, \neg eat\_crab, hurt\_host\}$.

Now, examine *Example 2*, without any value except that $lie \succ \neg lie$, which means that lying is bad. The optimal subset among the two maximal consistent subsets of $\mathcal{P} = \{lie, \neg lie, someone\_murdered\}$ that are $\{lie\}$ and $\{\neg lie, someone\_murdered\}$ is obviously $\{\neg lie, someone\_murdered\}$. If we add the value $someone\_murdered \succ \neg someone\_murdered$, which just tells that murder is bad, we obtain two possible consistent solutions among which it is not possible to discriminate, $\{lie\}$ and $\{\neg lie, someone\_murdered\}$. Lastly, with the value $someone\_murdered \succ lie$, only a subset succeeds: $\{lie\}$.

7 Comparison with Other Works

7.1 Relation with Deontic Logic

The preceding two examples and their variations show the efficiency of the method. The proposed formalization makes it possible to express ethical values under the form of binary relation and consequential rules, and to solve ethical conflicts. As many authors have said [32, 20, 17], most the the classical deontic logics, especially the said *Standard Deontic Logic* [36], which have been designed to represent obligations, laws and normative reasoning, fail to solve conflicts of norms. The above mentioned paradoxes, e.g. the *Chisholm's Paradox* [7] or the *paradox of the gentle murderer* [12] illustrate those difficulties.

Different solutions have been proposed. Some introduce priorities between norms [16]. Some other have proposed to introduce defeasible norms [21] or more generally non-monotonic logics. Here, we have proposed another solution that is to introduce priorities among intentions and consequences of activated intentions. As we have shown, there may be different solutions that are all consistent with the order $\succ_c$ induced by the context $\Theta =< \sigma, V, C >$. Each of the solutions is in accordance with the set of values V. In that way, it corresponds to an expression of the norms compatible with the values expressed in V. We want now to show that, independently of the set of values V, each maximal subset of $\mathcal{P}$ that is consistent with $\models_\Theta$ constitutes a system of norms that is compatible with the SDL. To do this, we replace the operator **O** that characterizes the deontic necessity by the operator **I** that stands for intention in the main theorems that are derived from the axioms of the *Standard Deontic Logic* [6]. Note that this assimilation of intentions to obligations could appears to be surprising to philosophers or logicians. However, we claim that it is totally justified in case of ethical robots, which intentions have to be considered as obligatory.

More precisely, by referring to the classical denomination of those theorems and by taking into account the semantics of the intentions (see Definition 4) the transcription that replaces the obligation **O** by the intention **I** gives:

D: $\neg\mathbf{I}(\bot)$
M: $\mathbf{I}(\Pi \wedge \Pi') \rightarrow (\mathbf{I}(\Pi) \wedge \mathbf{I}(\Pi'))$
C: $(\mathbf{I}(\Pi) \wedge \mathbf{I}(\Pi')) \rightarrow \mathbf{I}(\Pi \wedge \Pi')$
R: $\Pi \rightarrow \mathbf{I}(\Pi)$

Propositions For each α that is a *maximal non conflicting subset* (see Definition 4) of a non empty set of intentions $\Pi(\gamma, \Theta)$, we have: *P0* $\alpha \models_\Theta$ **R**, *P1* $\alpha \models_\Theta$ **D**, *P2* $\alpha \models_\Theta$ **M**, *P4* $\alpha \models_\Theta$ **C**. Below are sketches of proofs.
Proof P0: this trivially derives from definition 4, which explicitly states that $\alpha \models_\Theta \mathbf{I}(\Pi) \Leftrightarrow \alpha \models_\Theta \Pi$. □
Proof P1: by definition 5, $\alpha \not\models_\Theta \bot$. According to definition 4, $\alpha \models_\Theta \mathbf{I}\Pi \Leftrightarrow \alpha \models_\Theta \Pi$. Therefore $\alpha \not\models_\Theta \mathbf{I}(\bot)$, which gives $\alpha \models_\Theta \neg\mathbf{I}(\bot)$. □
Proof P2: from $\alpha \models_\Theta \mathbf{I}(\Pi \wedge \Pi')$ it follows $\alpha \models_c \Pi \wedge \Pi'[\Theta]$ (cf. Def. 4). This means $\alpha \models_c \Pi[\Theta] \wedge \alpha \models_c \Pi'[\Theta]$. As a consequence, $\alpha \models_\Theta \mathbf{I}(\Pi)$ and $\alpha \models_\Theta \mathbf{I}(\Pi')$, which means $\alpha \models_\Theta \mathbf{I}(\Pi) \wedge \mathbf{I}(\Pi')$ □
Proof P3: from $\alpha \models_\Theta \mathbf{I}\Pi \wedge \mathbf{I}\Pi'$ it follows $\alpha \models_\Theta \mathbf{I}\Pi$ and $\alpha \models_\Theta \mathbf{I}\Pi'$. According to Def. 4, it is equivalent to $\alpha \models_\Theta \Pi$ and $\alpha \models_\Theta \Pi'$ and consequently to $\alpha \models_\Theta \Pi \wedge \Pi'$, which, always according to Def. 4, gives $\alpha \models_\Theta \mathbf{I}(\Pi \wedge \Pi')$. □

7.2 Relation with Prioritized Logics

The comparison with deontic logics does not take into account the "optimality" of the maximal consistent subsets of $\Pi(\gamma, \Theta)$, which means that the values V of Θ are not considered in the semantic of intention. Those values are taken into account when computing the *optimal maximal non conflicting subsets* of $\Pi(\gamma, \Theta)$, which is defined in def. 6. The solutions are the least maximal consistent subsets according to the ordering $\succ_c$. At first sight, this approach looks very close to logics with priorities developed for instance in [4, 30, 31]. However, a close look at our examples shows that the problem cannot be easily solved with only priorities. More precisely, some of the approaches associate priorities to rules. However, in the case of Example 2, this wouldn't be relevant, because there doesn't exist any rule that allows to lie; so it's not possible to define a priority for this rule. Another solution would be to associate priorities to propositions and to choose the subsets of which priorities are the highest. However, in our case, it's far more complex, because if priorities correspond to the goodness of an action, it doesn't help, since the chosen solution is that one with the least highest priority. For instance, in example 2, the two sets of consequences $\{lie\}$ and $\{\neg lie, someone\_murdered\}$ cannot be compared in terms of priorities, because $someone\_murdered \succ lie \succ \neg lie$.

8 Conclusion

After having discussed what it means for an artificial agent to be moral, this paper introduces a consequentialist model of ethics for artificial agents based on the notion of BDI agent that is extended by adding moral values and consequences. We show how our approach is related to deontic logic. We also show that it is not reducible to existing prioritized logics. Apart this formal part, this paper constitutes an attempt to clarify what it means to be moral for artificial cognitive agents. The adopted approach, which is based on a consequentialist ethics, is very general. As the ethicists say, it covers many different ethical conceptions depending on the nature of the consequences: the consequences for the self, which corresponds to *hedonism*, or for the others, which corresponds to altruism, the intended consequences or the effective consequences, etc. All those different ethical conceptions can be formalized within the proposed framework. It can also be modified to model the utilitarianism, where the intentions are associated to numerical values, which depend on the pleasure or the pain that they are supposed to cause. Those numerical values are then summed up and the actions of which consequences have the highest score are chosen. This form of consequentialism could also be simulated with BDI agents, but it requires to modify the way consequences and values are processed. Lastly there are other ethical conceptions, for instance the deontic approaches, the ethics of responsibility, the ethics of discussion, etc. which require to model an assembly of subjects by multi-agents architectures and epistemic logics, which can be investigated by our approach.

REFERENCES

[1] A. Aaby, 'Computational Ethics', Technical report, Walla Walla College, (2005).

[2] I. Asimov, *I, Robot*, Gnome Press, 1950.

[3] Michael E. Bratman, *Intention, Plans, and Practical Reason*, CSLI publications, 1999 [1987].

[4] G. Brewka, 'Reasoning about priorities in default logic', in *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA*, eds., B. Hayes-Roth and R. E. Korf, volume 2, pp. 940–945, Menlo Park, (July–August 1994). AAAI Press.

[5] S. Bringsjord, K. Arkoudas, and P. Bello, 'Toward a General Logicist Methodology for Engineering Ethically Correct Robots', Technical report, Rensselaer Polytechnic Institute (RPI), Troy NY 12180 USA, (2006).

[6] B.F. Chellas, *Modal Logic: An Introduction.*, Cambridge University Press, Cambridge, 1980.

[7] R. Chisholm, 'Contrary-to-duty imperatives and deontic logic.', *Analysis*, (24), 33–36, (1963).

[8] B. Constant, *Des réactions politiques*, Éditions Flammarion, 1988.

[9] *Ethical and Legal Aspects of Unmanned Systems Interviews*, ed., Gerhard Dabringer, Ethica Themen, Institut fur Religion und Frieden, 2011.

[10] F.S. de Boer, K.V. Hindriks, W. van der Hoek, and J.-J.Ch. Meyer, 'A verification framework for agent programming with declarative goals', *Journal of Applied Logic*, **5**(2), 277 – 302, (2007).

[11] Eric Allen Engle, 'The history of the general principle of proportionality: An overview', *Dartmouth Law Journal*, **10**, 1–11, (July 2009). Available at SSRN: http://ssrn.com/abstract=1431179.

[12] J. W. Forrester, 'Gentle murder, or the adverbial samaritan.', *Journal of Philosophy*, (81), 193–196, (1984).

[13] H. Gensler, *Formal Ethics*, Routledge, 1996.

[14] Lou Goble, 'A logic for deontic dilemmas', *J. Applied Logic*, **3**(3-4), 461–483, (2005).

[15] R. Gutman and Rieff D., *Crimes of War: What the Public Should Know.*, W. W. Norton & Company, New-York, NY, USA, 1999.

[16] J. Hansen, 'Deontic logics for prioritized imperatives', *Artificial Intelligence and Law*, (14), 1–34, (2006).

[17] J. Hansen, 'The paradoxes of deontic logic', *Theoria*, (72), 221–232, (2006).

[18] G. Harman, 'Moral particularism and transduction', *Philosophical Issues*, **15**, (2005).

[19] Koen V. Hindriks, Frank S. de Boer, Wiebe van der Hoek, and John-Jules Ch. Meyer, 'Agent programming with declarative goals', in *Proceedings of the 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages*, ATAL '00, pp. 228–243, London, UK, (2001). Springer-Verlag.

[20] J. Horty, 'Moral dilemmas and nonmonotonic logic', *Journal of Philosophical Logic*, (23), 35–65, (1994).

[21] J. Horty, *Defeasible Deontic Logic*, chapter Nonmonotonic foundations for deontic logic, 17–44, Kluwer Academic Publishers, 1997.

[22] I. Kant, 'On a putative right to lie from the love of mankind, in the metaphysics of morals', in *Paperback, Cambridge Texts in the History of Philosophy*, Cambridge University Press, (1996).

[23] I. Kant, 'Critique of practical reason', in *Paperback, Cambridge Texts in the History of Philosophy*, Cambridge University Press, (1997).

[24] J.-J. C. Meyer, F.P.M. Dignum, and R.J. Wieringa, 'The paradoxes of deontic logic revisited: a computer science perspective', Technical Report UU-CS-1994-38, Utrecht University, Department of Computer Science, Utrecht, Netherlands, (1994).

[25] A. Newell, 'The knowledge level', *Artificial Intelligence Journal*, **18**, 87–127, (1982).

[26] T. Powers, 'Deontological Machine Ethics', Technical report, American Association of Artificial Intelligence Fall Symposium 2005, Washington, D.C., (2005).

[27] Anand Rao and Michael Georgeff, 'Bdi agents: From theory to practice', *Proceedings of the first international conference on multiagent systems ICMAS95*, **95**(Technical Note 56), 312–319, (1995).

[28] Anand S. Rao and Michael P. Georgeff, 'Modeling rational agents within a BDI-architecture', in *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, eds., James Allen, Richard Fikes, and Erik Sandewall, pp. 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, (1991).

[29] Stuart Russel and Peter Norvig, *Artificial Intelligence a Modern approach*, Series in Artificial Intelligence, Prentice Hall, 1995.

[30] C. Sakama and K. Inoue, 'Prioritized logic programming and its application to commonsense reasoning', *Artificial Intelligence*, (2000).

[31] T. Schaub and K. Wang, 'A semantic framework for preference handling in answer set programming.', *Theory and Practice of Logic Programming*, **3**(4), (2003).

[32] B. van Frassen, 'Values and the heart's command', *Journal of Philosophy*, (70), 5–19, (1973).

[33] Birna van Riemsdijk, Mehdi Dastani, and John-Jules Ch. Meyer, 'Goals in conflict: Semantic foundations of goals', *International Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, **18**(3), 471–500, (2009).

[34] P. Vayrynen, 'Particularism and default reasoning', *Ethical Theory and Moral Practice*, **7**, 53–79, (2004).

[35] P. Vayrynen, 'Moral generalism: Enjoy in moderation', *Ethics*, **116**, 707–741, (2006).

[36] G. H. vonWright, 'Deontic logics', *Mind*, (60), 1–15, (1951).

[37] Wendell Wallach and Colin Allen, *Moral Machines: Teaching Robots Right from Wrong.*, Oxford University Press, USA, November 2008.

Relevant Minimal Change in Belief Update

Laurent Perrussel[1] and **Jerusa Marchi**[2] and **Jean-Marc Thévenin**[3] and **Dongmo Zhang**[4]

Abstract.

The notion of *relevance* was introduced by Parikh in the belief revision field for handling minimal change. It prevents the loss of beliefs that do not have connections with the epistemic input. But, the problem of minimal change and relevance is still an open issue in belief update. In this paper, a new framework for handling minimal change and relevance in the context of belief update is introduced. This framework goes beyond relevance in Parikh's sense and enforces minimal change by first rewriting the Katzuno-Mendelzon postulates for belief update and second by introducing a new relevance postulate. We show that *relevant minimal change* can be characterized by setting agent's preferences on beliefs where preferences are indexed by subsets of models of the belief set. Each subset represents a prime implicant of the belief set and thus stresses the key propositional symbols for representing the belief set.

1 Introduction

Belief updating is the process of incorporating new pieces of information into a set of existing beliefs when the world described by this set has changed. It is usually assumed that this operation follows two principles: (i) the resulting belief set is consistent, and (ii) the change to the original belief set is minimal.

The most influential work within the area is the KM paradigm, which characterizes the belief update operation through a set of plausible axioms, generally referred to as the KM postulates [7]. Despite their popularity, the KM postulates are not sufficient to capture minimal change.

The notion of relevant belief was introduced by Parikh [13] in the context of belief revision. Relevant belief revision ensures that all beliefs in an initial belief set that are not related with the new piece of information are preserved. This notion avoids counter-intuitive changes of beliefs like those performed by the full meet revision operator [1], i.e. removing all statements from the original belief set and keeping only the new piece of information. Relevant change has been investigated in the belief revision context [10, 9, 14, 19]. However, relevance by its nature is a syntactical issue and model-based approaches provide only peripheral solutions. In this sense, approaches based on knowledge compilation [3] to prime implicates and prime implicants have been proposed [4, 15].

Particularly, in [15], we propose a relevant belief revision operator based on minimal change to general preference orderings via minimizing prime implicant changes to existing beliefs. This belief operator satisfies Katsuno-Mendelzon's postulates for belief revision as well as Parikh's postulate for relevant revision. However, that proposal was limited to the belief revision context.

The purpose of this paper is to extend our previous work in order to characterize the concept of *Relevant Belief Update*. Such characterization entails not only an adaptation of Parikh's postulate but also a new definition of the KM postulates for belief update in order to capture relevant minimal change. We consider that a belief update process should be performed over set of terms [16] instead of models by only looking at the literals that are concerned with the change issue. A natural way to focus on those literals is to represent the belief set as sets of prime implicants [11].

The paper is organized as follows. Section 2 reviews the notions of implicants and prime implicants and introduces some necessary definitions. Section 3 reviews the results obtained in [15], which are quite related to this work. Section 4 characterizes the class of *relevant minimal change* belief update operators in terms of postulates and constraints on preferences. Section 5 concludes the paper by considering some open issues.

2 Preliminaries

Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositional symbols and $\mathcal{L}$ be the propositional language associated with P. $\mathsf{Lang} : \mathcal{L} \mapsto 2^P$ is a function that assigns each formula φ in $\mathcal{L}$ the set of the propositional symbols occuring in φ.

Let $LIT = \{L_1, \ldots, L_{2n}\}$ be the set of associated literals: $L_i = p_j$ or $\neg p_j$. A *term* D_i is a *conjunction of literals*: $D_i = L_1 \wedge \cdots \wedge L_k$. Let $\overline{L_i}$ be the complementary literal, s.t. $\overline{L_i} = \neg p_j$ iff $L_i = p_j$ and $\overline{D}$ be the mirror of a term D s.t. $\overline{D} = \overline{L_1} \wedge \cdots \wedge \overline{L_k}$ iff $D = L_1 \wedge \cdots \wedge L_k$. In the following, terms can also be viewed as sets of literals ($D_i = \{L_1, \cdots, L_k\}$) and we will frequently switch between the two notations.

A term D is an *implicant* of an $\mathcal{L}$-formula ψ iff $D \models \psi$, where $\models$ is the satisfiability relation. A term D is said to be a *prime implicant* [17] of ψ if D is an implicant of ψ and for any term D' such that $D' \subset D$, we have $D' \not\models \psi$, i.e., a prime implicant of a formula ψ is an implicant of ψ without any subsumed terms.

Based on P and ψ, four specific sets of terms are considered:

1. $\mathcal{D}$ is the set of *all possible terms that can be built over* P. Since P is finite, $\mathcal{D}$ is also finite, because we only consider terms with non-redundant and non-contradicting literals;
2. PI_ψ is the set of *prime implicants of* ψ. This set is a disjunction of all non-contradictory and non-redundant prime implicants of ψ such that $\psi \equiv PI_\psi$. This set is unique and minimal in the sense that it consists of the smallest sets of terms closed for inference and without any subsumed terms;
3. $\mathcal{D}_\psi$ is the set of all *implicants of* ψ. This set is a disjunction of all non-contradictory and non-redundant implicants of ψ;

[1] Institut de Recherche en Informatique de Toulouse, Université Toulouse I, Toulouse, France, email: laurent.perrussel@irit.fr
[2] Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, Brazil, email: jerusa@inf.ufsc.br
[3] Université Toulouse I, Toulouse, France, email: thevenin@univ-tlse1.fr
[4] School of Computing and Mathematics, University of Western Sydney, Sydney, Australia, email: dongmo@scm.uws.edu.au

4. $\Gamma(\psi)$ is the set of *all possible terms based on* ψ defined as follows: for every $D_\psi \in PI_\psi$ and for every term $D \in \mathcal{D}$, a new term is obtained by adding to D all the literals of D_ψ which are non-conflicting with the literals of D. Formally:

$$\Gamma(\psi) = \{D \cup (D_\psi - \overline{D}) | D_\psi \in PI_\psi \text{ and } D \in \mathcal{D}\}$$

Figure 1 illustrates the inclusion relation between these sets: first prime implicants of ψ, then implicants of ψ, then terms that differ on some symbols with the implicants of ψ and finally all possible terms.

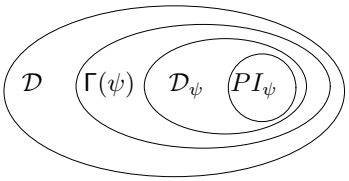


Figure 1. Inclusion relation of sets of terms.

In the sequel we omit "non-contradictory" and "non-redundant" when we mention prime implicants, implicants or terms.

Example 1. *Consider that* $P = \{p_1, p_2, p_3\}$ *is the set of propositional symbols and a formula* $\psi \in \mathcal{L}(P)$ *such that* $\psi = (p_1 \wedge p_2)$. *The following sets of terms can be obtained from* P *and* ψ*:*

$$\begin{array}{lcl} PI_\psi & = & \{\{p_1, p_2\}\} \\ \mathcal{D}_\psi & = & \{\{p_1, p_2\}, \{p_1, p_2, p_3\}, \{p_1, p_2, \neg p_3\}\} \\ \Gamma(\psi) & = & \{\{p_1, p_2\}, \{\neg p_1, p_2\}, \{p_1, \neg p_2\}, \\ & & \{\neg p_1, \neg p_2\}\{p_1, p_2, p_3\}, \{p_1, p_2, \neg p_3\}, \{\neg p_1, p_2, p_3\}, \\ & & \{\neg p_1, p_2, \neg p_3\}, \{p_1, \neg p_2, p_3\}, \{p_1, \neg p_2, \neg p_3\} \\ & & \{\neg p_1, \neg p_2, p_3\}, \{\neg p_1, \neg p_2, \neg p_3\}\} \\ \mathcal{D} & = & \{\{\}, \{p_1\}, \{p_2\}, \{p_3\}, \{\neg p_1\}, \{\neg p_2\}, \{\neg p_3\}, \\ & & \{p_1, p_2\}, \{p_1, \neg p_2\}, \{p_1, p_3\}, \cdots \{\neg p_2, \neg p_3\}, \\ & & \{p_1, p_2, p_3\}, \cdots, \{\neg p_1, \neg p_2, \neg p_3\}\} \end{array}$$

The cardinalities of these sets are: $| PI_\psi | = 1$, $| \mathcal{D}_\psi | = 3$, $| \Gamma(\psi) | = 12$ *and* $| \mathcal{D} | = 27$. *Let us stress that terms without the two propositional symbols involved in the prime implicants of* ψ *could not belong to* $\Gamma(\psi)$.

3 Relevance Criterion in Belief Change

In literature, *Belief Change* refers to two different but related theories: Belief Revision and Belief Update [1, 7]. Each of these activities is guided by a set of postulates that expresses some pre-requisites for belief change functions and describe how these functions should behave. In both theories, consistency maintenance and minimal change play a key role. However, Parikh observed that none of the theories follow the principle of minimal change. Ideally, if a statement φ in a belief base ψ does not share any propositional symbol with incoming information μ, then φ should belong to the resulting belief base after either the belief revision or belief update operation has been performed.

Formally, upon letting $\circ$ denote a belief revision operator, the following postulate captures the idea of relevant revision [13]:

(P) Let $\psi = \varphi \wedge \varphi'$ s.t. $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi') = \emptyset$. If $\mathsf{Lang}(\mu) \subseteq \mathsf{Lang}(\varphi)$, then $\psi \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi'$, where $\circ'$ is the revision operator restricted to language $\mathsf{Lang}(\varphi)$.

An open question, as stressed in [14], concerns the local revision operator mentioned in postulate **(P)**: this operator must be context-independent. Suppose there are two belief sets ψ and ψ' such that $\psi \equiv \varphi \wedge \varphi'$, $\psi' \equiv \varphi \wedge \varphi''$, $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi') = \emptyset$ and $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi'') = \emptyset$. Then only a single version of the local revision operator $\circ'$ should exist such that $\psi \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi'$ and $\psi' \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi''$ for any μ s.t. $\mathsf{Lang}(\mu) \subseteq \mathsf{Lang}(\varphi)$. Hereafter, we also commit to this *strong* version of **(P)**.

A relevant belief revision operator which minimizes the existing belief prime implicant change was proposed in [15]. That operator, denoted $\circ_{PI}$, satisfies Katsuno-Mendelzon's postulates for belief revision as well as Parikh's postulate for relevant revision. The first step in capturing the notion of relevance is to represent the belief base as its set of prime implicants. Prime implicants facilitate the splitting stage when performing the change by providing a canonical representation and the minimal language for representing belief base ψ.

Satisfaction of postulate (**P**) is assured then by the definition of faithful assignment, where preferences are defined within a subset of terms rather than on the whole set of possible models as required in [6, 8]. The pre-order is only required to be set over the set of terms that can be built from $\Gamma(\psi)$. Let $\leqslant_\psi$ be a preference relation defined over the set of all possible terms in $\Gamma(\psi)$: $D \leqslant_\psi D'$ states that D is at least as close as D' w.r.t. ψ. The notion of faithful assignment is defined as follows.

Definition 1. *[15] A* faithful assignment $\mathcal{F}_\psi$ *is a function which maps every formula* ψ *to a pre-order over* $\Gamma(\psi)$ *s.t.*:[5]

(C1-T) *if* $D, D' \in \mathcal{D}_\psi$, *then* $D \not<_\psi D'$.
(C2-T) *if* $D \in \mathcal{D}_\psi$ *and* $D' \notin \mathcal{D}_\psi$, *then* $D <_\psi D'$.
(C3-T) *if* $\psi \equiv \varphi$, *then* $\leqslant_\psi = \leqslant_\varphi$.
(C4-T) *For all* $D, D' \notin \mathcal{D}_\psi$, *if* $(D \subseteq D')$ *then* $D \sim_\psi D'$.

Constraint **(C4-T)** states that preferences should not favor too specific terms. This is the first step towards the enforcing the notion of relevance.

Operator $\circ_{PI}$ commits to the *strong* version of postulate **(P)** by setting a constraint on faithful assignment. Suppose that $\psi \equiv \varphi \wedge \varphi'$ such that $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$. Local revision operator $\circ'_{PI}$ used in (**P**) requires that there is only one pre-order $\leqslant_\varphi$ associated to φ. Suppose two terms D and $D' \in \Gamma(\varphi)$ such that $D \leqslant_\varphi D'$. Pre-order $\leqslant_\psi$ should also reflect these preferences; extending terms D and D' with any prime implicants belonging to $PI_{\varphi'}$ must not change preferences. The following constraint expresses the strong notion of relevance by considering multiple pre-orders.

(CS-T) Suppose φ and a faithful assignment $\mathcal{F}'_\psi$ s.t. $\mathcal{F}'_\psi(\varphi) = \leqslant_\varphi$. Faithful assignment $\mathcal{F}_\psi$ mapping each belief set ψ to a pre-order $\leqslant_\psi$ is said to be *relevant* iff for any φ, φ' s.t. $\psi \equiv \varphi \wedge \varphi'$ and $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$; for any $D, D' \in \Gamma(\varphi)$: $D \leqslant_\varphi D'$ iff $D \cup D_{\varphi'} \leqslant_\psi D' \cup D'_{\varphi'}$ s.t. $D_{\varphi'}, D'_{\varphi'} \in PI_{\varphi'}$ and $D \cup D_{\varphi'}, D' \cup D'_{\varphi'} \in \Gamma(\psi)$.

Revising a belief set ψ by μ is then defined as selecting the preferred terms w.r.t. $\leqslant_\psi$. It has been shown that the resulting revision operator $\circ_{PI}$ defined by faithful assignment $\mathcal{F}_\psi$ satisfies postulate **(P)** if faithful assignment $\mathcal{F}_\psi$ satisfies constraint **(CS-T)**:

[5] $D \sim_\psi D'$ stands for $D \leqslant_\psi D'$ and $D' \leqslant_\psi D$

Theorem 1. *[15] Let $\mathcal{F}'_\psi$ be a faithful assignment that maps each belief set ψ' to a total pre-order $\leqslant'_\psi$. Let $\circ'_{PI}$ be the revision operator defined by $\mathcal{F}'_\psi$. Let $\mathcal{F}_\psi$ be a faithful assignment that maps each belief set ψ to a total pre-order $\leqslant_\psi$. Let $\circ_{PI}$ be the revision operator defined by $\mathcal{F}_\psi$.*

*If $\mathcal{F}_\psi$ satisfies constraint **(CS-T)** w.r.t. $\mathcal{F}'_\psi$ then $\circ_{PI}$ satisfies **(P)** w.r.t. revision operator $\circ'_{PI}$.*

The result of relevance is rooted in two key aspects: defining the revision operator $\circ_{PI}$ and committing to the strong version of the relevance postulate. Hence, the prime implicant based revision operator exactly characterizes the notion of relevant belief revision.

Dalal's operator and Relevant Revision

According to [11], $\circ_{PI}$ revision operator is equivalent to Dalal's revision operator [2]. Considering that $\circ_{PI}$ is relevant, we show below that Dalal's revision operator is also relevant. The notion of distance used by Dalal can be rephrased with respect to distance between terms belonging to $\Gamma(\psi)$. Every term that belongs to $\Gamma(\psi)$ can be rewritten as $D \cup (D_\psi - \overline{D})$ s.t. $D_\psi \in PI_\psi$ and $D \in \mathcal{D}$. Hence, the set $D \cap \overline{D_\mu}$, where D_μ are the terms of the new information μ, represents the contradicting literals between the belief base ψ and the new information μ. We introduce function κ that returns the set of propositional symbols associated with this set of contradicting literals and which allows us to rephrase Dalal's pre-order $\leqslant^{\mathsf{Da}}_\psi$.

Definition 2 (κ). *Let $D_1 \in \mathcal{D}$, $D_\psi \in PI_\psi$ and $D \in \Gamma(\psi)$ s.t. $D = D_1 \cup (D_\psi - \overline{D_1})$: $\kappa(D) = \{p \in P | p \in (D_\psi \cap \overline{D_1})$ or $\neg p \in (D_\psi \cap \overline{D_1})\}$*

Definition 3 ($\leqslant^{\mathsf{Da}}_\psi$). *Let $D, D' \in \Gamma(\psi)$: $D \leqslant^{\mathsf{Da}}_\psi D' \iff |\kappa(D)| \leqslant_{\mathbb{N}} |\kappa(D')|$*

Let us state that Dalal's revision operator is relevant.

Proposition 1. *Let $\mathcal{F}^{\mathsf{Da}}_\psi$ be a function mapping a total pre-order $\leqslant^{\mathsf{Da}}_\psi$ to each belief set ψ. Function $\mathcal{F}^{\mathsf{Da}}_\psi$ is a faithful assignment which satisfies constraint **(CS-T)** w.r.t. faithful assignment $\mathcal{F}'_\psi = \mathcal{F}^{\mathsf{Da}}_\psi$.*

Proof. (sketch): It is straightforward to prove that **(C1-T)**–**(C3-T)** hold. Constraint **(C4-T)**: suppose $D, D' \notin \mathcal{D}_\psi$ s.t. $D \subseteq D'$; suppose l s.t. $l \in \kappa(D')$ and $l \notin \kappa(D)$: either (i) $D \cup \{l\}$ is not consistent and thus $D \cup \{l\} \notin \Gamma(\psi)$ or (ii) l is consistent with D and thus $D \cup \{l\} \in \Gamma(\psi)$ then $\kappa(D) = \kappa(D \cup \{l\})$ and thus $\kappa(D) = \kappa(D')$. Hence **(C4-T)** holds. Constraint **(CS-T)**: suppose it does not hold. Then it follows that $\exists \varphi, \varphi'$ s.t. $\psi \equiv \varphi \wedge \varphi'$, $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ and $\exists D, D' \in \Gamma(\varphi)$ s.t. $D \leqslant_\varphi D'$ and $D \cup D_{\varphi'} \not\leqslant_\psi D' \cup D'_{\varphi'}$. Since $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ it follows that $D_{\varphi'}$ is consistent with D and D'; hence $\kappa(D) = \kappa(D \cup D_{\varphi'})$ and $\kappa(D') = \kappa(D' \cup D_{\varphi'})$ which contradicts $D \cup D_{\varphi'} \not\leqslant_\psi D' \cup D'_{\varphi'}$. Since $D \cup D_{\varphi'}, D' \cup D'_{\varphi'} \in \Gamma(\psi)$, **(CS-T)** holds.

4 Relevant Belief Update

In this section we present the KM framework and we present how KM postulates are changed in order to consider sets of terms. We also show that Forbus' operator is relevant in the sense of Parikh, but it is not minimal. We present how a relevant and minimal operator can be obtained considering terms instead of models and we demonstrate how to achieve *Relevant Minimal Change*.

4.1 KM's Framework of Belief Update

Belief update concerns consistently inserting a new piece of information μ into a belief set ψ. The update operator is usually denoted by $\diamond$ and the resulting belief set is denoted $\psi \diamond \mu$. The KM postulates provide an axiomatic characterization of belief update operators in the context of finite propositional beliefs [7]:

(U1) $\psi \diamond \mu$ implies μ.
(U2) If ψ implies μ then $\psi \diamond \mu$ is equivalent to ψ.
(U3) If both ψ and μ are satisfiable then $\psi \diamond \mu$ is also satisfiable.
(U4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$ then $\psi_1 \diamond \mu_1 \equiv \psi_2 \diamond \mu_2$.
(U5) $(\psi \diamond \mu) \wedge \varphi$ implies $\psi \diamond (\mu \wedge \varphi)$.
(U6) If $\psi \diamond \mu_1$ implies μ_2 and $\psi \diamond \mu_2$ implies μ_1 then $\psi \diamond \mu_1 \equiv \psi \diamond \mu_2$.
(U7) If ψ is complete then $(\psi \diamond \mu_1) \wedge (\psi \diamond \mu_2)$ implies $\psi \diamond (\mu_1 \vee \mu_2)$.
(U8) $(\psi_1 \vee \psi_2) \diamond \mu \equiv (\psi_1 \diamond \mu) \vee (\psi_2 \diamond \mu)$.

Updating ψ by μ consists of choosing the closest models of μ with respect to each model of ψ [8, 7]. Let $\preceq_w$ be a pre-order representing preferences defined over $\mathcal{W}$, where $\mathcal{W}$ is the set of all propositional interpretations defined over P. The closeness criterion: $w' \preceq_w w''$ states that w' is at least as close as w'' w.r.t. w. Faithful assignment represents preferences related to w, i.e, the most preferred model is w:[6]

Definition 4. *A faithful assignment $\mathcal{F}_w$ is a function that maps each interpretation w to a partial pre-order $\preceq_w$ s.t.:*

(C1) *for all $w' \in \mathcal{W}$ if $w \neq w'$ then $w \prec_w w'$*

Let $[\![\psi]\!]$ be the set of propositional interpretations that satisfy ψ, i.e., the models of ψ. Updating a belief set is then defined by selecting the preferred models of μ w.r.t. each $\preceq_w$.

Theorem 2. *[8] An update operator $\diamond$ satisfies* **(U1)**–**(U8)** *if and only if there exists a faithful assignment $\mathcal{F}_w$ that maps each interpretation w to a partial pre-order $\preceq_w$ s.t. $[\![\psi \diamond \mu]\!] = \bigcup_{w \in [\![\psi]\!]} \min([\![\mu]\!], \preceq_w)$.*

One of the simplest ways to set preferences is to consider the propositional symbols that may change. This has been proposed by Dalal in [2] and is applied to belief update in [18, 5]. It consists of characterizing a belief change operator as a function which changes the minimal number of propositional symbol truth values in each ψ model so that incoming information can be added without entailing inconsistency.

4.2 Relevance Criterion on Belief Update

Since Dalal's operator is a relevant belief revision operator, the immediate question becomes: is it also the case for Dalal's belief update counter-part, the Forbus' operator [5]?

To get the answer, we first need to rephrase the Parikh's postulate for belief update. A naive translation is:

(P-U) Let $\psi = \varphi \wedge \varphi'$ s.t. $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi') = \emptyset$. If $\mathsf{Lang}(\mu) = \mathsf{Lang}(\varphi)$, then $\psi \diamond \mu \equiv (\varphi \diamond' \mu) \wedge \varphi'$, where $\diamond'$ is the update operator restricted to language $\mathsf{Lang}(\varphi)$.

Let us consider one example that illustrates the relevance issue with Forbus' belief update operator.

[6] $\prec_w$ is defined from $\preceq_w$ as usual, i.e., $w' \prec_w w''$ iff $w' \preceq_w w''$ but not $w' \preceq_w w''$.

Example 2. *Consider belief base* $\psi = (p_2 \wedge p_3 \wedge p_5) \vee (p_4 \wedge p_5)$ *and new piece of information* $\mu = (p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)$. *Performing the update process using Forbus' operator means to calculating distances and preferences between models of* ψ *and* μ*:* $w' \preceq_w w''$ *iff* $|d(w,w')| \leqslant_{\mathbb{N}} |d(w,w'')|$, *where function* d *gives the set of propositional symbols that differ between* w *and* w'. *The resulting belief base is given by the models of* μ *that are the closest to each model of* ψ*:*

$$[\![\psi \diamond \mu]\!] = \{\{p_1,p_2,\neg p_3,p_4,p_5\},\{\neg p_1,\neg p_2,\neg p_3,p_4,p_5\} \{p_1,p_2,\neg p_3,\neg p_4,p_5\},\{\neg p_1,\neg p_2,\neg p_3,\neg p_4,p_5\}\}$$

that corresponds to the following implicants:

$$\psi \diamond \mu = (p_1 \wedge p_2 \wedge \neg p_3 \wedge p_5) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_5)$$

The belief base ψ *can be split into* φ *and* φ' *such that* $\mathsf{Lang}(\varphi) = \{p_1,p_2,p_3,p_4\}$ *and* $\mathsf{Lang}(\varphi') = \{p_5\}$, *such that* $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi') = \emptyset$. *Literal* p_5 *is preserved in the resulting belief base, and thus the update process performed using Forbus' operator seems relevant in the sense of Parikh.*

However, we face two caveats. First, it is not minimal: literal p_4 appears in one implicant of ψ but not in the representation of μ and p_4 is also concerned with the update operation (p_4 no longer explicitly appears in the resulting belief base); second, the constraint requiring equal languages is too strong if we want to perform update as suggested by the example.

Since each prime implicant of ψ stresses up the relevant literals for representing ψ, update should also focus on these relevant literals. It means that update should be performed by considering each prime implicant of ψ rather than considering each model of ψ.

To enforce this new way to update a belief set, we extend the definition of $\Gamma(\psi)$ so that we consider one term D and a formula μ such that every prime implicant of μ is extended with the maximal consistent part of term D:

$$\Gamma(D,\mu) = \{D_\mu \cup (D - \overline{D_\mu})| \text{ and } D_\mu \in PI_\mu\}$$

Hence, the "relevant minimal change" operator should pick up terms $D_{(\psi,\mu)}$ of set $\bigcup_{D_\psi \in PI_\psi} \Gamma(D_\psi,\mu)$ that are the closest to each prime implicant D_ψ in PI_ψ as illustrated in Figure 2.

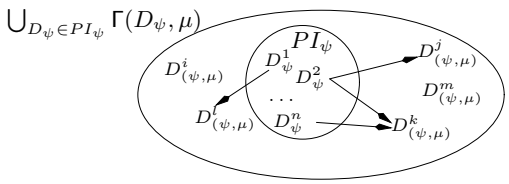


Figure 2. Belief update performed over terms of PI_ψ.

4.3 Belief Update in Prime Implicants

Our aim is to develop a theorem similar to theorems 1 and 2 that describes belief update operation in terms of preferences over terms. As explained, preferences are now indexed by the prime implicants D_ψ of ψ rather than by the models of ψ. First we rewrite constraint (**C1**) that characterizes preferences overs models: all terms which are entailed by an implicant D are strictly preferred to the terms that are not entailed by D. Secondly, we rewrite constraint (**C4-T**), which avoids setting preferences that favor too specific terms (cf. section 3). These two constraints characterize the notion of faithful assignment defined over terms.

Definition 5. *A* faithful assignment $\mathcal{F}_D$ *is a function which maps every* $D \in \mathcal{D}$ *to a partial pre-order* $\leqslant_D$ *defined over* $\Gamma(D)$ *s.t.:*

(CU1-T) *For all* $D', D'' \in \mathcal{D}$, *if* $D' \in \Gamma(D)$, $D'' \notin \Gamma(D)$ *then* $D' <_D D''$
(C4U-T) *For all* $D', D'' \in \Gamma(D)$, *if* $D' \subseteq D''$ *then* $D' \sim_D D''$.

Since preferences are indexed by terms instead of models, postulates **(U1)–(U8)** that characterize the notion of update by applying change to each model of the initial belief set have to be reformulated in order to accommodate the notion of change that leads to relevance: changes should be applied to each *prime implicant* of the initial belief set. In fact, all postulates are identical to postulates **(U1)–(U8)** except postulates **(U7-T)** and **(U9-T)**. Let $\psi \diamond_{PI} \mu$ denote the updated belief base. The following postulates characterize $\diamond_{PI}$:

(U1-T) $\psi \diamond_{PI} \mu$ implies μ.
(U2-T) If ψ implies μ then $\psi \diamond_{PI} \mu$ is equivalent to ψ.
(U3-T) If both ψ and μ are satisfiable then $\psi \diamond_{PI} \mu$ is also satisfiable.
(U4-T) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$ then $\psi_1 \diamond_{PI} \mu_1 \equiv \psi_2 \diamond_{PI} \mu_2$.
(U5-T) $(\psi \diamond_{PI} \mu) \wedge \varphi$ implies $\psi \diamond_{PI} (\mu \wedge \varphi)$.
(U6-T) If $\psi \diamond_{PI} \mu_1$ implies μ_2 and $\psi \diamond_{PI} \mu_2$ implies μ_1 then $\psi \diamond_{PI} \mu_1 \equiv \psi \diamond_{PI} \mu_2$.
(U7-T) If $PI_\psi = \{D_\psi\}$ then $(\psi \diamond_{PI} \mu_1) \wedge (\psi \diamond_{PI} \mu_2)$ implies $\psi \diamond_{PI} (\mu_1 \vee \mu_2)$.
(U8-T) $(\psi_1 \vee \psi_2) \diamond_{PI} \mu \equiv (\psi_1 \diamond_{PI} \mu) \vee (\psi_2 \diamond_{PI} \mu)$.
(U9-T) If $PI_\psi = \{D_\psi\}$ and $PI_\mu = \{D_\mu\}$ then $\psi \diamond_{PI} \mu = D_\mu \cup (D_\psi - \overline{D_\mu})$.

Postulate **(U7-T)** rephrases the condition "ψ is complete" as "ψ is represented by only one prime implicant". Combining **(U8-T)** and **(U7-T)** leads to update ψ by considering each prime implicant alternately. Postulate **(U9-T)** stresses up the second key difference between $\diamond$ and $\diamond_{PI}$: the result given by $\diamond_{PI}$ is a subset of the set $\bigcup_{D_\psi \in PI_\psi} \Gamma(D_\psi,\mu)$ while the result given by $\diamond$ is a subset of $\mathcal{W}$. Following [7], we now show that whenever constraints **(CU1-T)** and (**C4U-T**) hold, the nine update postulates are satisfied:

Theorem 3 (Update operator)**.** *Let* $\mathcal{F}_D$ *be a faithful assignment that maps each term* $D \in \mathcal{D}$ *to a partial pre-order* $\leqslant_D$. *PI update operator* $\diamond_{PI}$ *defined by* $\mathcal{F}_D$ *satisfies* **(U1-T)–(U9-T)** *if*

$$\psi \diamond_{PI} \mu =_{def} \bigcup_{D_\psi \in PI_\psi} \min(\Gamma(D_\psi,\mu), \leqslant_{D_\psi})$$

Proof. (Sketch) The proof is almost a direct translation of the proof of theorem 2 given in [7] and theorem 5 in [11]. (**U1-T**)–(**U4-T**) and (**U8-T**) are consequences of the definitions of Γ and $\diamond_{PI}$. Constraint (**C4U-T**) enforces postulates (**U5-T**)–(**U7-T**). Let us focus on (**U5-T**): suppose the case where ψ and $(\psi \diamond_{PI} \mu) \wedge \varphi$ are consistent, then there exists $D \in (\psi \diamond_{PI} \mu)$ and D_φ s.t. $D \wedge D_\varphi$ is consistent. It follows that there exists D_ψ s.t. D is minimal w.r.t. $\leqslant_{D_\psi}$. Constraint (**C4U-T**) entails that $D \cup D_\varphi$ is also minimal. Hence (**U5-T**) holds. Postulate (**U9-T**) holds since the results given by $\diamond_{PI}$ is a subset of $\bigcup_{D_\psi \in PI_\psi} \Gamma(D_\psi,\mu)$. □

4.4 Relevant Belief Update

Postulates (**U7-T**) and (**U9-T**) represent the first key step to handling *Relevant Minimal Change*. The second step is to rewrite Parikh's postulate in the context of belief update. Relevance has to be set by constraining faithful assignments. Consider a term D which can be split in a conjunction of two terms which do not share any symbols: $D \equiv D_1 \wedge D_2$. Suppose one pre-order $\leqslant_{D_1}$ defined by faithful assignment $\mathcal{F}'_D$. Now, suppose two terms $D, D' \in \Gamma(D_1, \mu)$ such that $D \leqslant_{D_1} D'$. Relevance states that adding D_2 to D and D' should not switch the preferences about D and D' since D_2 is expressed with symbols that differ from the symbols of D_1; that is $D \cup D_2 \leqslant_D D' \cup D_2$ (provided that $D \cup D_2$ and $D' \cup D_2$ are consistent, i.e. they belong to $\Gamma(D)$).

(CUS-T) Suppose D_1 and a faithful assignment $\mathcal{F}'_D$ s.t. $\mathcal{F}'_D(D_1) = \leqslant_{D_1}$. Faithful assignment $\mathcal{F}_D$ mapping each $D \in \mathcal{D}$ to a pre-order $\leqslant_D$ is *relevant* iff for any D, D_2 s.t. $D \equiv D_1 \wedge D_2$ and $\mathsf{Lang}(D_1) \cap \mathsf{Lang}(D_2) = \emptyset$; for any $D', D'' \in \Gamma(D_1)$: $D' \leqslant_{D_1} D''$ iff $D' \cup D_2 \leqslant_D D'' \cup D_2$ s.t. $D' \cup D_2, D'' \cup D_2 \in \Gamma(D)$.

Now, we show that operator $\diamond_{PI}$ characterizes relevant belief update by satisfying postulate based on **(P)**. The constraint (**CUS-T**) stating relevance by considering multiple assignments stresses that changes should be performed by handling implicants. Hence, the postulate for relevance should explicitly mention operator $\diamond_{PI}$ in its definition. We rephrase Parikh's postulate in terms of the prime implicant representation of belief since it enables the clear separation of relevant and non-relevant literals used to represent ψ:[7]

(PU-T) Let $PI_\psi = PI_\varphi \times PI_{\varphi'}$. If (i) $\mathsf{Lang}(PI_\mu) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ and (ii) $\forall \varphi'', \varphi'''$ s.t. $PI_\psi = PI_{\varphi''} \times PI_{\varphi'''}$ and $\mathsf{Lang}(PI_\mu) \cap \mathsf{Lang}(PI_{\varphi'''}) = \emptyset$, $\mathsf{Lang}(PI_{\varphi'''}) \subseteq \mathsf{Lang}(PI_{\varphi'})$; then $\psi \diamond_{PI} \mu \equiv (\varphi \diamond'_{PI} \mu) \wedge \varphi'$, where $\diamond'_{PI}$ is the PI update operator restricted to the language $\mathsf{Lang}(PI_\varphi)$.

The definition of the constraint states that if there exist φ and φ' s.t. $\psi = \varphi \wedge \varphi'$ and φ' is the formula that has the largest set of symbols (condition (ii)) which are not shared with those of μ (condition (i)), then $\diamond_{PI}$ should not change φ'. Hence, the postulate no longer requires equality between the languages of μ and φ as initially stated by Parikh.

If a faithful assignment satisfies constraint **(CUS-T)**, then operator $\diamond_{PI}$ satisfies the relevance postulate for update.

Theorem 4. *Suppose PI update operator $\diamond'_{PI}$ defined by the faithful assignment $\mathcal{F}'_D$. Let $\mathcal{F}_D$ be a faithful assignment that maps each $D \in \mathcal{D}$ to a partial pre-order $\leqslant_D$. PI update operator $\diamond_{PI}$ defined by $\mathcal{F}_D$ satisfies **(PU-T)**, w.r.t. operator $\diamond'_{PI}$, if $\mathcal{F}_D$ satisfies **(CUS-T)** w.r.t. faithful assignment $\mathcal{F}'_D$.*

Proof. (sketch) If it is not the case, there exist φ and φ' s.t. $PI_\psi = PI_{\varphi \wedge \varphi'}$, $\mathsf{Lang}(PI_\mu) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ and $\psi \diamond_{PI} \mu \not\equiv (\varphi \diamond'_{PI} \mu) \wedge \varphi'$. Suppose that $\psi \diamond_{PI} \mu \not\models (\varphi \diamond'_{PI} \mu) \wedge \varphi'$. It entails, because of the definition of $\diamond_{PI}$, that there exist D_ψ and $D \in \min(\Gamma(D_\psi, \mu), \leqslant_\psi)$ s.t. $D \not\models (\varphi \diamond'_{PI} \mu) \wedge \varphi'$. There also exist $D' \in \Gamma(\varphi)$ and $D_{\varphi'} \in PI_{\varphi'}$ s.t. $D = D' \cup D_{\varphi'}$ because of the definition of Γ and $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$. Condition **(CUS-T)** entails that D' belongs to $\min(\Gamma(D_\varphi, \mu), \leqslant_\varphi)$ and thus $D \models (\varphi \diamond'_{PI} \mu) \wedge \varphi'$. Contradiction. Proof for the case $(\varphi \diamond'_{PI} \mu) \wedge \varphi' \Rightarrow \psi \diamond_{PI} \mu$ is similar. □

[7] $PI_\varphi \times PI_{\varphi'}$ is the Cartesian product of sets PI_φ and $PI_{\varphi'}$.

Let us look at the opposite way: suppose an update operator $\diamond_{PI}$ which satisfies postulates **(U1-T)–(U9-T)** and **(PU-T)**; the question becomes "is there a relevant faithful assignment that can produce the same result?" If the answer is positive then it means that in fact operator $\diamond_{PI}$ characterizes the family of belief update operators that produces minimal relevant change. The following theorem shows that it is in fact the case if we focus on the strong meaning of relevance:

Theorem 5. *Suppose PI update operator $\diamond'_{PI}$ s.t. **(U1-T)–(U9-T)**; Suppose PI update operator $\diamond_{PI}$ s.t. **(U1-T)–(U9-T)** and **(PU-T)** hold w.r.t. PI update operator $\diamond'_{PI}$. Then (i) there exists a faithful assignment $\mathcal{F}'_D$ that maps every $D \in \mathcal{D}$ to a pre-order $\leqslant'_D$ s.t.*

$$\psi \diamond'_{PI} \mu =_{def} \bigcup_{D_\psi \in PI_\psi} \min(\Gamma(D_\psi, \mu), \leqslant'_{D_\psi})$$

*and (ii) there exists a relevant faithful assignment $\mathcal{F}_\mathcal{D}$ satisfying constraint **(CUS-T)** w.r.t. to faithful assignment $\mathcal{F}'_D$ s.t.*

$$\psi \diamond_{PI} \mu =_{def} \bigcup_{D_\psi \in PI_\psi} \min(\Gamma(D_\psi, \mu), \leqslant_{D_\psi})$$

Proof. (Sketch) Suppose $\psi \diamond_{PI} \mu$ s.t. postulates (**U1-T**)–(**U9-T**) and (**PU-T**) hold; let us define preferences of faithful assignment $\mathcal{F}_D$ as follows: for any terms D, D' and $D'' \in \mathcal{D}$, there exist D_1 and $D_2 \in \mathcal{D}$ s.t. $D' = D_1 \cup (D - \overline{D_1})$ and $D'' = D_2 \cup (D - \overline{D_2})$. We set $D' \leqslant_D D''$ iff $D \subseteq D'$ or $D \diamond_{PI} (D_1 \vee D_2) = \{D'\}$. Reflexivity and transitivity are proven as in [11]. (**CU1-T**) holds because : (i) for all terms D' subsumed by D, it holds that $D' \leqslant_D D''$ and (ii) (**U2-T**) entails that $D'' \not\leqslant_D D'$ for all $D'' \not\subseteq D$. Constraint (**C4-T**) holds because of postulate (**U5-T**) and also (**U7-T**). Finally, (**U5-T**), (**U7-T**)–(**U9-T**) entails that $D_\psi \diamond_{PI} \mu = \min(\Gamma(D_\psi, \mu), \leqslant_{D_\psi})$ which then entails that $\psi \diamond_{PI} \mu = \cup_{D_\psi \in PI_\psi} \min(\Gamma(\psi, \mu), \leqslant_{D_\psi})$. Finally, we prove that constraint (**CUS-T**) holds: suppose $\mathcal{F}'_D$ is defined in a similar way to $\mathcal{F}_D$ and based on $\diamond'_{PI}$. For all $D \in \mathcal{D}$, assume $D \equiv D_3 \wedge D_4$ and let us go back to the way we set preferences: $D \diamond_{PI} (D_1 \vee D_2) = \{D'\}$ and by (**PU-T**), it holds that $D' \equiv D_3 \diamond'_{PI} (D_1 \vee D_2) \wedge D_4$. Consequently $D_3 \diamond'_{PI} (D_1 \vee D_2) \equiv D' - D_4$. Moreover D' is minimal and also $D' - D_4$ (see above). Hence (**CUS-T**) holds. □

We conclude the characterization of $\diamond_{PI}$ by showing that Forbus-based PI update is minimal and relevant:

Proposition 2. *Let $\mathcal{F}_D^{\mathsf{Fo}}$ be a function mapping a pre-order $\leqslant_D^{\mathsf{Da}}$ to each $D \in \mathcal{D}$ (cf. Def. 2 and 3). Function $\mathcal{F}_D^{\mathsf{Fo}}$ is a faithful assignment which satisfies **(CUS-T)** w.r.t. faithful assignment $\mathcal{F}'_D = \mathcal{F}_D^{\mathsf{Fo}}$.*

The proof is similar to the proof of Proposition 1. Let us illustrate the proposition by reconsidering Example 2:

Example 3. *Consider belief base ψ and new piece of information μ as presented in Example 2 and represented as $PI_\psi = (p_2 \wedge p_3 \wedge p_5) \vee (p_4 \wedge p_5)$ and $PI_\mu = (p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)$. Definitions 2 and 3 give the following faithful assignment $\mathcal{F}_D^{\mathsf{Fo}}$ with pre-orders $\leqslant_{D_\psi}^{\mathsf{Da}}$:*

$$\begin{array}{lll} \{p_1, p_2, \neg p_3, p_5\} & <^{\mathsf{Da}}_{\{p_2, p_3, p_5\}} & \{\neg p_1, \neg p_2, \neg p_3, p_5\} \\ \{p_1, p_2, \neg p_3, p_4, p_5\} & \leqslant^{\mathsf{Da}}_{\{p_4, p_5\}} & \{\neg p_1, \neg p_2, \neg p_3, p_4, p_5\} \end{array}$$

Let $\diamond_{PI}^{\mathsf{Fo}}$ be the PI update operator defined by $\mathcal{F}_D^{\mathsf{Fo}}$. We get:

$$\begin{array}{rl} \psi \diamond_{PI}^{\mathsf{Fo}} \mu = & (p_1 \wedge p_2 \wedge \neg p_3 \wedge p_5) \vee \\ & (p_1 \wedge p_2 \wedge \neg p_3 \wedge p_4 \wedge p_5) \vee \\ & (\neg p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4 \wedge p_5) \end{array}$$

As expected, operator $\diamond_{PI}^{\mathsf{Fo}}$ preserves literals of prime implicant $(p_4 \wedge p_5)$.

5 Conclusion

This paper proposed a framework for handling relevant minimal update. We go beyond Parikh to ensure that literals without relation with new information are preserved. Operator $\diamond_{PI}$ is characterized both in terms of postulates and faithful assignment over terms. Performing belief update over terms, i.e, set of models ensures the syntax independence principle. Besides that, since beliefs are represented as sets of prime implicants, the belief update operator $\diamond_{PI}$ is not computationally more complex when applied in a relevant belief update process. In fact, Theorems 3–5 stress that $\diamond_{PI}$ exactly characterizes update operators that produce *relevant minimal change*.

There is a subtle link between relevance belief update and the frame problem [12]. On the one hand, these two problems are closely related. A solution to the frame problem requires separating irrelevant fluents from relevant fluents. If we know which fluents we should update after performing an action, these fluents are relevant and the rest are irrelevant. This means that a solution to relevance updating is a solution to the frame problem. On the other hand, a solution to the frame problem needs to be attached to an action logic, which is normally a high-order logic, either dynamic logic or situation calculus. Prime implicants are not expressive enough to represent actions and their effects. How to apply the techniques we introduced in this paper to an action logic will be a promising research topic for the future.

REFERENCES

[1] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *J. of Symbolic Logic*, 50(2):510–530, 1985.

[2] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of AAAI'88*, pages 475–479, 1988.

[3] Adnan Darwiche and Pierre Marquis. A Knowledge Compilation Map. *Journal of Artificial Intelligence Research*, (17):229–264, 2002.

[4] F. Van de Putte. Prime implicates and relevant belief revision. *Journal of Logic and Computation*, 7:1–11, Nov. 2011. doi:10.1093/logcom/exr040.

[5] K. Forbus. Introducing actions into qualitative simulation. In *Proc. of IJCAI-89*, pages 1273–1278, 1989.

[6] A. Grove. Two Modellings for Theory Change. *J. of Philosophical Logic*, 17:157–170, 1988.

[7] H. Katsuno and A. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proc. of KR'91*, pages 387–394. 1991.

[8] H. Katsuno and A. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

[9] George Kourousias and David Makinson. Parallel interpolation, splitting, and relevance in belief change. *J. Symb. Log.*, 72(3):994–1002, 2007.

[10] D. Makinson and G. Kourousias. Respecting relevance in belief change. *Análisis Filosófico*, 26(1):53–61, May 2006.

[11] J. Marchi, G. Bittencourt, and L. Perrussel. Prime forms and minimal change in propositional belief bases. *Annals of Math. and AI*, 2010.

[12] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pages 463–502. 1969.

[13] R. Parikh. *Beliefs, belief revision, and splitting languages*, volume 2, pages 266–278. Center for the Study of Language and Information, Stanford, CA, USA, 1999.

[14] P. Peppas, S. Chopra, and N. Foo. Distance semantics for relevance-sensitive belief revision. In *Proc. of KR'04*, pages 319–328, 2004.

[15] L. Perrussel, J. Marchi, and D. Zhang. Characterizing relevant belief revision operators. In *Proc. of AI'2010*, volume 6464 of *LNCS*, pages 42–51, 2010.

[16] Laurent Perrussel, Jerusa Marchi, and Guilherme Bittencourt. Prime implicants and belief update. In *Proceedings of the Twenty-Second International FLAIRS Conference (2009)*, pages 577 – 598, 2009.

[17] W.V.O. Quine. On cores and prime implicants of truth functions. *American Mathematics Monthly*, 66:755–760, 1959.

[18] M. Winslett. Reasoning about action using a possible models approach. In *Proc. of AAAI'88*, pages 89–93, 1988.

[19] Maonian Wu, Dongmo Zhang, and Mingyi Zhang. Language splitting and relevance-based belief change in horn logic. In *AAAI*, 2011.

Minimality Postulates for Semantic Integration

Özgür L. Özçep[1]

Abstract. Though for a long time the set of classical belief revision belief postulates of Alchourrón, Gärdenfors and Makinson (AGM) was thought to incorporate a principle of minimality, according to which the outcome of revising a knowledge base (KB) by new information had to be minimally different from the original KB, it was realised that one had to add additional postulates, called relevance postulates, in order to exclude forgetful revision. In this paper, we investigate two minimality postulates for a particular semantic integration scenario in which conflicts are caused by ambiguous use of symbols: A relevance postulate which says that only conflict relevant information is allowed to be eliminated and a generalised inclusion postulate which limits the creativity of the operators. Both postulates exploit the (satisfiably) equivalent representation of a first order logic KB by its prime implicates, which are its most logical atoms. As an example for a revision based operator in a semantic integration scenario, the definition of reinterpretation operators is recapitulated and it is shown that these fulfil both postulates.

1 INTRODUCTION

Not long after the seminal papers of Alchourrón, Gärdenfors and Makinson (AGM) [1, 2] it was realised that belief-revision techniques could be fruitfully applied for ontology based semantic integration [23], in particular for different types of ontology change such as ontology evolution, ontology alignment, ontology merge, ontology debugging etc. [13]. Most of the work exploiting belief revision for ontology change [14, 21, 12, 29, 28, 27] follows the general two-way approach of classical belief revision of, on the one hand, defining axiomatic specifications in the form of postulates and, on the other hand, constructing operators that fulfil these postulates.

Postulates provide means to declaratively describe the properties that an (revision, merge, integration etc.) operator to be built in some application context or scenario should fulfil. Moreover, postulates allow for the comparison of different operators. In this paper, we will look at postulates that are intended to specify a minimal change of a knowledge base (or more concretely an ontology) and show that their is a class of operators (reinterpretation operators) fulfilling them.

The intended integration scenario of this paper for which the minimality postulates are going to be developed can be described as follows. A receiver agent holds an ontology which is formally described by a knowledge base (KB) in some expressive formal language like first order logic (FOL) or a fragment of it (like description logic). In particular, a KB is a finite set of sentences in FOL (or a fragment of it). He receives information from another agent, who owns a possibly but only minimally different ontology, and he wants to integrate the information into his ontology.

[1] Institute for Software Systems (STS), Hamburg University of Technology, Germany, email: oezguer.oezcep@tu-harburg.de

It is assumed that both the sender's KB and the receiver's KB are well developed ontologies over the same application domain (e.g., ontologies for an online library system in universities); further it is assumed that the terms used in the ontologies either denote the same individuals, concepts and relations or are strongly related. Nonetheless, there may be terms that are used in different (related) ways in between the sender's and the receiver's KB (ambiguous use of terms). Here we constrain the ambiguous use to terms that denote concepts or relations but not individuals. Think, e.g., of two ontologies for an online library system where the receiver uses the term $Article$ in order to denote publications either in proceedings or journals while the sender uses $Article$ in a narrower sense to stand only for publications in journals. The receiver is assumed to give priority to the sender's meanings of the symbols and so the integration result will contain the trigger (this is similar to classical belief revision and different from non-prioritised belief revision [17]) and trigger a change of the receiver's ontology to conserve consistency. But, as the ontology of the receiver is assumed to be well developed the receiver is interested in changing his ontology only minimally, i.e., he wants to delete sentences of his KB and add additional sentences to it only as much as needed.

In belief revision the theme of minimality is mainly dealt with within the context of relevance postulates [16, 26] which specify that only those sentences of the receiver's KB that are relevant for conflicts with the trigger are allowed to be eliminated. But also inclusion postulates [18] can be seen as contributions to a minimal-change specification as they limit the operators's "creativity" by prescribing an upper bound to the result. In this paper, we start from these postulates for classical belief revision, argue why they are not proper minimality specifications for the intended integration scenario and formulate radically adapted versions that exploit the fine grained structure of ontologies by the notion of prime implicates. This adaptation is needed for aligning the symbol-oriented conflict diagnosis of the integration scenario (ambiguous use of symbols causes the conflict) with the fact that conflicts show themselves on the level of sentences.

The work of this paper continues previous work on integration operators for the intended integration scenario [11, 24]. We show that the reinterpretation operators, which exploit the idea of reinterpreting symbol's of the receiver's ontology and relating them with bridging axioms [10], fulfil the adapted relevance and inclusion postulate.

The paper is structured as follows. After this introduction we gather the logical preliminaries in Sect. 2. Section 3 describes relevance and inclusion postulates for belief revision operators that are associated with minimal change and argues why they are not adequate specifications for the intended integration scenario. Sections 4 and 5 describe new relevance and inclusion postulates that fit to the intended integration scenario. The last section before the conclusion, Sect. 6, defines a class of integration operators that fulfil the new postulates. An extended ver-

sion of this paper containing full proofs can be found under the URL http://dl.dropbox.com/u/65078815/oezcep12relevanceExt.pdf or http://www.sts.tu-harburg.de/people/oezcep/papers/papers.html.

2 LOGICAL PRELIMINARIES

A first order logic (FOL) vocabulary $\mathcal{V}$ consists of constants, predicate symbols and function symbols. For a FOL formula or set of formulas X let $\mathcal{V}(X)$ be the set of non-logical symbols occurring in X. A *literal* is an atomic or a negated atomic formula. The notion of a FOL *structure or interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}) \in \mathrm{Int}(\mathcal{V})$ over a vocabulary $\mathcal{V}$ is defined as usual; $\Delta^{\mathcal{I}}$ is the *domain* and $\cdot^{\mathcal{I}}$ is the *denotation* function; the truth of a formula α in $\mathcal{I}$, denoted $\mathcal{I} \models \alpha$ or equivalently $\alpha^{\mathcal{I}} = \mathrm{true}$, is defined in the well known Tarskian style. Let P be a unary predicate symbol, $D \subseteq \Delta^{\mathcal{I}}$ and $\mathcal{I}$ an interpretation. The interpretation $\mathcal{I}_{[P \mapsto D]}$ is called a P-variant of $\mathcal{I}$; it has the same denotations as $\mathcal{I}$ for all non-logical symbols except P, which is interpreted by D. For other non-logical symbols the variant is defined similarly. FOL formulas without free variables are called *sentences*. The set of sentences containing only non-logical symbols in the vocabulary $\mathcal{V}$ are denoted $\mathrm{Sent}(\mathcal{V})$. The set of sentences in $\mathrm{Sent}(\mathcal{V})$ following from a set of sentences X (over a perhaps larger vocabulary) is denoted by $\mathrm{Cn}^{\mathcal{V}}(X)$. If two sets of FOL sentences X_1, X_2 are logically equivalent, we write $X_1 \equiv X_2$.

A non-logical symbol $s \in \mathcal{V}$ *properly occurs in a sentence* $\alpha \in \mathrm{Sent}(\mathcal{V})$ iff there are FOL interpretations $\mathcal{I}_1, \mathcal{I}_2 \in \mathrm{Int}(\mathcal{V})$, s.t.: $\mathcal{I}_1$ and $\mathcal{I}_2$ differ only in the denotation of s and $\alpha^{\mathcal{I}_1} \neq \alpha^{\mathcal{I}_2}$. Let $P \in \mathcal{V}$ be an n-ary predicate symbol in $\mathcal{V}$. It occurs *syntactically positive (negative)* in an FOL formula iff it occurs in the scope of an even (uneven) number of negations—assuming that only the propositional truth functions $\wedge, \vee, \neg$ are used. For $P \in \mathcal{V}(\alpha)$ we say that P *occurs semantically positive in sentence* α, $posOcc(P, \alpha)$ for short, iff: For all interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and for subsets $D_1, D_2 \subseteq (\Delta^{\mathcal{I}})^n$ of the n-ary cartesian product of $\Delta^{\mathcal{I}}$ one has: If $D_1 \subseteq D_2$ and $\mathcal{I}_{[P \mapsto D_1]} \models \alpha$, then also $\mathcal{I}_{[P \mapsto D_2]} \models \alpha$. P *occurs semantically negative in sentence* α, $negOcc(P, \alpha)$ for short, iff $posOcc(P, \neg\alpha)$. P occurs *mixed in* α, $mixOcc(P, \alpha)$ for short, iff it properly occurs in α but neither $posOcc(P, \alpha)$ nor $negOcc(P, \alpha)$. We write $posOccOrNot(P, \alpha)$ (resp. $negOccOrNot(P, \alpha)$) iff $posOcc(P, \alpha)$ (resp. $negOcc(P, \alpha)$) or P does not occur syntactically in α.

The reinterpretation operators described in Sect. 6 are based on the concept of *dual remainder sets* [8, 29, 24], which is similar to the concept of remainder sets [2] used in the classical paper of AGM [1] for the construction of partial-meet revision functions. Let $B \top \alpha$, *the dual remainder sets modulo* α, denote the set of inclusion maximal subsets X of B that are consistent with α, i.e., $X \in B \top \alpha$ iff $X \subseteq B$, $X \cup \{\alpha\}$ is consistent and for all $\bar{X} \subseteq B$ with $X \subset \bar{X}$ the set $\bar{X} \cup \{\alpha\}$ is not consistent. The notion of dual remainders is extended to arbitrary KBs B_1 as second argument by defining $B \top B_1$ as $B \top \bigwedge B_1$.

3 MINIMALITY IN BELIEF REVISION

In his paper on two dogmas of belief revision, Hans Rott [30] pointed out the long standing belief (dogma) that classical belief revision à la Alchourrón, Gärdenfors and Makinson (AGM) [1] obeys a principal of minimality, according to which a KB is allowed to be revised only minimally in the light of new information. The AGM postulates do not constrain the revision result in the main interesting case of conflict between KB and triggering information. In fact, the amnesic (forgetful) revision operator defined by $B * \alpha = \mathrm{Cn}(\alpha)$ fulfils all AGM postulates though it is clearly not minimal as it completely deletes the sentences of the knowledge base B.

The relevance postulates of Hansson [16] and of Parikh [26] are two different possibilities that remedy the unwanted property of amnesic revision. Relevance postulates specify that only those sentences of the initial KB are allowed to be eliminated that are potential candidates for the conflict of the KB and the trigger. These kind of postulates constrain the revision result by an approximation from below in the sense that they say which set of sentences X (namely those not relevant for the conflict) have to be in the (set of consequences of the) revision result: $X \subseteq \mathrm{Cn}(B * \alpha)$. Note that the AGM postulate called Expansion 2 (Exp 2) constrains the result only in the trivial case where the trigger does not contradict the belief set. (AGM formulated their postulates for logically closed KBs which they call *belief sets*.)

(Exp 2) If $\neg\alpha \notin B$, then $\mathrm{Cn}(B \cup \alpha) \subseteq B * \alpha$.

The relevance postulate of Hansson [16] is formulated for arbitrary, i.e. not necessarily logically closed, sets of sentences B called belief bases. The postulate says in words: If a sentence β of the belief base B is not contained in the revision result $B * \alpha$, then it would lead to an inconsistency if it were added to a consistent extension B' of the revision result.

(Rel-H) If $\beta \in B$ and $\beta \notin B * \alpha$, then there is a set B', such that:

- $B * \alpha \subseteq B' \subseteq B \cup \{\alpha\}$;
- B' is consistent;
- $B' \cup \{\beta\}$ is not consistent.

Though this postulate formulates a moderate relevance condition for belief base operators it is not an adequate postulate for the intended integration scenario. In this scenario, it is not individual sentences which cause a conflict but different uses of (concept or role or more generally predicate but not constant) symbols in the knowledge base B and the trigger β. And indeed, the reinterpretation based operators defined below do not fulfil this postulate.

Example 1 *Let, e.g., be given a knowledge base according to which we think that the media pr_1, pr_2, which are published in some proceedings are articles: $B = \{Article(pr_1), Article(pr_2)\}$. The trigger $\alpha = \neg Article(pr_1)$ stems from an agent who has a different understanding of 'article' according to which only publications in journals (but not proceedings) are articles. An appropriate revision result $B * \alpha$ would not only delete $Article(pr_1)$ but also $Article(pr_2)$; because the next time the sender sends a trigger containing $Article$ negatively, namely $\neg Article(pr_2)$, a conflict will occur. But this operator $*$ does not fulfil (Rel-H). As we will show below we can formulate a radically adapted version of this relevance postulate that is fulfilled by the reinterpretation operators.*

A completely different relevance postulate, which is more symbol-oriented and hence works equally for belief-base revision and belief-set revision, was formulated by Parikh [26] and further developed by him and colleagues [6, 7], as well as generalised by [20] and [19]. The idea rests on representing a KB B equivalently with KB components with pairwise disjoint symbols sets $\mathcal{V}_n$. Then a formula β is considered to be relevant for the revision with the trigger α iff β and α have symbols in one of the symbol sets $\mathcal{V}_n$ in common.

Formally, let $\mathcal{V}$ be an FOL vocabulary and $\mathbf{V} = \{\mathcal{V}_n\}_{n \in I}$ be a partition of $\mathcal{V}$. $\mathbf{V}$ is a *splitting* of a KB B iff there exists a family

of KBs $\{B_n\}_{n\in I}$ s.t.: $\mathcal{V}(B_n) \subseteq \mathcal{V}_n$ and $\bigcup\{B_n\}_{n\in I} \equiv B$ [20]. Ordering splittings as partitions in the usual way, one can prove that for every KB B there is always a unique finest splitting of B [20, 26]. Now let B be a consistent KB and $\mathbf{V} = \{\mathcal{V}_n\}_{n\in I}$ the unique finest splitting $\mathcal{V}$ of B. A formula β is *irrelevant* w.r.t. to the revision of B with trigger α—β is *irrelevant for α modulo B for short*—iff for all $\mathcal{V}_n \in \mathbf{V}$: $\mathcal{V}_n \cap \mathcal{V}(\beta) = \emptyset$ or $\mathcal{V}_n \cap \mathcal{V}(\alpha) = \emptyset$. The relevance criterion of Parikh now reads:

(Rel-P) If β is irrelevant for α modulo B, then $\beta \in \mathrm{Cn}(B * \alpha)$.

Parikh's criterion (Rel-P) is not strong enough to exclude a kind of semantic integration operation that in some sense is too sceptical.

Example 2 *Think again of an integration scenario where the sender has a stronger notion of article than the receiver. Assume that the receiver's KB is $B = \{Article(pr_1), Article(pr_2), \neg Article(bo_1)\}$, which in particular says that the publication bo_1 is not an article, and the trigger stemming form the sender is $\alpha = \neg Article(pr_1)$. Consider the following integration operator $*$: For arbitrary KBs B and trigger α the operator renames concept and role symbols s of the receiver's KB into new fresh symbols s' in order to regain consistency. In case of this example only the occurrences of $Article$ in B are renamed into $Article'$ and one gets $B * \alpha = \{Article'(pr_1), Article'(pr_2), \neg Article'(bo_1), \neg Article(pr_1)\}$. This operator $*$ clearly fulfils the criterion (Rel-P). But we lose the information of B that the book bo_1 is not an $Article$. Hence (Rel-P) is not a relevance criterion that prohibits all too sceptical (though symbol oriented) revision.*

The relevance postulates cover only one aspect of minimality, but completely miss the other aspect of minimality which is to constrain the (consequences of the) revision result from above. That is, one has to prescribe a set X such that $\mathrm{Cn}(B * \alpha) \subseteq X$. In classical AGM belief revision [1] the first expansion postulate (Exp 1) constrains the revision result only in the uninteresting case where α does not contradict B. In the more interesting case of contradiction, $\mathrm{Cn}(B \cup \alpha)$ is the set of all sentences, hence the postulate becomes vacuous.

(Exp 1) $B * \alpha \subseteq \mathrm{Cn}(B \cup \alpha)$.

For belief base revision, the (revised) knowledge base and the result do not have to be logically closed. Hence the postulate corresponding to (Exp 1), called inclusion postulate, really results in a approximation from above—thereby hampering all too creative base revision.

(Incl) B$*\alpha \subseteq B \cup \alpha$.

But for the integration scenario, belief base revision is not the means of choice as its results depend on the syntactic representation of the belief bases. For example, if the result of the revision of $\{Article(pr_1), \neg Article(bo_1), \neg Article(bo_2)\}$ with $\neg Article(pr_1)$ results in $\{\neg Article(bo_1) \wedge \neg Article(bo_2)\}$, this should be considered to be a non-creative (acceptable) revision result, though (Incl) is not fulfilled. Hence, we will define a different form of inclusion postulate that abstracts from the syntactic representations of the knowledge bases. Thereby we will have described a postulate for operators on the knowledge level [22], in which ontologies are first-class citizens.

4 THE POSTULATE OF REINTERPRETATION RELEVANCE

For the following we will assume that B is a predicate logical KB without the identity and function symbols, i.e., B is a finite set of sentences in first order (predicate) logic without identity and functional symbols. The new relevance postulate starts off from Hansson's relevance postulate (Rel-H) and adapts it in the direction of making it more symbol-oriented. The main technical tool for the adaptation is the concept of a prime implicate, which roughly represents a most atomic component of the KB. Though it is the different use of symbols that leads to conflicts in our integration scenario, it is sentences that make up a conflict. Hence, by representing a KB in a specific normal form by its implied prime implicates, one gets a fine-grained means for identifying the real culprit symbols for conflicts: just identify the prime implicates in which the symbols are contained and which are involved in a conflict. While the notion of prime implicate is omnipresent for propositional logic [3] and has been exploited for the definitions of propositional revision operators [5, 25, 31], there is no real semantic notion of prime implicate for FOL that deserves this term (but compare the prime implicate definition for modal logics in [4]), and there is no approach that uses prime implicates in the postulates. We will work with a more syntactic notion of prime implicates and use it for the (satisfiably) equivalent representations of KBs.

The core idea of the new relevance theorem is this: A sentence β entailed by B is allowed to be eliminated from the integration result if there is a related sentence ϵ of the normal form of B that together with other formulas of the normal form leads to a contradiction. The kind of relatedness between β and ϵ will be further specified below. We now formalise the notions in order to formulate the relevance postulate.

A FOL formula α is universal iff α is equivalent to a formula in prenex form containing only all-quantifiers $\forall$. A universal formula of the form $\forall x_1 \ldots \forall x_n(li_1 \vee \cdots \vee li_m)$, where the li_j are literals with variables in $\{x_1, \ldots, x_n\}$, is a *FOL clause*. An FOL clause $\alpha_1 = \forall x_1 \ldots \forall x_n \beta$ is a *(proper) subclause* of a FOL clause α_2, iff α_2 is of the form $\alpha_2 = \forall y_1 \ldots \forall y_n \delta$, where all x_i are among the y_j and the set of literals in β is a (proper) subset of the literals in δ.

Let X be a set of universal formulas. The set of *FOL clauses of X w.r.t. to a vocabulary $\mathcal{V}$*, $\mathrm{Cl}^{\mathcal{V}}(X)$, is the set of all FOL clauses in $\mathrm{Sent}(\mathcal{V})$ entailed by X. For formulas α let $\mathrm{Cl}^{\mathcal{V}}(\alpha) = \mathrm{Cl}^{\mathcal{V}}(\{\alpha\})$. If X is an arbitrary set of FOL sentences, let X^* be the result of skolemizing every sentence in X (with fresh constants). Let $\mathcal{V}_{\mathrm{sk}}$ be the set of used skolem symbols. The set of *FOL clause of X w.r.t. $\mathcal{V}$ and skolem symbols $\mathcal{V}_{\mathrm{sk}}$* is defined by $\mathrm{Cl}^{\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}}}(X^*)$.

Now we can define the set of *FOL prime implicates* of a set of universal formula X w.r.t. $\mathcal{V}$ as the set of non-tautological clauses of X for which there is no proper subclause in $\mathrm{Cl}^{\mathcal{V}}(X)$.

$$\mathrm{PI}^{\mathcal{V}}(X) \quad = \quad \{pr \in \mathrm{Cl}^{\mathcal{V}}(X) \mid pr \text{ is non-tautological and has no proper subclauses in } \mathrm{Cl}^{\mathcal{V}}(X)\}$$

The notion of an FOL prime implicate leads to a logically equivalent characterisation of sets X containing only universal formulas.

Proposition 1 *Let $\mathcal{V}$ be a predicate logical vocabulary. For every set X of universal formulas X with $\mathcal{V}(X) \subseteq \mathcal{V}$ we have: $X \equiv \mathrm{PI}^{\mathcal{V}}(X)$.*

The notion of relatedness mentioned above is explicated technically by the (semantically) positive and negative occurrences of symbols; it says that β and ϵ are related w.r.t. to a symbol P occurring in both iff P occurs in the same polarity in both sentences or at least mixed in one of the sentences.

Definition 1 *Let P be a predicate symbol which occurs properly in β and ϵ. β and ϵ are called* related w.rt. P *iff a) either $mixOcc(P, \epsilon)$ or $mixOcc(P, \beta)$; or b) $posOcc(P, \epsilon)$ and $posOcc(P, \beta)$; or c) $negOcc(P, \epsilon)$ and $negOcc(P, \beta)$.*

The new relevance postulate (Rel-R) which we call the postulate of *reinterpretation relevance* now has the following form:

(Rel-R) Let be given a vocabulary $\mathcal{V}$, an FOL KB B over $\mathcal{V}$, an FOL sentence α over $\mathcal{V}$ and an FOL clause β over $\mathcal{V}$. Let B^* be a skolemization of $\bigwedge B$ with skolem constants from $\mathcal{V}_{\text{sk}}$.
If $B \models \beta$ and $B * \alpha \not\models \beta$, then there is a set X and a sentence $\epsilon \in X$ s.t.:

1. $X \subseteq \text{PI}^{\mathcal{V} \cup \mathcal{V}_{\text{sk}}}(B^*)$;
2. $X \cup \{\alpha\}$ is inconsistent;
3. $(X \setminus \{\epsilon\}) \cup \{\alpha\}$ is consistent and
4. ϵ is related with β w.r.t. a predicate symbol P.

In words the postulates says the following: If there is a sentence β (in fact we constrain β to be a clause) which follows from the original KB B but is not contained in the integration result $B * \alpha$, then there must be a good reason for excluding it from the result. The reason for the exclusion is explained by a reference to the prime implicate form of B: There is a sentence ϵ related to β such that its addition to a subset $X \setminus \{\epsilon\}$ of the prime implicate form contradicts α. Hence, the exclusion is not necessarily justified by identifying β as a culprit for the conflict but (possibly) another related sentence ϵ. Note that the set $X \setminus \{\epsilon\}$ has the role of B' in the relevance postulate (Rel-H) of Hansson. Though (Rel-R) expresses a very weak form of relevance, it prohibits revision operators like those of Ex. 2.

Example 3 *As in Ex. 2 assume that the KB has the form* $B = \{Article(pr_1), Article(pr_2), \neg Article(bo_1)\}$ *and the trigger is* $\alpha = \neg Article(pr_1)$. *Clearly* $\text{PI}^{\mathcal{V} \cup \mathcal{V}_{\text{sk}}}(B^*) = B^* = B$. *Let* $\beta = \neg Article(bo_1)$. *Then* $B \models \beta$ *and* $B * \alpha \not\models \beta$. *But for the predicate* $Article$ *there is no* $X \subseteq \text{PI}^{\mathcal{V} \cup \mathcal{V}_{\text{sk}}}(B^*)$ *that fulfils the conditions of (Rel-R) because the only* β*-related prime implicate is* $\neg Article(bo_1)$ *which is not involved in a conflict.*

5 AN EXTENDED INCLUSION POSTULATE

We further exploit the idea of prime implicates to define a postulate that captures the other aspect of minimal integration where one constrains the (consequences of the integration) result from above. The idea, in principle, is to enrich the given KB B to an equivalent set $\text{Enr}(B)$ that contains enough consequences of B in order to identify the real potential culprits in the integration process. A typical example for an enrichment operator is the disjunctive closure of a belief base according to which a belief base is closed up with all (finite) disjunctions of sentences in it [15]. The general schema of the extended inclusion axiom is the following:

(Incl-ES) For all α there is an $X \subseteq \text{Enr}(B)$ such that $X \cup \{\alpha\} \not\models \bot$, and for all β: If $B * \alpha \models \beta$, then $X \cup \{\alpha\} \models \beta$.

This schema says: There is a subset of the enrichment of B such that all sentences β entailed by the integration result follow from a subset X of the enrichment together with the trigger α. Instantiations of this schema with different enrichment operators Enr result in different extended inclusion postulates whose usability relies heavily on the properties of Enr. If, e.g., Enr is the identity, we get an all too strict inclusion postulate. If Enr is Cn, we get an all too weak inclusion postulate. That means, the good candidates for Enr lie in between the identity and the consequence operator, and hence one should ensure that $\text{Enr}(B) \equiv B$. As we allow the enrichment also to introduce new symbols (like those needed for skolemization) we weaken this goodness criterion to the restriction that B and $\text{Enr}(B)$ should be equivalent w.r.t. to the old vocabulary $\mathcal{V}$. The enrichment operator Enr we will use in the following is an operator that enriches B with prime implicates of its skolemization.

$$\text{Enr}(B) = B \cup \text{PI}^{\mathcal{V} \cup \mathcal{V}_{\text{sk}}}(B^*)$$

And in fact, though the enriched KB $\text{Enr}(B)$ is not equivalent to B, it is at least equivalent w.r.t. to the non-skolem symbols.

Proposition 2 *For (finite) KBs* B *over* $\mathcal{V}$*:*

$$\text{Cn}^{\mathcal{V}}(B) = \text{Cn}^{\mathcal{V}}(\text{Enr}(B))$$

We call the postulate that results from (Incl-ES) by instantiating the parameter Enr by $\text{Enr}(B) = B \cup \text{PI}^{\mathcal{V} \cup \mathcal{V}_{\text{sk}}}(B^*)$ the extended inclusion postulate (Incl-E).

6 REINTERPRETATION OPERATORS

The extended relevance postulate and inclusion postulate are intended to specify minimal changes of revision-like operators which are used in a particular semantic integration scenario described in the introduction. In this section, we recapitulate the definition of operators of this kind [11, 24] and show that they fulfil the new postulates. Other postulates that are fulfilled by these operators (cf. ([24]) will not be discussed in this paper. The construction of the operators mimics the construction of the propositional revision operators of [9].

The integration operator to be defined in the following is denoted by $\circ$ and is called a reinterpretation operator. (In [24] it is called weak reinterpretation operator of type 2, but as we define only this reinterpretation operator, here we do not use the additional specifications.) $\circ$ is a binary operator with a finite FOL KB as left and an FOL sentence α as right argument. Before giving the technical definition, the main construction idea will be illustrated with the KB and the trigger of Ex. 2.

Example 4 *Let* $B = \{Article(pr_1), Article(pr_2), \neg Article(bo_1)\}$ *and the trigger* $\alpha = \neg Article(pr_1)$. *The reinterpretation operator* $\circ$ *results in the following KB:*

$$\begin{aligned} B \circ \alpha \quad = \quad & \{Article'(pr_1), Article'(pr_2), \neg Article'(bo_1), \\ & \neg Article(pr_1), \\ & \forall x(Article(x) \rightarrow Article'(x))\} \end{aligned}$$

The conflict between B *and* α *is traced back to ambiguous use of symbols. As we assume that only predicate symbols (and not constants) may be used ambiguously, the conflict can only be caused by different uses of the unary predicate* $Article$. *The receiver (holder of* B*) gives priority to the sender's use of* $Article$ *over his use of* $Article$, *and hence he adds* $\neg Article(pr_1)$ *into the result* $B \circ \alpha$. *Its own use of* $Article$ *is internalized, i.e., all occurrences of* $Article$ *in* B *are substituted by a new symbol* $Article'$. *This step of internalization will also be called the step of dissociation or disambiguation as the uses of* $Article$ *according to sender and receiver are put apart. But as we assumed that in the integration scenario the uses of* $Article$ *by sender and receiver are similar, the receiver adds hypotheses on the semantical relatedness (bridging axioms, cf. [23]) of his and the sender's use of* $Article$. *The hypothesis in this case is* $\forall x(Article(x) \rightarrow Article'(x))$ *which says that articles in the sender's sense are also articles in the receiver's sense. Note that because of this hypothesis the result* $B \circ \alpha$ *entails the assertion* $\neg Article(bo_1)$ *from the initial KB* B. *The other direction of the hypothesis, namely* $\forall x(Article'(x) \rightarrow Article(x))$ *cannot be added to the result as it would lead to a contradiction.*

So the general construction for the reinterpretation operators in case of conflict is first to disambiguate the symbols involved in a conflict and second add bridging axioms. Technically the disambiguation is realised by uniform substitutions called *ambiguity compliant resolution substitutions*, $\mathrm{AR}(\mathcal{V}, \mathcal{V}')$ for short. Here, we assume $\mathcal{V} \cap \mathcal{V}' = \emptyset$ where $\mathcal{V}'$ is the set of symbols used for internalization. The substitutions in $\mathrm{AR}(\mathcal{V}, \mathcal{V}')$ get as input a non-logical symbol in $\mathcal{V}$ (in case of this paper: a predicate symbol) and map it either to itself or to a new non-logical symbol (of the same type) in $\mathcal{V}'$. In our case we only consider the substitution of predicate symbols. The set of symbols $s \in \mathcal{V}$ for which $\sigma(s) \neq s$ is called the support of σ and is denoted $\mathrm{supp}(\sigma)$. A substitution with support S is also denoted by σ_S. For substitutions $\sigma_1, \sigma_2 \in \mathrm{AR}(\mathcal{V}, \mathcal{V}')$ we define an ordering by: $\sigma_1 \leq \sigma_2$ iff $\mathrm{supp}(\sigma_1) \subseteq \mathrm{supp}(\sigma_2)$. $\mathrm{AR}(\mathcal{V}, \mathcal{V}')$ can be partitioned into equivalence classes of substitutions that have the same support. We assume that for every equivalence class a representative substitution $\Phi(S) \in ars(\mathcal{V}, \mathcal{V}')$ with support S is fixed. Φ is called a *disambiguation schema*.

In the general case, there may be more than one predicate symbol which has to be disambiguated in order to get consistency; and even more, there may be many different sets of symbols for which a disambiguation leads to consistency. These sets are called minimal conflict symbol sets and are defined formally as follows:

Definition 2 *Let B be an FOL KB over $\mathcal{V}$ and α an FOL sentence over $\mathcal{V}$. The set of* minimal conflicting symbols sets*, $\mathrm{MCS}(B, \alpha)$, is defined by:*

$$\mathrm{MCS}(B,\alpha) = \{S \subseteq \mathcal{V} \quad | \quad \begin{array}{l} \textit{There is a } \sigma_S \in \mathrm{AR}(\mathcal{V},\mathcal{V}'), \textit{ s.t.} \\ B\sigma_S \cup \{\alpha\} \textit{ is consistent, and for} \\ \textit{all } \sigma_R \in \mathrm{AR}(\mathcal{V},\mathcal{V}') \textit{ with } \sigma_R < \sigma_S \\ B\sigma_R \cup \{\alpha\} \textit{ is not consistent.} \} \end{array}$$

As no symbol set in $\mathrm{MCS}(B, \alpha)$ is a better candidate then the other, we assume that a selection function γ_1 selects the good candidates: $\gamma_1(\mathrm{MCS}(B, \alpha)) \subseteq \mathrm{MCS}(B, \alpha)$. In the end, the symbol set $S^{\#} = \bigcup \gamma_1(\mathrm{MCS}(B, \alpha))$ is the set of symbols which will be internalized.

In the second step, the disambiguated symbols of $S^{\#}$ are related by bridging axioms. Depending on what kind of bridging axioms are chosen, different integration operators result. Here, we choose a very conservative simple class of initial bridging axioms called *simple bridging axioms*. (For other types of bridging axioms see [24] and [11].) Let be given a substitution $\sigma = \sigma_S \in \mathrm{AR}(\mathcal{V}, \mathcal{V})$ with support $S \subseteq \mathcal{V}$. Let P be an n-ary predicate symbol in S, $\sigma(P) = P'$ and let $\vec{x} = x_1, \ldots, x_n$. Then define $\overrightarrow{P} = \forall \vec{x}(P(\vec{x}) \rightarrow P'(\vec{x}))$ and $\overleftarrow{P} = \forall \vec{x}(P'(\vec{x}) \rightarrow P(\vec{x}))$.

Definition 3 *Let $\sigma = \sigma_S \in \mathrm{AR}(\mathcal{V}, \mathcal{V})$ for $S \subseteq \mathcal{V}$. The set of simple bridging axioms w.r.t. σ is* $\mathrm{BA}(\sigma) = \{\overrightarrow{P}, \overleftarrow{P} \mid P \in S\}$.

In case of conflict, not all bridging axioms of $\mathrm{BA}(S^{\#})$ can be added to the integration result (compare Ex. 4). Hence, we search for subsets that are compatible with the union of the internalized KB and the trigger, $B\sigma \cup \{\alpha\}$. That means, possible candidate sets of bridging axioms can be described by dual remainder sets (see section on logical preliminaries) as $\mathrm{BA}(\sigma)^{\top}(B\sigma \cup \{\alpha\})$. Again, as there is no preference for one candidate over the other we assume that a second selection function γ_2 is given with $\gamma_2(\mathrm{BA}(\sigma)^{\top}(B\sigma \cup \{\alpha\})) \subseteq (\mathrm{BA}(\sigma)^{\top}(B\sigma \cup \{\alpha\})$. The intersections of the selected bridging axioms is the set of bridging axioms added to the integration result. (Compare this with the partial meet revision functions of AGM [1]). The reinterpretation operator $\circ = \circ^{\overline{\gamma}}$ now is defined as follows:

Definition 4 *Let $\mathcal{V}$ be a predicate logical vocabulary, $\mathcal{V}'$ a disjoint predicate logical vocabulary (for internalization) and let be given a disambiguation scheme Φ. Moreover let be given selection functions γ_1, γ_2 and for short let $\overline{\gamma} = (\gamma_1, \gamma_2)$. For any FOL KB B and FOL sentence α over $\mathcal{V}$ let $S^{\#} = \bigcup \gamma_1(\mathrm{MCS}(B, \alpha))$ and $\sigma = \Phi(S^{\#})$. Then the* reinterpretation operator $\circ = \circ^{\overline{\gamma}}$ *is defined by*

$$B \circ \alpha \quad = \quad \sigma(B) \cup \{\alpha\} \cup \bigcap \gamma_2\big(\mathrm{BA}(\sigma)^{\top}(\sigma(B) \cup \{\alpha\})\big)$$

It can easily be checked that this definition of $\circ$ gives the results in Ex. 4 (for any pair of selection functions γ_1, γ_2).

7 REINTERPRETATION OPERATORS ARE MINIMAL SEMANTIC INTEGRATION OPERATORS

We now justify the introduction of the reinterpretation operators by proving that they fulfil the reinterpretation postulate and the extended inclusion postulate. The main component in the proofs are propositions that explicate the interaction of the internalization and of the bridging axioms with the prime implicates implied by the KB B. The first main proposition is explicated in the following:

Proposition 3 *Let be given vocabularies $\mathcal{V}$ and $\mathcal{V}'$ with $\mathcal{V} \cap \mathcal{V}' = \emptyset$. Let B be a set of universal formula in FOL (without identity and function symbols) over $\mathcal{V}$, let σ be a substitution of predicate symbols P by new symbols $\sigma(P) \in \mathcal{V}'$ and let* $\mathrm{PI}(\cdot) = \mathrm{PI}^{\mathcal{V} \cup \mathcal{V}'}(\cdot)$*. Then:*

$$\mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}(B\sigma)) = \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}(B\sigma) \cap \mathrm{Sent}(\mathcal{V}))$$

If a KB B is internalized w.r.t. to some symbols (those in the support of σ), then some of the original consequences of B are lost, and hence this is also true for the equivalent set of prime implicates of $B\sigma$ over the (larger) vocabulary $\mathcal{V} \cup \mathcal{V}'$. But remarkably, according to this proposition, if we restricted the prime implicates to those containing only symbols of $\mathcal{V}$, the loss of $\mathcal{V}$-consequences of B does not become bigger. That means that in order to register losses of $\mathcal{V}$-consequences of B we can stick to the prime implicates of B.

While this proposition hints to the interaction of prime implicates with the internalization, the following proposition talks about their interaction with simple bridging axioms. The proposition refers to the notion of an admissible skolemization. Let $B^* = \forall \tilde{x}_1 \ldots \forall \tilde{x}_m \tilde{B}$ be a skolemization of B with skolem constants not in $\mathcal{V}(B \cup B\sigma)$. Then $B^*\sigma = \forall \tilde{x}_1 \ldots \forall \tilde{x}_m \tilde{B}\sigma$ is a skolemization of $B\sigma$. Let $\forall z \tilde{ba}$ be a prenex form of some set of bridging axioms $ba \subseteq \mathrm{BA}(\sigma)$. Then $(B\sigma \cup ba)^*$ is called an B^**- admissible skolemization* of $B\sigma \cup ba$ iff it has the form $(B\sigma \cup ba)^* = \forall z \forall \tilde{x}_1 \ldots \forall \tilde{x}_m (\tilde{B\sigma} \wedge \tilde{ba})$.

Proposition 4 *Let $\mathcal{V}$, $\mathcal{V}'$, $\mathcal{V}_{\mathrm{sk}}$ be pairwise disjoint vocabularies. Let B be a KB over $\mathcal{V}$ and σ be a substitution of predicate symbols P by new predicate symbols $\sigma(P) \in \mathcal{V}'$. Let $ba \subseteq \mathrm{BA}(\sigma)$ be a subset of bridging axioms and $(B\sigma \cup ba)^*$ be a B^*-admissible skolemization of $B\sigma \cup ba$ with skolem constants from $\mathcal{V}_{\mathrm{sk}}$; then:*

$$\mathrm{PI}^{\mathcal{V} \cup \mathcal{V}' \cup \mathcal{V}_{\mathrm{sk}}}((B\sigma \cup ba)^*) \cap \mathrm{Sent}(\mathcal{V} \cup \mathcal{V}(B^*)) \subseteq \mathrm{PI}^{\mathcal{V} \cup \mathcal{V}_{\mathrm{sk}}}(B^*)$$

This proposition says that the internalization with symbols from $\mathcal{V}'$ and the addition of bridging axioms to the KB does not enlarge the capability of prime implicates to entail sentences not containing internal symbols. Again, that means that the original prime implicates of the KB B can be used as indicators for possible conflicts. Note that a corresponding proposition for more complex bridging axioms may not hold.

Using these propositions one can show the desired theorem.

Theorem 1 *The reinterpretation operators according to Definition 4 fulfil the postulates of reinterpretation relevance (Rel-R) and extended inclusion (Incl-ES).*

We give a proof of the theorem based on the propositions above and the following proposition which is part of the folklore.

Proposition 5 *Let β be an FOL formula over $\mathcal{V}$ and β^* be a skolemization with constants not in $\mathcal{V}$. Then $\mathrm{Cn}^{\mathcal{V}}(\beta) = \mathrm{Cn}^{\mathcal{V}}(\beta^*)$.*

Proof that postulate (Rel-R) is fulfilled

We need the following lemma which can be proved by considering resolution. In the lemma we use the auxiliary boolean function g; let P be an n-ary predicate symbol and β be an arbitrary sentence. $g(ba, \beta)$ holds iff P occurs in β in a polarity corresponding to its occurrence in the simple bridging axiom ba.

$$g(ba,\beta) = \begin{cases} posOccOrNot(P,\beta), & \text{if } ba = \overleftarrow{P} \\ negOccOrNot(P,\beta), & \text{if } ba = \overrightarrow{P} \end{cases}$$

Lemma 1 *Let $S = \{P_1, \ldots, P_n\}$ be a set of pairwise disjoint predicate symbols from a vocabulary $\mathcal{V}$, $\sigma = [P_1/P_1', \ldots, P_n/P_n']$ an injective substitution with $P_i' \in \mathcal{V} \setminus S$, $1 \leq i \leq n$. Let $\mathcal{V}_n = \mathcal{V} \setminus \{P_1', \ldots, P_n'\}$. Let B be a KB with $\mathcal{V}(B) \subseteq \mathcal{V}_n$ and $ba \subseteq \mathrm{BA}(\sigma)$ a set of bridging axioms of the form $ba(P_i) \in \{\overrightarrow{P_i}, \overleftarrow{P_i}\}$, $1 \leq i \leq n$. Let $(B\sigma \cup ba)^*$ be an B^*-admissible skolemization of $B\sigma \cup ba$ with skolem constants from $\mathcal{V} \setminus \mathcal{V}(B \setminus \mathcal{V}(B\sigma)$. Last but not least let $U \subseteq S$ be the set of symbols $P_i \in S$ such that $\{\overrightarrow{P_i}, \overleftarrow{P_i}\} \subseteq ba$. Then*

$$\begin{aligned} Cl^{\mathcal{V}_n}((O\sigma \cup B)^*) \quad = \quad & \{\beta \in Cl^{\mathcal{V}_n}(O^*) \mid \textit{There is an } \epsilon \textit{ with:} \\ & \epsilon \in Cl^{\mathcal{V}_n}((B\sigma \cup ba)^*); \\ & \epsilon \models \beta; \\ & \epsilon \textit{ has no symbol of } S \setminus \mathcal{V}(ba) \textit{ and for all} \\ & P_i \in (S \cap \mathcal{V}(ba)) \setminus U\textit{: } g(ba(P_i), \epsilon)\} \end{aligned}$$

Let B_{res} abbreviate $B \circ \alpha = B\sigma \cup ba \cup \{\alpha\}$ for a subset $ba \subseteq BA(\sigma)$. Let $B_{\mathrm{res}} \not\models \beta$ and $B \models \beta$. Because of Prop. 5 it holds that $B^* \models \beta$ and $(B\sigma \cup ba)^* \not\models \bot$. Because of Lemma 1 it follows that there is a predicate symbol P in β s.t.: P does not occur in ba or we have $ba(P) \in ba$, but not $g(ba(P), \beta)$. I consider only the latter case as the former can be reduced to it. W.lo.g. let $ba(P) = \overleftarrow{P}$. That means that P either occurs mixed in β or positively.

That $\overleftarrow{P}$ is not contained in the integration result means that there is a subset $ba' = \{ba(P_1), \ldots, ba(P_k)\} \subseteq \mathrm{BA}(\sigma)$ of the bridging axioms s.t.

$$Y := B\sigma \cup ba'$$

is compatible with α but

$$Z := B\sigma \cup ba' \cup \{\overleftarrow{P}\}$$

is not compatible with α. Hence $\neg\alpha \notin \mathrm{Cn}^{\mathcal{V}}(Y)$, whilst $\neg\alpha \in \mathrm{Cn}^{\mathcal{V}}(Z)$. Let Y^* and Z^* be B^*-admissible skolemizations with skolem constants in $\mathcal{V}_{\mathrm{sk}}$. With Prop. 5 it follows $\neg\alpha \notin \mathrm{Cn}^{\mathcal{V}}(Y^*)$, whilst $\neg\alpha \in \mathrm{Cn}^{\mathcal{V}}(Z^*)$. Because of Prop. 1 it follows

$$\neg\alpha \quad \notin \quad \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Y^*)), \text{ but} \tag{1}$$

$$\neg\alpha \quad \in \quad \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Z^*)) \tag{2}$$

Because of Prop. 3 we have further

$$\begin{aligned} \mathrm{Cn}^{\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}}}(Y^*) &= \mathrm{Cn}^{\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Y^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \\ \mathrm{Cn}^{\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}}}(Z^*) &= \mathrm{Cn}^{\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Z^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \end{aligned}$$

Intersecting both sides of the equation with $\mathrm{Sent}(\mathcal{V})$ results in the equations:

$$\begin{aligned} \mathrm{Cn}^{\mathcal{V}}(Y^*) &= \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Y^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \\ \mathrm{Cn}^{\mathcal{V}}(Z^*) &= \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Z^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \end{aligned}$$

Therefore with (1) and (2) one can infer that

$$\begin{aligned} \neg\alpha &\notin \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Y^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \\ \neg\alpha &\in \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Z^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})) \end{aligned}$$

Because of Prop. 4 the sets $X_1 = \mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Y^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})$ and $X_2 = \mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(Z^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}_{\mathrm{sk}})$ are prime implicates of B^* with $X_1 \subseteq X_2$. Choose an X such that $X_1 \subseteq X \subseteq X_2$ and X is inclusion minimal w.r.t. the property that $\neg\alpha \in \mathrm{Cn}^{\mathcal{V}}(X)$. Such an X exists, as for X_2 one has $\neg\alpha \in \mathrm{Cn}^{\mathcal{V}}(X_2)$. X must contain prime implicates in which P occurs positively or in mixed form; otherwise we would have $X_1 = X_2$. Hence there is also an ϵ which is related to β w.r.t. P. So all conditions of (Rel-R) are fulfilled.

Proof that postulate (Incl-E) is fulfilled

Assume that $B \circ \alpha = B\sigma \cup ba \cup \{\alpha\}$ for some set of bridging axioms $ba \subseteq \mathrm{BA}(\sigma)$. Now we have the following chain of equations:

$$\begin{aligned} \mathrm{Cn}^{\mathcal{V}}(B\sigma \cup ba) &\overset{\text{Prop. 5}}{=} \mathrm{Cn}^{\mathcal{V}}((B\sigma \cup ba)^*) \\ &\overset{\text{Prop. 1}}{=} \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}((B\sigma \cup ba)^*)) \\ &\overset{\text{Prop. 3}}{=} \mathrm{Cn}^{\mathcal{V}}(\mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}((B\sigma \cup ba)^*) \\ &\qquad \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}(B^*))) \end{aligned}$$

Let $X = \mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}((B\sigma \cup ba)^*) \cap \mathrm{Sent}(\mathcal{V}\cup\mathcal{V}(B^*))$. Then we continue with

$$X \overset{\text{Prop. 4}}{\subseteq} \mathrm{PI}^{\mathcal{V}\cup\mathcal{V}'\cup\mathcal{V}_{\mathrm{sk}}}(B^*) \subseteq \mathrm{Enr}(B)$$

Hence $X \cup \{\alpha\}$ is consistent, because $B \circ \alpha$ is consistent. For all β with $B \circ \alpha \models \beta$ and $\beta \in \mathrm{Sent}(\mathcal{V})$ it holds that $B\sigma \cup ba \models \alpha \to \beta$, hence $X \models \alpha \to \beta$ and so also $X \cup \{\alpha\} \models \beta$.

8 CONCLUSION

This paper investigated minimality postulates for a particular integration scenario where a receiver agents wants to integrate information stemming from a sender agent. We assumed that the understandings of the symbols by the sender and the receiver are in most cases identical; but if they are used in different meanings, they differ only minimally. Starting off from relevance postulates and inclusion postulates for belief revision operators we defined the postulate of reinterpretation relevance and the postulate of extended inclusion. These specify a global kind of minimal change of the receiver's KB by specifying what is allowed to be eliminated (conflict relevant sentences) from the result and what sentences at most are allowed to be contained in the result.

A novel feature of the postulates is the exploitation of prime implicates. The introduction of prime implicates makes it possible to align

one of the assumption for the intended semantic scenario (namely that it is ambiguous use of symbols which causes the conflict) with the fact that contradictions show themselves not on the symbol level but on the sentence level.

The reinterpretation operators recapitulated in this paper can be shown to fulfil the new postulates and as such can be thought of realising a semantic integration which changes the meanings of the receiver's symbols only in a minimal way.

Concerning future work we mention that the postulates (Rel-R) and (Incl-E) are intended to be used as main components for an envisioned representation theorem for predicate logical reinterpretation operators.

ACKNOWLEDGEMENTS

I would like to thank the referees for their comments, which helped improve this paper.

REFERENCES

[1] Carlos Eduardo Alchourrón, Peter Gärdenfors, and David Makinson, 'On the logic of theory change: partial meet contraction and revision functions', *Journal of Symbolic Logic*, **50**, 510–530, (1985).

[2] Carlos Eduardo Alchourrón and David Makinson, 'Hierarchies of regulations and their logic', in *New Studies in Deontic Logic*, ed., R. Hilpinen, 125–148, D. Reidel Publishing, (1981).

[3] Tania Armstrong, Kim Marriott, Peter Schachte, and Harald Søndergaard, 'Two classes of boolean functions for dependency analysis', *Science of Computer Programming*, **31**(1), 3–45, (1998).

[4] Meghyn Bienvenu, 'Prime implicates and prime implicants: From propositional to modal logic', *J. Artif. Intell. Res. (JAIR)*, **36**, 71–128, (2009).

[5] Meghyn Bienvenu, Andreas Herzig, and Guilin Qi, 'Prime implicate-based belief revision operators', in *ECAI*, eds., Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, volume 178, pp. 741–742. IOS Press, (2008).

[6] Samir Chopra, Konstantinos Georgatos, and Rohit Parikh, 'Relevance sensitive non-monotonic inference on belief sequences', *Journal of Applied Non-Classical Logics*, **11**(1–2), 131–150, (2001).

[7] Samir Chopra and Rohit Parikh, 'Relevance sensitive belief structures', in *Annals of Mathematics and Artificial Intelligence*, pp. 259–285, (2000).

[8] James P. Delgrande, 'Horn clause belief change: Contraction functions', in *Principles of Knowledge Representation and Reasoning: Proceedings of the 11th International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, eds., Gerhard Brewka and Jérôme Lang, pp. 156–165. AAAI Press, (2008).

[9] James P. Delgrande and Torsten Schaub, 'A consistency-based approach for belief change', *Artificial Intelligence*, **151**(1–2), 1–41, (2003).

[10] Dejing Dou, Drew V. McDermott, and Peishen Qi, 'Ontology translation by ontology merging and automated reasoning', in *Proceedings of the EKAW-02 Workshop on Ontologies for Multi-Agent Systems*, pp. 3–18, (2002).

[11] Carola Eschenbach and Özgür L. Özçep, 'Ontology revision based on reinterpretation', *Logic Journal of the IGPL*, **18**(4), 579–616, (2010). First published online August 12, 2009.

[12] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache, 'Inconsistencies, negations and changes in ontologies', in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, pp. 1295–1300, (2006).

[13] Giorgos Flouris, Dimitris Manakanatas, Haridimos Kondylakis, Dimitris Plexousakis, and Grigoris Antoniou, 'Ontology change: classification and survey', *The Knowledge Engineering Review*, **23**(2), 117–152, (2008).

[14] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou, 'On applying the AGM theory to DLs and OWL', in *Proceedings of the 4th International Semantic Web Conference (ISWC-05)*, pp. 216–231, (2005).

[15] Sven Ove Hansson, 'Changes of disjunctively closed bases', *Journal of Logic, Language and Information*, **2**, 255–284, (1993).

[16] Sven Ove Hansson, 'Reversing the Levi identity', *Journal of Philosophical Logic*, **22**, 637–669, (1993).

[17] Sven Ove Hansson, 'A survey of non-prioritized belief revision', *Erkenntnis*, **50**(2-3), 413–427, (1999).

[18] Sven Ove Hansson, *A Textbook of Belief Dynamics*, Kluwer Academic Publishers, 1999.

[19] David Makinson, 'Propositional relevance through letter-sharing: review and contribution', in *Formal Models of Belief Change in Rational Agents*, (2007).

[20] David Makinson and George Kourousias, 'Parallel interpolation, splitting, and relevance in belief change', *Journal of Symbolic Logic*, **72**, 994–1002, (September 2007).

[21] Thomas Meyer, Kevin Lee, and Richard Booth, 'Knowledge integration for description logics', in *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*, pp. 645–650, (2005).

[22] Allan Newell, 'The knowledge level', *Artificial Intelligence*, **18**, 87–127, (1982).

[23] Natalya Fridman Noy, 'Semantic integration: a survey of ontology-based approaches', *SIGMOD Record*, **33**(4), 65–70, (2004).

[24] Özgür L. Özçep, 'Towards principles for ontology integration', in *FOIS*, eds., Carola Eschenbach and Michael Grüninger, volume 183, pp. 137–150. IOS Press, (2008).

[25] Maurice Pagnucco, 'Knowledge compilation for belief change', in *AI 2006: Advances in Artificial Intelligence*, eds., Abdul Sattar and Byeong-ho Kang, volume 4304 of *Lecture Notes in Computer Science*, 90–99, Springer Berlin / Heidelberg, (2006).

[26] Rohit Parikh, 'Beliefs, belief revision, and splitting languages', in *Logic, Language and Computation*, eds., L.S. Moss, J. Ginzburg, and M. de Rijke, volume 2, 266–278, CSLI Publications, (1999).

[27] Guilin Qi and Jianfeng Du, 'Model-based revision operators for terminologies in description logics', in *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pp. 891–897, San Francisco, CA, USA, (2009). Morgan Kaufmann Publishers Inc.

[28] Guilin Qi, Qiu Ji, and Peter Haase, 'A conflict-based operator for mapping revision', in *Proceedings of the 22nd International Workshop on Description Logics (DL-09)*, eds., B. Cuenca Grau, J. Horrocks, B. Motik, and U. Sattler, volume 477 of *CEUR Workshop Proceedings*, (2009).

[29] Marcio M. Ribeiro and Renata Wassermann. Base revision for ontology debugging. Journal of Logic and Computation. Advanced Access, published September 5, 2008, 2008.

[30] Hans Rott, 'Two dogmas of belief revision', *The Journal of Philosophy*, **97**(9), 503–522, (2000).

[31] Zhi Zhuang, Maurice Pagnucco, and Thomas Meyer, 'Implementing iterated belief change via prime implicates', in *AI 2007: Advances in Artificial Intelligence*, eds., Mehmet Orgun and John Thornton, volume 4830 of *Lecture Notes in Computer Science*, 507–518, Springer Berlin / Heidelberg, (2007).

Equivalence Relations for Abstract Argumentation

Sjur K Dyrkolbotn [1]

Abstract. We study equivalence relations between argumentation frameworks, taking a relation to be an equivalence with respect to some semantics if it preserves and reflects the extensions of that semantics. We argue that this notion of equivalence is useful and should be considered in abstract argumentation. We go on to consider what conditions can be placed on arbitrary relations to ensure that they behave nicely with respect to equivalence. This leads us to consider bisimulations, and we show that while they do not ensure equivalence, equivalences that are also bisimulations have some nice properties with respect to semantic agreement. Then we introduce bisimulations that we call finitely collapsing. They satisfy an additional, non-local condition, and we show that they are equivalence relations with respect to all the semantics for argumentation that we consider.

1 Introduction

In abstract argumentation following Dung [8], the notion of equivalence usually adopted states that two frameworks are equivalent with respect to a semantics if they have *syntactically identical* sets of extensions under that semantics, see e.g., [13]. This is problematic for a number of reasons. First of all, it involves a peculiar attachment to the *names* of arguments - out of place, we think, in the study of abstract argumentation. This objection is typically countered by a statement to the effect that it is both well known and trivial that you can rename arguments without affecting their semantical status. While true, this is hardly satisfactory. The question immediately becomes *how* we should rename arguments so that two argumentation frameworks admit the same extensions. This, it seems, is the most interesting question, far more significant then trying to describe circumstances when the relation of identity happens to be an equivalence.

Secondly, we do not in general wish to restrict attention only to bijective functional relations that can be thought of as renamings. In fact, what seems more interesting and useful is to introduce congruences, grouping arguments together whenever they display the same behavior with respect to some semantics. The natural way to do this, we think, is to introduce a more general notion of equivalence, saying that two frameworks are equivalent with respect to a semantics if there is a relation between their arguments that both preserves and reflects extensions of that semantics. Then we must ask: *when* is a relation an equivalence? What structural properties does it need to preserve? This is the question we address in this paper.

To motivate the general notion of equivalence we adopt, we remark that relations which preserve and reflect extensions preserve and reflect what we will call *consistency*: the ability of a semantics to provide any answers about the status of an argument as either accepted or defeated. In general, semantics for argumentation can only provide a partial answer. Some arguments have no clear status, the paradigmatic example being that of a single self-attacking argument. Such an argument is inconsistent in the sense that it cannot be accepted without being defeated, and cannot be defeated without being accepted. This, it seems to us, is the general property that arguments that can neither be accepted nor defeated always share (although in general, such a picture might arise only when we consider a chain of dependencies, e.g., an attack-cycle of odd length).

This notion of consistency, while non-standard, seems like a very natural and suggestive way to talk about arguments that do not have a clear status, and for semantics based on admissible sets, a formal connection to consistency in classical logic can also be established, c.f., the discussion in Section 2. Moreover, we hope that the general notion of equivalence presented in this paper can be used to shed light on two questions that seem to be of great importance to abstract argumentation: *why* do inconsistencies sometimes arise, and *how* do we deal with them? Apart from the case of the grounded semantics, these two questions, albeit phrased in a different manner, seem to both motivate and confound most of the usual semantics adopted for argumentation frameworks.

We think that a very interesting direction of research is to attempt at exploiting the graph-theoretical structure of argumentation frameworks in order to see if some combinatorial account of inconsistency can be given. Under the stable semantics, this question is particularly critical: an argument is inconsistent (can be neither defeated nor accepted) precisely when *all* arguments are inconsistent. This happens iff the framework does not admit a stable set, and the result that a finite framework admits a stable extension as long as it does not contains attack-cycles of odd length can therefore be seen as the first non-trivial result concerning consistency in argumentation. This results was established by Dung in his original paper [8], and by Richardson, with respect to a different (but equivalent) formalism, already in the 1950s [14]. The result is very satisfying, and we find it somewhat strange that this general direction of research has received so little attention from the community. We find it strange,

[1] Department of Computer Science, University of Bergen, Norway, email: sjur.dyrkolbotn@ii.uib.no

in particular, that not more work has been devoted to the question of establishing structural conditions on frameworks that ensure the existence of stable sets (or, more generally, the existence of non-empty admissible sets). Hopefully, this paper can generate some renewed interest. We show, in particular, that it is possible to arrive at non-trivial structural conditions ensuring that a relation between frameworks is an equivalence (which preserves and reflects consistency). This, we believe, suggests that the general notion of equivalence deserves attention, especially from the point of view of trying to arrive at a graph-theoretical account of the semantic behavior of argumentation frameworks, and especially with regards to questions regarding inconsistency.

2 Background

An *argumentation framework*, framework for short, is a digraph, $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, with $\mathcal{A}$ a set of vertices, called *arguments*, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ a set of directed edges, called the *attack relation*. Unless stated otherwise, we also consider argumentation frameworks that are infinite. For $(a, b) \in \mathcal{R}$ we say that the argument a *attacks* the argument b. We use the notation $\mathcal{R}^-(x) = \{y \mid (y, x) \in \mathcal{R}\}$ and $\mathcal{R}^+(x) = \{y \mid (x, y) \in \mathcal{R}\}$, extended pointwise to sets, such that, for instance, $\mathcal{R}^+(X) = \bigcup_{x \in X} \mathcal{R}^+(x)$. For general relations $\alpha \subseteq X \times Y$, we drop $+$ as a superscript and use $\alpha(x) = \{y \mid (x, y) \in \alpha\}$ and $\alpha^-(y) = \{x \mid (x, y) \in \alpha\}$. This notation also extends pointwise to sets.

A framework $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ is a *subframework* of a framework $\mathsf{F}_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ iff $\mathcal{A} \subseteq \mathcal{A}_2$ and $\mathcal{R} \subseteq \mathcal{R}_2$. A subset of arguments $X \subseteq \mathcal{A}$ gives rise to the *induced subframework* $\mathsf{X} = \langle X, \mathcal{R}_X \rangle$ with $\mathcal{R}_X = \{(x, y) \in \mathcal{R} \mid x, y \in X\}$. $\mathsf{F} \setminus X$ denotes the subframework of F induced by $\mathcal{A} \setminus X$. A backwards infinite walk is a sequence $\lambda = x_1 x_2 x_3 \ldots$ such that $x_{i+1} \in \mathcal{R}^-(x_i)$ for all $i \geq 1$. Notice that in finite argumentation frameworks, there can be backwards infinite walks, but they must involve one or more arguments twice, i.e., they involve cycles.

The most well-known semantics for argumentation, first introduced in [8] and [3] (semi-stable semantics), are given in the following definition.[2]

Definition 2.1 *Given any argumentation framework* $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ *and a subset* $A \subseteq \mathcal{A}$*, we define* $\mathcal{D}(A) = \{x \in \mathcal{A} \mid \mathcal{R}^-(x) \subseteq \mathcal{R}^+(A)\}$*, the set of vertices* defended *by* A*. We say that*

- A *is* conflict-free *if* $\mathcal{R}^+(A) \subseteq \mathcal{A} \setminus A$*, i.e., if there are no two arguments in* A *that attack each other.*
- A *is* admissible *if it is conflict free and* $A \subseteq \mathcal{D}(A)$*. The set of all admissible sets in* F *is denoted* $a(\mathsf{F})$*.*
- A *is* complete *if it is conflict free and* $A = \mathcal{D}(A)$*. The set of all complete sets in* F *is denoted* $c(\mathsf{F})$*.*
- A *is the* grounded *set if it is complete and there is no complete set* $B \subseteq \mathcal{A}$ *such that* $B \subset A$*, it is the unique member of* $g(\mathsf{F})$*.*
- A *is* preferred *if it is admissible and not strictly contained in any admissible set. The set of all preferred sets in* F *is denoted* $p(\mathsf{F})$*.*
- A *is* stable *if* $\mathcal{R}^+(A) = \mathcal{A} \setminus A$*. The set of all stable sets in* F *is denoted* $s(\mathsf{F})$
- A *is* semi-stable *if it is admissible and there is no admissible set* B *such that* $A \cup \mathcal{R}^+(A) \subset B \cup \mathcal{R}^+(B)$*. The set of all semi-stable sets in* F *is denoted by* $ss(\mathsf{F})$*.*

For any $\mathcal{S} \in \{a, c, g, p, s, ss\}$, one also says that $A \in \mathcal{S}(\mathsf{F})$ is an *extension* (of the type prescribed by $\mathcal{S}$). For an argument $x \in \mathcal{A}$, one says that x is *credulously* accepted with respect to $\mathcal{S} \in \{a, c, g, p, s, ss\}$ if there is some $S \in \mathcal{S}(\mathsf{F})$ such that $x \in S$. One says that x is *sceptically* accepted with respect to $\mathcal{S} \in \{a, c, g, p, s, ss\}$ if $x \in \bigcap \mathcal{S}(\mathsf{F})$.

Before we embark on the question of equivalence, we briefly survey some links between argumentation, graph theory and logic. We start with graph theory. Given a directed graph (digraph) $\mathsf{D} = \langle D, N \rangle$ with $N \subseteq D \times D$, a set $K \subseteq D$ is said to be a *kernel* in D if:

$$N^-(K) = D \setminus K$$

Kernels were introduced by Von Neumann and Morgenstern in the 1940s [15] in the context of cooperative game theory and they have later attracted a fair bit of interest from graph-theorists, see [2] for a recent overview. The connection to argumentation should be apparent. If we let $\overleftarrow{\mathsf{D}}$ denote the digraph obtained by reversing all edges in D, then it is not hard to verify that a kernel in D is a stable set in $\overleftarrow{\mathsf{D}}$ and vice versa.

In kernel theory, one also considers *semikernels* [12], which are sets $L \subseteq D$ such that

$$N^+(L) \subseteq N^-(L) \subseteq D \setminus L$$

It is easy to verify that a semikernel in D is an admissible set in $\overleftarrow{\mathsf{D}}$ and vice versa. In the context of graph theory, several interesting results and techniques have been found, especially concerning the question of finding structural conditions that ensure the *existence* of kernels, see e.g., [11, 6, 7]. In our view, the connection to argumentation has not received the attention it deserves, although it has been mentioned, for instance in [5]

The second link we wish to present is with classical logic and classical consistency. This link is implicit already in much work done on argumentation, but as far as we are aware, it has only recently been pointed out that argumentation frameworks and the stable actually provide an equivalent formulation of classical propositional logic [10]. We would like to stress this point a little, since it shows that when we study structural conditions that ensure preservation of extensions based on admissible sets under mappings between frameworks, we are also studying - from a novel point of view - conditions that ensure preservation of classical consistency of theories.

For a formal account of the connection we have in mind, we refer to [1]. There the authors show that digraphs provide a normal form for propositional theories such that an assignment is satisfying for a theory iff it gives rise to a kernel in the corresponding digraph [1]. They introduce, in particular, a new normal form for propositional logic, called the *graph normal form*, where a formula ϕ is said to be in graph normal form iff $\phi = x \leftrightarrow \bigwedge_{y \in X} \neg y$ for propositional letters $\{x\} \cup X$. It is shown that it is indeed a normal form for propositional logic - every propositional theory has an equisatisfiable one containing only formulas of this form.[3] The connection between

[2] The formulation used here is not always identical to the one originally given, but is easily seen to be equivalent to it

[3] Equisatisfiable means that for every satisfying assignment to one there is a satisfying assignment to the other, i.e., the assignments are not necessarily the same (new propositional letter might need to be introduced)

theories in graph normal form and argumentation frameworks is quite obvious, and obtaining a theory from an argumentation framework is particularly easy; given a framework F, we simply form the following set of equivalences:

$$\mathsf{TF} = \{x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y \mid x \in \mathcal{A}\} \tag{2.2}$$

We adopt the convention that $x \leftrightarrow \bigwedge \emptyset$ is a tautology, and then it is easy to see that an assignment $\Gamma : \mathcal{A} \to \{\mathbf{0}, \mathbf{1}\}$ is a satisfying assignment for TF iff $S_\Gamma = \{x \in \mathcal{A} \mid \Gamma(x) = \mathbf{1}\}$ is a stable set in F. Going the other way, from theories in graph normal form to argumentation frameworks, is also straightforward, but for the details we refer to [1] (the construction is presented with respect to directed graphs, so edges must be reversed for argumentation).

So we have an immediate formal expression of the conceptual link between stable sets in argumentation and classical consistency. The difference is only a matter of *perspective*, and it is our belief that both the combinatorial perspective offered by directed graphs, and the procedural, somewhat pragmatic, perspective offered by argumentation, can serve to enhance our understanding of classical intuitions. Also, while the stable semantics expresses full classical consistency, i.e., consistency of the theory corresponding to the whole framework, other semantics based on admissible sets can be seen as identifying consistent subparts of a framework/theory that satisfy certain additional properties. To see this, it is enough to note that if $A \in a(\mathsf{F})$ is an admissible set in F, then it is a stable set in the subframework of F induced by $A \cup \mathcal{R}^+(S)$, so it corresponds to a satisfying assignment to the theory which represents this subframework. The upshot is that *all* semantic notions expressed in Definition 2.1 are based on, and expand upon, a notion of consistency that is essentially classical. This provide a fresh point of view, and we think it is particularly interesting to ask about preservation of various forms of consistency under relations between frameworks, not only because it is relevant for abstract argumentation, but also because it addresses consistency in classical logical from a new perspective.

3 A General Notion of Equivalence

Consider two arbitrary attack-cycles of even length, say F and F_2 depicted in Figure 1. How do we reason semantically about an even length attack-cycle? Well, suppose that the argument x_1 from F has some proponent. Then this proponent will probably recognize that his argument is attacked by the argument x_2, and, most likely, he will then become a proponent of argument x_3, recognizing that this argument attacks x_2 and therefore defends x_1. In F, this is when the story stops, since the proponent notices at this point that although x_4 attacks x_3, it is in turn attacked by x_1. In F_2, the story is basically the same; a proponent of x_1 realizes he should also support x_3, but now, since the cycle is longer, he also comes to support x_5.

The observation we want to make is that while the length of cycles F and F_2 differ, they are still similar. So similar, in fact, that it seems completely natural - at this level of abstraction - to say that they are semantically the same. More generally, it seems that whatever an even length cycle has to tell us with respect to any semantical notion from Definition 2.1 has been

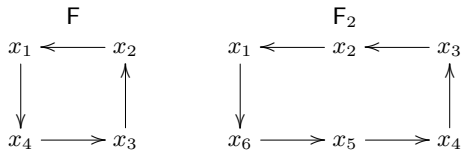


Figure 1. Two even cycles

told already by this one: $x \rightleftarrows y$. Essentially, all even cycles behave the same way; they are different manifestations of exactly the same argumentation scenario. Unfortunately, the notion of equivalence adopted in the literature on argumentation does not allow us to conclude this; even cycles of different length do not have the same set of extensions under any reasonable semantics.

The case of even cycles seems to illustrate in a very simple way why the current notion of equivalence used in argumentation is too restrictive. It relies on a crude syntactic criterion requiring extension - semantic in nature - to be syntactically the same. In light of this, we believe that the following notion of equivalence should be investigated. It seems completely natural and is determined not by looking for syntactic identity between sets of arguments, but by looking for sets of arguments that can be grouped together upon noting that they have the same semantic status.

Definition 3.1 *Given two argumentation frameworks* F *and* F_2*, we say that they are equivalent with respect to* $\mathcal{S} \in \{a, c, g, p, s, ss\}$*, and we write* $\mathsf{F} \equiv^{\mathcal{S}} \mathsf{F}_2$*, if there is a relation* $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ *such that*

- *If* $A \in \mathcal{S}(\mathsf{F})$*, then* $\alpha(A) \in \mathcal{S}(\mathsf{F}_2)$ *- the relation* preserves *extensions*
- *If* $A_2 \in \mathcal{S}(\mathsf{F}_2)$ *then* $\alpha^-(A_2) \in \mathcal{S}(\mathsf{F})$ *- the relation* reflects *extensions*

If $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ witnesses to the equivalence of F and F_2, we say that α is an equivalence relation. For the case of even cycles, it is easy to see that this definition is adequate. It allows us to state formally what our intuition told us to be the case, namely $\mathsf{F} \equiv^S \mathsf{F}_2$ for all $\mathcal{S} \in \{a, c, g, p, s, ss\}$. The relation $\alpha = \{(x_1, x_1), (x_1, x_3), (x_2, x_2), (x_2, x_4), (x_3, x_5), (x_4, x_6)\}$, for instance, is easily seen to be an equivalence relation with respect to all $\mathcal{S} \in \{a, c, g, p, s, ss\}$. Indeed, for arbitrary even cycles $x_1 \ldots x_{2i} x_1$, it is easy to see that for all $\mathcal{S} \in \{a, c, g, p, s, ss\}$ they are all equivalent to each other. In particular, they are equivalent to the even cycle $x_1 x_2 x_1$, witnessed by the equivalence relation $\alpha = \bigcup_{1 \leq i \leq n} \{(x_1, x_{2i-1}), (x_2, x_{2i})\}$.

3.1 First Observation: Skeptical and Credulous Acceptance

The first observation we would like to make regarding Definition 3.1 is that - unsurprisingly - equivalences preserve and reflect skeptical and credulous acceptance of arguments. It is clear, in particular, that if $\mathsf{F} \equiv^{\mathcal{S}} \mathsf{F}_2$ and $S \subseteq \mathcal{A}$ is a set of skeptically accepted arguments from F, then $\alpha(S)$ is a skeptically accepted set of arguments in F_2 (and similarly for the inverse α^-). Also, if $C \subseteq \mathcal{A}$ is a set of credulously accepted

arguments, then for each $x \in C$, we have $S_x \in \mathcal{S}(\mathsf{F})$ such that $x \in S_x$, and since $\alpha(S_x) \in \mathcal{S}(\mathsf{F}_2)$ by α being an equivalence, it follows that $\alpha(C)$ is a set of credulously accepted arguments in F_2 as well. More is true, however, and what our definition of equivalence ensures is that the *logical* properties of frameworks are preserved. For instance, if one of the logical properties of F is that all extension under $\mathcal{S}$ containing $x \in \mathcal{A}$ must also contain $y \in \mathcal{A}$, the same relationship obtains between all $x_2 \in \alpha(x)$ and all $y_2 \in \alpha(y)$. We obtain, in particular, two collections of equivalent arguments in F_2 such that one logically implies the other. Then the benefit of having defined equivalence as in Definition 3.1 becomes clear; since our notion of an equivalence does not impose any restrictions on what the relation must look like, we can investigate logical properties of complex frameworks by looking for equivalences with more simple frameworks that have already been analyzed.

3.2 Second Observation: Collapse with respect to the Single-Status Semantics

The second observation we will make is almost as trivial as the first, but might make the notion of equivalence introduced in Definition 3.1 somewhat controversial to the argumentation community. Consider, in particular, two frameworks F and F_2 and a semantics $\mathcal{S} \in \{a, c, g, p, s, ss\}$ such that both F and F_2 have a *unique* extension $\{S\} = \mathcal{S}(\mathsf{F}), \{S_2\} = \mathcal{S}(\mathsf{F}_2)$. If we assume both S, S_2 to be non-empty, it is obvious that we can always construct a relation $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ such that $\alpha(S) = S_2$ and $\alpha^-(S_2) = S$, allowing us to conclude that $\mathsf{F} \equiv^{\mathcal{S}} \mathsf{F}_2$.

With respect to the grounded extension, which always gives rise to a unique extension, this means that all frameworks fall into one of two classes; those that admit non-empty grounded extensions and those that do not. More is true, since it is well known, see e.g., [8], that for any two non-empty *finite acyclic* frameworks, all semantics from Definition 2.1 coincide and deliver a unique non-empty extension - the grounded one. This means, in particular, that with equivalence conceived of as in Definition 3.1, all finite, non-empty, acyclic frameworks are equivalent. Also, we note that other semantics for argumentation have also been proposed that always yield a unique extension - they are called *single-status* in the literature. In light of this, the collapse of frameworks with respect to all such semantics might disconcert some, but to us it signals only that we have arrived at a notion of equivalence that is appropriate. It allows us to abstract away from superficial syntactical differences and focus instead on genuine semantic problems.

The grounded semantics for argumentation is particularly trivial; the grounded extension can always be computed in linear time (iterate $\mathcal{D}()$ from Definition 2.1, starting from the set, U, of unattacked arguments), and it contains arguments that, intuitively speaking, cannot be disputed by any rational agent. Indeed, if a semantics for argumentation was proposed that did not include the grounded extension as a subset of all extensions, it would probably be dismissed without further comment. But in some sense - and we believe it is the most relevant sense - all single-status semantics are trivial. They leave no room for dispute, no contingency, and, most critically, no interesting dependencies between arguments. Such semantics simply pick a set, and it seems clear that the interesting question, and the only possible source of non-triviality, lies in *how* the set is chosen. Clearly, if this is something more than an arbitrary choice, it must involve other notions, and it is these notions - which typically *do* involve interesting dependencies - that are truly semantic in nature and deserve attention. The point we are trying to make is beautifully illustrated by the so-called *ideal semantics* [9]. The ideal set of arguments is the maximal set of arguments that is contained in all preferred extensions. As such, the ideal semantics should, in our opinion, not be seen as a separate semantics at all, but just as a new notion of acceptance for preferred semantics, asking you to accept an argument only if it is skeptically accepted and is also in an admissible set which contains only skeptically accepted arguments (since defense is preserved under union and the set of skeptically accepted arguments is conflict-free, the set of all such arguments will obviously be the maximal admissible subset of skeptically accepted arguments). It seems to us that the relevant notion of equivalence is still the one which preserves and reflects preferred sets - there is nothing you can say about the ideal set and what it captures unless you make reference to the notion of a preferred set. [4]

We remark that the collapse with respect to single-status semantics has an obvious generalization, allowing us to conclude that any two frameworks with exactly $n \in \mathbb{N}$ disjoint extensions under some semantics are equivalent with respect to that semantics. Any two such frameworks are equivalent, as they should be, because there is a way to associate arguments such that a one-to-one correspondence between the extensions of these frameworks will result.

Thinking of arguments as propositional formulas (remember the discussion in Section 2), makes for a further argument in favor of the possibly controversial point of view that we adopt here. What single-status approaches provide us with is basically a set of tautologies - arguments that cannot be disputed. In a logical sense, any two collections of tautologies are equivalent, and they should be; no questions arise at all about how their semantic status is dependent on that of other formulas, the point being precisely that no such dependencies influence their status as indisputable. It seems clear, therefore, that a collection of arguments that cannot be disputed should be regarded as logically equivalent to any other such collection, in exactly the same way as a collection of tautologies of some logical language is equivalent to any other such collection. What is interesting about tautologies is how to locate them, and the general notion of equivalence is potentially useful in this regard precisely because it does not care what they look like. That way, it becomes possible to look for relations that allows simplification of the framework under consideration, potentially simplifying the search for tautologies. For the finite case and semantics based on admissible sets, this is only a relevant consideration for cyclic frameworks, however, since the search for tautologies in a finite acyclic frameworks is already completely trivial.

3.3 Third Observation: Structural Conditions Needed

We have introduced a new semantic notion of equivalence between frameworks, and argued that it is the appropriate notion that we want to work with when we consider two frame-

[4] We mention that we can impose the same restriction starting from semi-stable semantics, leading to the *eager* set [4]

works and ask about the relationship between them. Some might object that it is too abstract, referring to how it conflates frameworks with respect to the grounded semantics and in any unique status situation. But as we have tried to argue above, we actually believe that such a conflation is in order when we work at a high level of abstraction. For the case of the grounded extension, in particular, it seems to us that there is not much more to be said about it at the level of abstraction that we address. The grounded extension might be very useful in applications, and it might be possible to focus on more intermediate levels of abstraction where some, but not all implementation-specific aspects are studied. But from the point of view of *pure* abstract argumentation, as introduced by Dung, we are bold enough to suggest that the grounded extension is perhaps properly understood already. What is not understood, however, not even at a high level of abstraction, is the notion of an admissible set; in particular, we do not seem to have a clear understanding of *when* non-empty such sets can be found, *why* they sometimes fail to exist, and *how* we best should go about locating them. As discussed earlier, this question hinges on the notion of *consistency*, in various forms and guises. If the question is simply whether or not a framework admits a stable set, the question becomes that of deciding classical consistency, as discussed above in Section 2. But when we make the move to consider admissible sets, we are free to also reason about and locate consistent sub-parts of a system that could, as a whole, be inconsistent. However, since what - in terms of structural properties - leads to inconsistency in argumentation frameworks is not properly understood, it is also difficult to pin down where the problem lies, with repercussion also for what exactly the non-stable semantics contribute in such cases. A fundamental, overreaching research goal - as we see it - should be to attempt giving an account of this by combinatorial means.

We think it is obvious that in this regard, the notion provided by Definition 3.1 is appropriate and should be considered. Still, it only states what an equivalence is, not how to find one. Unless we can establish some structural properties on relations that ensure that they are equivalences, it would be fairly useless, pointing only to an unattainable ideal that would have to be replaced by more pragmatic notions in practice. In the following section, however, we present first results on this, exploring the notion of bisimulation.

4 Bisimulation and Equivalence in Argumentation

In this section, we first work with a standard notion of bisimulation, and show that if equivalence with respect to admissible semantics is witnessed by a bisimulation, we can conclude equivalence also for some (but not all) semantics based on admissible sets. Then we add a further requirement to bisimulations - introducing finitely collapsing bisimulations - and we show that they are equivalences with respect to all the semantics we consider in this paper.

Definition 4.1 *Given argumentation frameworks* F *and* F_2*, a relation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ *is said to be a bisimulation if we have:*
forth*: For every* $x \in \mathcal{A}$*,* $y \in \mathcal{R}^-(x)$*, for all* $x_2 \in \beta(x)$ *there is* $y_2 \in \mathcal{R}_2^-(x_2) \cap \beta(y)$
back*: For every* $x_2 \in \mathcal{A}_2, y_2 \in \mathcal{R}_2^-(x_2)$*, for all* $x \in \beta^-(x_2)$ *there is* $y \in \mathcal{R}^-(x) \cap \beta^-(y_2)$

Notice that the definition asks for mutual simulation of *incoming* attacks. For $\mathcal{S} \in \{a, c, p, s, ss\}$, it is not hard to see that bisimulations are neither necessary nor sufficient for equivalence. The problem is that a bisimulation ensures only that attacks between arguments are preserved and reflected, but does not ensure that attacks are absent when they need to be in order to ensure conflict-freeness. It is easy to see, for instance, that an even cycle is bisimilar to a single self-attacking argument, and these two frameworks are only equivalent under the grounded semantics. We have the following easy fact, however, stating that bisimulation behaves nicely when it comes to defense.

Fact 4.2 *Assume we have frameworks* F, F_2 *and some bisimulation* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$*. Then we have*

(1) For all $A \subseteq \mathcal{A}$*,* $\beta(\mathcal{D}(A)) = \mathcal{D}(\beta(A))$ *- β preserves defended arguments*
(2) For all $A_2 \subseteq \mathcal{A}_2$*,* $\beta^-(\mathcal{D}(A_2)) = \mathcal{D}(\beta^-(A_2))$ *- β reflects defended arguments*

Proof. (1) We consider arbitrary $A \subseteq \mathcal{A}$ and prove the claim by showing both inclusions.
($\subseteq$) Consider arbitrary $y \in \mathcal{D}(A)$, $y_2 \in \beta(y)$ and $z_2 \in \mathcal{R}_2^-(y_2)$. Then by β being a bisimulation (back), it follows that there is some $z \in \beta^-(z_2)$ such that $z \in \mathcal{R}^-(y)$. Since $y \in \mathcal{D}(A)$ it follows that there is $x \in A$ such that $x \in \mathcal{R}^-(z)$. Then by β being a bisimulation (forth) it follows that there is some $x_2 \in \beta(x)$ such that $x_2 \in \mathcal{R}_2^-(z_2)$, meaning $z_2 \in \mathcal{R}_2^-(\beta(A))$. We conclude $y_2 \in \mathcal{D}(\beta(A))$ as desired.
($\supseteq$) Consider arbitrary $y_2 \in \mathcal{D}(\beta(A))$, $y \in \beta^-(y_2)$ and $z \in \mathcal{R}^-(y)$. Then by β being a bisimulation (forth), it follows that there is some $z_2 \in \beta(z)$ such that $z_2 \in \mathcal{R}^-(y_2)$. Since $y_2 \in \mathcal{D}(\beta(A))$, it follows that there is some $x_2 \in \beta(A)$ such that $x_2 \in \mathcal{R}^-(z_2)$. From β being a bisimulation (back), it follows that there is some $x \in \beta^-(x_2)$ such that $x \in \mathcal{R}^-(z)$. It follows that $y \in \mathcal{D}(A)$, meaning that $y_2 \in \beta(\mathcal{D}(A))$ as desired.
(2) The argument is symmetric to that used to show (1). □

We note that a trivial corollary of this is that bisimulations are equivalences with respect to the grounded semantics. The next result concerns the relationship between various semantics. We ask, in particular, if equivalences that are also bisimulations will automatically preserve and reflect extensions for more than one type of semantics from Definition 2.1 at once. We show, in particular, that if an equivalence with respect to admissible sets is also a bisimulation, then it is also an equivalence with respect to preferred, stable and semi-stable semantics, yet *not* with respect to the complete semantics.

Theorem 4.3 *Given frameworks* F *and* F_2*, if* $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ *is a bisimulation, then if β preserves and reflects admissible sets, it also preserves and reflects preferred, semi-stable and stable sets.*

Proof. For all semantics, we only show preservation. Reflection can be shown symmetrically.

Stable: Assume that $S \subseteq \mathcal{A}$ is stable. We know $\beta(S)$ is conflict-free and must show $\mathcal{A}_2 \setminus \beta(S) = \mathcal{R}_2^+(\beta(S))$. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus \beta(S)$. Then $\beta^-(x_2) \subseteq \mathcal{A} \setminus S$, so there is $y \in S$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. By β being a bisimulation ("forth"), we have $x_2 \in \mathcal{R}_2^+(\beta(S))$ as desired.
Preferred: Assume that $S \subseteq \mathcal{A}$ is preferred. Then $\beta(S)$ is admissible. Assume towards contradiction that there is $A_2 \supset \beta(S)$ which is admissible in F_2. Then $\beta^-(A_2)$ is admissible in F and since $\beta(\beta^-(A_2)) \supseteq A_2 \supset \beta(S)$, we have $\beta^-(A_2) \supset S$, contradiction.
Semi-stable: Assume that $S \subseteq \mathcal{A}$ is semi-stable, i.e. that S is admissible, and that there is no admissible $A \subseteq \mathcal{A}$ such that $S \cup \mathcal{R}^+(S) \subset A \cup \mathcal{R}^+(A)$. Assume towards contradiction that $\beta(S)$ is not semi-stable. Then there is $S_2 \subseteq \mathcal{A}_2$ such that a) $S_2 \cup \mathcal{R}_2^+(S_2) \supset \beta(S) \cup \mathcal{R}_2^+(\beta(S))$. By β being a bisimulation ("forth"), we have b) $\beta(\mathcal{R}^+(S)) \subseteq \mathcal{R}_2^+(\beta(S))$ and also ("back") that c) $\beta^-(\mathcal{R}_2^+(S_2)) \subseteq \mathcal{R}^+(\beta^-(S_2))$. We will show that $\beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) = \beta^-(S_2) \cup \beta^-(\mathcal{R}_2^+(S_2)) \supset S \cup \mathcal{R}^+(S)$, which is a contradiction since it allows us to conclude, by applying c), that $\beta^-(S_2) \cup \mathcal{R}^+(\beta^-(S_2)) \supset S \cup \mathcal{R}^+(S)$. We show inclusion first.

$$\begin{aligned}\beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) &\overbrace{\supseteq}^{a)} \beta^-(\beta(S) \cup \mathcal{R}_2^+(\beta(S))) \\ &= \beta^-(\beta(S)) \cup \beta^-(\mathcal{R}_2^+(\beta(S))) \\ &\overbrace{\supseteq}^{b)} \beta^-(\beta(S)) \cup \beta^-(\beta(\mathcal{R}^+(S))) \\ &\supseteq S \cup \mathcal{R}^+(S)\end{aligned}$$

To show that the inclusion is strict, consider $x_2 \in (S_2 \cup \mathcal{R}_2^+(S_2)) \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S))$. For arbitrary $x \in \beta^-(x_2)$, observe first that since $x_2 \notin \beta(S)$, we have $x \notin S$. We also have $x_2 \notin \mathcal{R}_2^+(\beta(S))$ and from b) it follows that $x_2 \notin \beta(\mathcal{R}^+(S))$. Then we conclude that $x \notin \mathcal{R}^+(S)$. □

Interestingly, a bisimulation that preserves and reflects admissible sets might not preserve complete sets, as shown by the frameworks F and F_2 in Figure 2. Here, we have the bisimulation $\beta = \{(a,a),(e,a),(b,b),(d,b),(f,b),(c,c)\}$ which is also an equivalence with respect to the admissible semantics. We notice, however, that $\{a\}$ is a complete set in F while $\beta(a) = \{a\}$ is not complete in F_2 since d is defended by $\{a\}$.

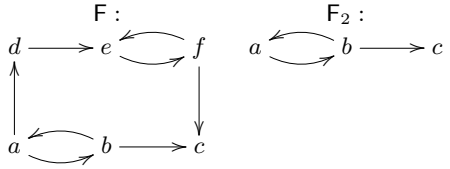


Figure 2. Frameworks F, F_2 such that we have $\mathsf{F} \equiv^{\mathcal{S}} \mathsf{F}_2$ for $\mathcal{S} \in \{g, a, p, s, ss\}$ but $\mathsf{F} \not\equiv^{c} \mathsf{F}_2$

As mentioned, the intuitive reason why bisimulations do not preserve extensions is that they do not preserve conflict-freeness. Still, they fail to do so only in specific circumstances. To see how this works, assume that you have two arguments a, b in some framework F such that a and b are not in any conflict, and that you then relate them by a bisimulation β to some a_2, b_2 in F_2 with $b_2 \in \mathcal{R}^-(a_2)$. It then follows by β being a bisimulation (back), that there must be some $c \in \beta^-(b_2)$ such that $c \in \mathcal{R}^-(a)$. So an attacker of a, the argument c, was merged with a non-attacker of a, the argument b. So this type of collapse has to occur when bisimulations fail to be equivalences. It makes sense, then, to see what happens if we attempt to limit it by introducing a further requirement. In particular, we will investigate what happens when we do not allow the collapse of any two disjoint infinite backwards walks.

Definition 4.4 *Given two frameworks F and F_2, a bisimulation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ is* finitely collapsing *if the following holds:*
global forth: *For every backwards infinite walk $\lambda = x_1 x_2 x_3 \ldots$ in F_2, there exists some $i \in \mathbb{N}$ such that $|\beta^-(x_i)| = 1$*
global back: *For every backwards infinite walk $\lambda = x_1 x_2 x_3 \ldots$ in F, there exists some $i \in \mathbb{N}$ such that $|\beta(x_i)| = 1$*

For short we will call bisimulations that are finitely collapsing fc-bisimulations. As an example, consider the frameworks in Figure 3. They are fc-bisimilar witnessed by $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ where $\beta(a) = a_2, \beta(b) = \beta(d) = b_2, \beta(c) = c_2$.

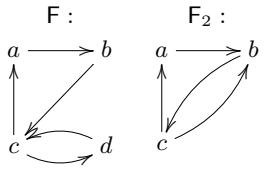


Figure 3. Two fc-bisimilar argumentation frameworks

The main result in this paper now follows. It shows that fc-bisimulations are equivalences with respect to all semantics in Definition 2.1. We remark that it is sufficient to show that fc-bisimulations preserve and reflect admissible and complete sets, from which it follows by Theorem 4.3 that they also preserve and reflect preferred, stable and semi-stable sets.

Theorem 4.5 *Given frameworks F and F_2, if there is an fc-bisimulation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$, then $\mathsf{F} \equiv^{\mathcal{S}} \mathsf{F}_2$ for all $\mathcal{S} \in \{s, a, p, ss, c\}$*

Proof. **Admissible sets:** Let $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ be an arbitrary fc-bisimulation. We show that β preserves admissible sets. Then, by symmetry, β also reflects them, since the inverse of β, $\beta^- \subseteq \mathcal{A}_2 \times \mathcal{A}$ is clearly also an fc-bisimulation. Let $E \subseteq \mathcal{A}$ be an admissible set in F and consider $E_2 = \beta(E)$. If $x_2 \in \mathcal{R}_2^-(y_2)$ for $y_2 \in E_2$, then there is $y \in E$ such that $y_2 \in \beta(y)$, and by β being a bisimulation ("back"), there is some $x \in \mathcal{R}^-(y)$ such that $x_2 \in \beta(x)$. Since E defends itself, it follows that there is $z \in \mathcal{R}^-(x) \cap E$. Then, by β being a bisimulation ("forth"), it follows that there is some $z_2 \in \mathcal{R}_2^-(x_2)$ such that $z_2 \in \beta(z)$, meaning $z_2 \in E_2$. This shows that $E_2 \subseteq \mathcal{D}(E_2)$. To show that E_2 is conflict free, assume towards contradiction that there is $x_2, b' \in E_2$ with $x_2 \in \mathcal{R}_2^-(b')$. Then, by definition of E_2, there is $x, b \in E$ with $x_2 \in \beta(x)$ and $b' \in \beta(b)$. Also, we know that $x \notin \mathcal{R}^-(b)$ since E is conflict-free. But

by β being a bisimulation ("back"), there must be $z \in \mathcal{R}^-(b)$ such that $x_2 \in \beta(z)$. Since E is conflict-free, we know that $z \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$ Now we have $x_2 \in E_2 \cap \beta(x) \cap \beta(z)$ such that z attacks E, and this is the first step towards showing that there exists an infinite backwards walk $\lambda = y_1 y_2 y_3 \ldots$ in $\mathcal{A}_2$ such that for all $i \geq 1$, we have $|\beta^-(y_i)| \geq 2$. This will contradict the assumption that β is an fc-bisimulation ("global forth"). We take $y_1 = x_2$ and let $w_1 = x$, $v_1 = z$. Then for all $i \geq 2$, we define y_i, w_i, v_i inductively, assuming that $y_{i-1}, w_{i-1}, v_{i-1}$ have been defined such that $w_{i-1} \in E, v_{i-1} \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$ and $y_{i-1} \in \beta(w_{i-1}) \cap \beta(v_{i-1})$. The construction is visualized in Figure 4. Since E defends itself

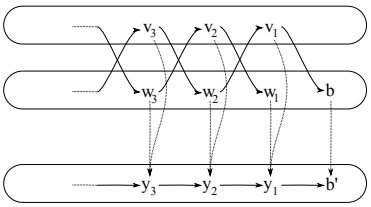


Figure 4. Illustrating the construction of $\lambda = y_1 y_2 y_3 \ldots$

against all attacks, we can find $w_i \in E \cap \mathcal{R}^-(v_{i-1})$. Since we have $y_{i-1} \in \beta(v_{i-1})$ it follows by β being a bisimulation ("forth") that we can find $y_i \in \beta(w_i) \cap \mathcal{R}^-(y_{i-1})$. But we also have $y_{i-1} \in \beta(w_{i-1})$, so by β being a bisimulation ("back"), we find $v_i \in \beta^-(y_i) \cap \mathcal{R}^-(w_{i-1})$. Since $w_{i-1} \in E$ and E is conflict-free, it follows that $v_i \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$. So y_i, w_i, v_i can be found for all $i \in \mathbb{N}$, proving existence of λ that contradicts "global forth".

Complete sets: We know that β preserves and reflects admissible sets, and now we assume that $S \subseteq \mathcal{A}$ is complete. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S)))$. By β being a bisimulation ("forth"), we get $\beta^-(x_2) \cap \mathcal{R}^+(S) = \emptyset$, which implies $\beta^-(x_2) \subseteq \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. Then, since S is complete, there is $y \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. Then, since β is a bisimulation ("forth"), it follows that there is $y_2 \in \beta(y) \cap \mathcal{R}_2^-(x_2)$. Since $x_2 \notin \mathcal{R}_2^+(\beta(S))$ it follows that $y_2 \notin \beta(S)$. Assume towards contradiction that $y_2 \in \mathcal{R}_2^+(z_2)$ for some $z_2 \in \beta(S)$. Then there is $z \in S \cap \beta^-(z_2)$ and also, since β is a bisimulation ("back"), there is $z' \in \mathcal{R}^-(y) \cap \beta^-(z_2)$. Since $y \notin \mathcal{R}^+(S)$, $z' \notin S$. Since β is a bisimulation ("forth") and $z_2 \in \beta(S)$ and $\beta(S)$ is conflict-free, $z' \notin \mathcal{R}^+(S)$. It follows that $z' \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. To contradict global forth, we prove existence of a backwards infinite walk $\lambda = x_1 x_2 x_3 \ldots$ in F_2 such that for all $i \geq 1$ we have $|\beta^-(x_i)| \geq 2$. We take $x_1 = z_2$, $v_1 = z', w_1 = z$ and for all $i \geq 2$, we assume that we have $x_{i-1}, v_{i-1}, w_{i-1}$ with $x_{i-1} \in \beta(S) \cup \mathcal{R}_2^+(\beta(S))$ and $w_{i-1} \in (S \cup \mathcal{R}^+(S)) \cap \beta^-(x_{i-1})$, $v_{i-1} \in (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S))) \cap \beta^-(x_{i-1})$. There are two cases.

I) $x_{i-1} \in \beta(S)$. Then since $\beta(S)$ is admissible and $w_{i-1} \in \beta^-(x_{i-1})$, we have $w_{i-1} \notin \mathcal{R}^+(S)$ by β being a bisimulation ("forth"). Since S is complete, we find $v_i \in \mathcal{R}^-(v_{i-1}) \cap (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S)))$. Since β is a bisimulation ("forth"), we find $x_i \in \mathcal{R}_2^-(x_{i-1}) \cap \beta(v_i)$, and since $\beta(S)$ is admissible, $x_i \in \mathcal{R}_2^+(\beta(S))$. Then, going back, we find $w_i \in \beta^-(x_i) \cap \mathcal{R}^-(w_{i-1})$, and since $w_{i-1} \in S$ and S is admissible, $w_i \in \mathcal{R}^+(S)$.

II) $x_{i-1} \in \mathcal{R}_2^+(\beta(S))$. Since $w_{i-1} \in \beta^-(x_{i-1}) \cap (S \cup \mathcal{R}^+(S))$ and $\beta(S)$ is admissible, we have $w_{i-1} \in \mathcal{R}^+(S)$. We choose $w_i \in S \cap \mathcal{R}^-(w_{i-1})$. By β being a bisimulation ("forth"), we find $x_i \in \beta(w_i) \cap \mathcal{R}_2^-(x_{i-1})$ and ("back") $v_i \in \beta^-(x_i) \cap \mathcal{R}^-(v_{i-1})$. Since $v_{i-1} \notin \mathcal{R}^+(S)$, $v_i \notin S$. Also, by β being a bisimulation ("forth") and $x_i \in \beta(v_i) \cap \beta(S)$ and $\beta(S)$ being conflict-free, we have $v_i \notin \mathcal{R}^+(S)$.

Having established the claim for $\mathcal{S} \in \{a, c\}$, the claim follows by Theorem 4.3 for all $\mathcal{S} \in \{a, c, p, ss, s\}$

□

5 Conclusion

We have addressed the notion of equivalence in abstract argumentation, arguing for a general notion that allows us to consider arbitrary relations between frameworks. We suggested that searching for maps between frameworks that preserve and reflect extensions is worthwhile, and we established a first result on this, introducing finitely collapsing bisimulations and proving that they are equivalences with respect to all the semantics we consider. On a more general note, we suggested that investigating equivalence should be conceived of as part of a direction of research where one attempts to provide graph-theoretical characterizations of various logical properties of argumentation frameworks. We suggested that the notion of *consistency*, in particular, is interesting to look at from a combinatorial point of view. For future work, we hope to be able to identify further structural requirements that ensure relations to be equivalences, and we hope to arrive at a more complete understanding of what structures needs to be present in frameworks in order for different semantics for argumentation to actually disagree.

REFERENCES

[1] Marc Bezem, Clemens Grabmayer, and Michał Walicki. Expressive power of digraph solvability. *Annals of Pure and Applied Logic*, 2011. [to appear].

[2] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.

[3] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[4] Martin Caminada. Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *BNAIC 2007*, pages 81–87, 2007.

[5] Sylvie Coste-marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *Proc. 8th European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU), volume 3571 of LNAI*, pages 317–328. Springer-Verlag, 2005.

[6] Pierre Duchet. Graphes noyau-parfaits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.

[7] Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l'existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.

[8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[9] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(1015):642 – 674, 2007.

[10] Sjur Dyrkolbotn and Michał Walicki. Propositional discourse logic. *(submitted)*. www.ii.uib.no/ michal/graph-paradox.pdf.

[11] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.

[12] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.

[13] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, 175(14–15):1985–2009, 2011.

[14] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.

[15] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).

BNC@ECAI 2012 Workshop Notes