Paraconsistent Reasoning with OWL – Algorithms and the ParOWL Reasoner *

Yue Ma[1,2], Pascal Hitzler[2], and Zuoquan Lin[1]

[1]Department of Information Science, Peking University, China
[2]AIFB, Universität Karlsruhe, Germany
{mayue,lz}@is.pku.edu.cn, {yum,hitzler}@aifb.uni-karlsruhe.de

Technical Report
AIFB, University of Karlsruhe, Germany
December 2006

Abstract. In an open, constantly changing and collaborative environment like the forthcoming Semantic Web, it is reasonable to expect that knowledge sources will contain noise and inaccuracies. Practical reasoning techniques for ontologies therefore will have to be tolerant to this kind of data, including the ability to handle inconsistencies in a meaningful way. For this purpose, we employ paraconsistent reasoning based on four-valued logic, which is a classical method for dealing with inconsistencies in knowledge bases. Its transfer to OWL DL, however, necessitates the making of fundamental design choices in dealing with class inclusion, which has resulted in differing proposals for paraconsistent description logics in the literature. In this paper, we build on one of the more general approaches which due to its flexibility appears to be most promising for further investigations. We present two algorithms suitable for implementation, one based on a preprocessing before invoking a classical OWL reasoner, the other based on a modification of the KAON2 transformation algorithms. We also report on our implementation, called ParOWL.

1 Introduction

Real knowledge bases and data for Semantic Web applications will rarely be perfect. They will be distributed and multi-authored. They will be assembled from different sources and reused. It is unreasonable to expect such reaslistic knowledge bases to be always logically consistent, and it is therefore important to study ways of dealing with inconsistent knowledge. This is particularly important if the full power of logic-based approaches like the Web Ontology Language OWL [1] shall be employed, as classical logic breaks down in the presence of inconsistent knowledge.

The study of inconsistency handling in Artificial Intelligence has a long tradition, and corresponding results are recently being transferred to description logics, which

* We acknowledge support by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 B), by the EU under the IST project NeOn (IST-2006-027595, http://www.neon-project.org/), by the Deutsche Forschungsgemeinschaft (DFG) in the ReaSem project and by the China National Study Abroad Fund.

underly OWL. Two fundamentally different approaches can be distinguished. The first is based on the assumption that inconsistencies indicate erroneous data which is to be repaired in order to obtain a consistent knowledge base, e.g. by selecting consistent subsets for the reasoning process [2, 3]. The other approach yields to the insight that inconsistencies are a natural phenomenon in realistic data which are to be handled by a logic which tolerates it [4–7]. Such logics are called paraconsistent, and the most prominent of them are based on the use of additional truth values standing for *underdefined* (i.e. neither true nor false) and *overdefined* (or *contradictory*, i.e. both true and false). Such logics are appropriately called *four-valued logics* [8, 9]. We believe that either of the approaches is useful, depending on the application scenario.

In this paper, we contribute to the paraconsistency approach in terms of four-valued description logic. We indeed extend on the preliminary work in [7], which has the following features.

- It is grounded in prominent research results from Artificial Intelligence [10].
- It is very flexible in terms of design choices which have to be made when developing a paraconsistent description logic. This concerns the issues arising from the fact that there are different ways of defining the notion of logical implication in four-valued logics. The approach which we follow allows the full and simultaneous use of the different notions of implication.
- It does not increase worst-case computational complexity of reasoning if compared to standard reasoning methods for consistent knowledge bases [11].

In this paper, we present two algorithms for practical paraconsistent reasoning based on this approach. The first one is based on a transformation from a paraconsistent ontology O to a classical two-valued ontology $\overline{O}$ in such a way that paraconsistent reasoning on O can be simulated by classical reasoning on $\overline{O}$. The second algorithm is based on an adaptation of the algorithms underlying the KAON2 OWL Reasoner[1] [12] and thus realises a resolution-based decision procedure for paraconsistent reasoning. We spell out the details for the description logic $\mathcal{ALC}$, which is considered to be the most foundational one and comprises a large fragment of OWL DL.

The paper is structured as follows. We first review briefly preliminaries in Section 2. In Section 3 we then describe the syntax and semantics of the paraconsistent description logic which we will use. Sections 4 and 5 describe the two reasoning procedures, respectively based on a transformation for preprocessing and on an adaptation of the KAON2 algorithms. We discuss future work and conclude in Section 7.

This paper is a substantial continuation of work presented as preliminary results in [7].

2 Preliminaries

2.1 The Description Logic $\mathcal{ALC}$

We briefly review notation and terminology of the description logic $\mathcal{ALC}$, but we basically assume that the reader is familiar with description logics. For comprehensive background reading, please refer to [13].

[1] http://kaon2.semanticweb.org

Table 1. Syntax and semantics of $\mathcal{ALC}$

Constructor Name	**Syntax**	**Semantics**
atomic concept A	A	$A^I \subseteq \Delta^I$
abstract role R_A	R	$R^I \subseteq \Delta^I \times \Delta^I$
individuals I	o	$o^I \in \Delta^I$
top concept	$\top$	Δ^I
bottom concept	$\bot$	$\emptyset$
conjunction	$C_1 \sqcap C_2$	$C^I \cap D^I$
disjunction	$C_1 \sqcup C_2$	$C^I \cup D^I$
negation	$\neg C$	$\Delta^I \setminus C^I$
exists restriction	$\exists R.C$	$\{x \mid \exists y, (x,y) \in R^I \text{ and } y \in C^I\}$
value restriction	$\forall R.C$	$\{x \mid \forall y, (x,y) \in R^I \text{ implies } y \in C^I\}$

Axiom Name	**Syntax**	**Semantics**
concept inclusion	$C_1 \sqsubseteq C_2$	$C_1^I \subseteq C_2^I$
individual inclusion	$C(a)$	$a^I \in C^I$

We assume that we are given a set of atomic concepts (or concept names), a set of roles (or role names), and a set of individuals. With the symbols $\top$ and $\bot$ we furthermore denote the top concept and the bottom concept, respectively.

Complex concepts in $\mathcal{ALC}$ can be formed from these inductively as follows.

1. $\top$, $\bot$, and each atomic concept are concepts;
2. If C, D are concepts, then $C \sqcup D$, $C \sqcap D$, and $\neg C$ are concepts;
3. If C is a concept and R is a role, then $\forall R.C$ and $\exists R.C$ are concepts.

An $\mathcal{ALC}$ ontology consists of a set of assertions, called the $ABox$ of the ontology, and a set of inclusion axioms, calld the $TBox$ of the ontology. Assertions are of the form $C(a)$ or $R(a,b)$, where a, b are individuals and C and R are concepts and roles, respectively. Inclusion axioms are of the form $C \sqsubseteq D$, where C and D are concepts. Informally, an assertion $C(a)$ means that the individual a is an instance of concept C, and an assertion $R(a,b)$ means that individual a is related with individual b via the property R. The inclusion axiom $C \sqsubseteq D$ means that each individual of C is an individual of D.

The formal definition of the (model-theoretic) semantics of $\mathcal{ALC}$ is given by means of interpretations $I = (\Delta^I, \cdot^I)$ consisting of a non-empty domain Δ^I and a mapping $\cdot^I$ satisfying the conditions in Table 1, interpreting concepts as subsets of the domain and roles as binary relations on the domain. An interpretation satisfies an $\mathcal{ALC}$ ontology (i.e. is a model of the ontology) iff it satisfies each axiom in both the $ABox$ and the $TBox$. An ontology is called satisfiable (unsatisfiable) iff there exists (does not exist) such a model. In $\mathcal{ALC}$, reasoning tasks, i.e. the derivation of logical consequences, can be reduced to satisfiability checking of ontologies [13, 14].

Table 2. Truth table for 4-valued connectives

α	f	f	f	f	t	t	t	t	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\bot}$	$\ddot{\bot}$	$\ddot{\bot}$	$\ddot{\bot}$
β	f	t	$\ddot{\top}$	$\ddot{\bot}$	f	t	$\ddot{\top}$	$\ddot{\bot}$	f	t	$\ddot{\top}$	$\ddot{\bot}$	f	t	$\ddot{\top}$	$\ddot{\bot}$
$\neg\alpha$	t	t	t	t	f	f	f	f	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\top}$	$\ddot{\bot}$	$\ddot{\bot}$	$\ddot{\bot}$	$\ddot{\bot}$
$\alpha \wedge \beta$	f	f	f	f	f	t	$\ddot{\top}$	$\ddot{\bot}$	f	$\ddot{\top}$	$\ddot{\top}$	f	f	$\ddot{\bot}$	f	$\ddot{\bot}$
$\alpha \vee \beta$	f	t	$\ddot{\top}$	$\ddot{\bot}$	t	t	t	t	$\ddot{\top}$	t	$\ddot{\top}$	t	$\ddot{\bot}$	t	t	$\ddot{\bot}$
$\alpha \mapsto \beta$	t	t	t	t	f	t	$\ddot{\top}$	$\ddot{\bot}$	$\ddot{\top}$	t	$\ddot{\top}$	t	$\ddot{\bot}$	t	t	$\ddot{\bot}$
$\alpha \supset \beta$	t	t	t	t	f	t	$\ddot{\top}$	$\ddot{\bot}$	f	t	$\ddot{\top}$	$\ddot{\bot}$	t	t	t	t
$\alpha \rightarrow \beta$	t	t	t	t	f	t	f	$\ddot{\bot}$	f	t	$\ddot{\top}$	$\ddot{\bot}$	$\ddot{\bot}$	t	$\ddot{\bot}$	t

2.2 Four-valued Logic

The major studies of four-valued logics have been carried out in the setting of propositional logic. We will very briefly review the preliminaries which set the state for the four-valued version of $\mathcal{ALC}$ which we will present later in Section 3.

The idea of four-valued logic is based on the idea of having four truth values, instead of the classical two. The four truth values stand for *true, false, unknown* (or *undefined*) and *both* (or *overdefined, contradictory*). We use the symbols $t, f, \ddot{\bot}, \ddot{\top}$, respectively, for these truth values, and the set of these four truth values is denoted by FOUR. The truth value $\ddot{\top}$ stands for contradictory information, hence four-valued logic lends itself to dealing with inconsistent knowledge. The value $\ddot{\top}$ thus can be understood to stand for *true and false*, while $\ddot{\bot}$ stands for *neither true nor false*, i.e. for the absence of any information about truth or falsity.

Syntactically, four-valued logic is very similar to classical logic. Care has to be taken, however, in defining meaningful notions of implication, as there are several ways to do this. Indeed, there are three major notions of implication in the literature, all of which we will employ in our approach. The logical connectives we allow are thus negation $\neg$, disjunction $\vee$, conjunction $\wedge$, material implication $\mapsto$, internal implication $\supset$, and strong implication $\rightarrow$. We will discuss them in detail later on as the presence of all three implications is crucial for our approach.

Interpretations for four-valued formulae (i.e. 4-interpretations) are obviously mappings from formulae to (the set of four) truth values, respecting the truth tables for the logical connectives, as detailed in Table 2. 4-models are defined in the obvious way, as follows, where t and $\ddot{\top}$ are the designated truth values.

Definition 1 *Let I be a 4-interpretation, let Σ be a theory (i.e. set of formulae) and let φ be a formula in four-valed logic. Then I is a 4-model of φ if and only if $I(\varphi) \in \{t, \ddot{\top}\}$. I is a 4-model of Σ if and only if I is a 4-model of each formula in Σ. Σ four-valued entails φ, written $\Sigma \models_4 \varphi$, if and only if every 4-model of Σ is a 4-model of φ.*

Proposition 1. *We note the following general properties.*

- *The language* $L = \{\neg, \vee, \wedge, \supset, \ddot{\bot}, \ddot{\top}\}$ *is functional complete for the set* FOUR *of truth values, i.e. every function from* FOUR^n *to* FOUR *is representable by some formula in L [10, Theorem 12].*
- *Any formula containing only connectives from* $\{\neg, \vee, \wedge, \supset\}$ *always has a four-valued model.*

Some general remarks about the different notions of implication are in order. They are the major notions of implication used in the literature, and are discussed in detail in [10, 15]. The basic rationales behind them are the following: Material implication can be defined by means of negation and disjunction as known from classical logic. However, it does not satisfy Modus Ponens or the deduction theorem, and is thus of limited use as an *implication* in the intuitive sense. Internal implication satisfies Modus Ponens and the deduction theorem, but cannot be defined by means of other connectives. Furthermore, internal implication does not satisfy contraposition. Strong implication is stronger than internal implication, in that it additionally satisfies contraposition. Indeed, an alternative view on the truth tables for the implication connectives is as follows.

$$\varphi \mapsto \psi \text{ is definable as } \neg\varphi \vee \psi. \qquad \text{(Material Implication)}$$

$$\varphi \supset \psi \quad \text{evaluates to} \quad \begin{cases} \psi & \text{if } \varphi \in \{t, \ddot{\top}\} \\ t & \text{if } \varphi \in \{f, \ddot{\bot}\} \end{cases} \qquad \text{(Internal Implication)}$$

$$\varphi \rightarrow \psi \text{ is definable as } (\varphi \supset \psi) \wedge (\neg\psi \supset \neg\varphi) \qquad \text{(Strong Implication)}$$

Further properties of the implication connectives are summarised in the following proposition (as shown in [10, Corollary 9] and [15]).

Proposition 1 *The following hold, where* Γ *is a theory and* ψ, ϕ *are formulae.*

- *Internal implication is not definable in terms of the connectives* $\neg, \vee, \wedge$.
- $\Gamma, \psi \models_4 \phi$ *iff* $\Gamma \models_4 \psi \supset \phi$.
- *If* $\Gamma \models_4 \psi$ *and* $\Gamma \models_4 \psi \supset \phi$ *then* $\Gamma \models_4 \phi$.
- $\psi \rightarrow \phi$ *implies that* $\neg\phi \rightarrow \neg\psi$.

Apart from the formal properties of the different notions of implication, it is obviously important to consider their intuitive meaning and their usefulness for knowledge base modelling. We will discuss this in detail in the next section.

3 The Four-valued Description Logic $\mathcal{ALC}4$

We describe the syntax and semantics of our four-valued description logic $\mathcal{ALC}4$. The approach is fairly standard apart from the fact that we allow the simultaneous use of all three notions of implication introduced before. We will thus devote significant space to a detailed discussion of the intuitions behind these different implications.

Syntactically, $\mathcal{ALC}4$ hardly differs from $\mathcal{ALC}$. Complex concepts and assertions are defined in exactly the same way. For class inclusion, however, the question arises how to interpret the underlying implication connective in the four-valued setting. We thus allow three kinds of class inclusions, corresponding to the three implication connectives we

have discussed. They are as follows, and correspond to material implication, implicit implication, and strong implication, respectively:

$$C \mapsto D \text{ (material inclusion axiom)},$$
$$C \sqsubset D \text{ (internal inclusion axiom)},$$
$$C \rightarrow D \text{ (strong inclusion axiom)}.$$

Semantically, interpretations map individuals to elements of the domain of the interpretation, as usual. For concepts, however, we need to make modifications to the notion of interpretation in order to allow for reasoning with inconsistencies.

Intuitively, in four-valued logic we need to consider four situations which can occur in terms of containment of an individual in a concept: (1) we know it is contained, (2) we know it is not contained, (3) we have no knowledge whether or not the individual is contained, (4) we have contradictory information, namely that the individual is both contained in the concept and not contained in the concept. There are several equivalent ways how this intuition can be formalised, one of which is described in the following.

For a given domain Δ^I and a concept C, an interpretation over Δ^I assigns to C a pair $\langle P, N\rangle$ of (not necessarily disjoint) subsets of Δ^I. Intuitively, P is the set of elements known to belong to the extension of C, while N is the set of elements known to be not contained in the extension of C. For simplicity of notation, we define functions $\text{proj}^+(\cdot)$ and $\text{proj}^-(\cdot)$ by

$$\text{proj}^+\langle P, N\rangle = P \qquad \text{and} \qquad \text{proj}^-\langle P, N\rangle = N.$$

Formally, a four-valued interpretation is a pair $I = (\Delta^I, \cdot^I)$ with Δ^I as domain, where $\cdot^I$ is a function assigning elements of Δ^I to individuals, and subsets of $(\Delta^I)^2$ to concepts, such that the conditions in Table 3 are satisfied. Note that the conditions in Table 3 for role restrictions are designed in such a way that the logical equivalences $\neg(\forall R.C) = \exists R.(\neg C)$ and $\neg(\exists R.C) = \forall R.(\neg C)$ are retained – this is the most convenient way for us for handling role restrictions, as it will allows for a straightforward translation from $\mathcal{ALC}4$ to classical $\mathcal{ALC}$. Note also that for roles we actually require only the positive part of the extension – we nevertheless require interpretations to assign pairs of sets to roles, which is a technical formality to retain consistency of notation with possible extensions to more expressive description logics (see [7]).

Obviously, under the constraints $P \cap N = \emptyset$ and $P \cup N = \Delta$, four-valued interpretations become just standard two-valued interpretations.

The correspondences between truth values from FOUR and concept extensions, between truth values from FOUR and role relations are the obvious one: For instances $a, b \in \Delta^I$, concept name C, and role name R we have

- $C^I(a) = t$, iff $a^I \in \text{proj}^+(C^I)$ and $a^I \notin \text{proj}^-(C^I)$,
- $C^I(a) = f$, iff $a^I \notin \text{proj}^+(C^I)$ and $a^I \in \text{proj}^-(C^I)$,
- $C^I(a) = \ddot{\top}$, iff $a^I \in \text{proj}^+(C^I)$ and $a^I \in \text{proj}^-(C^I)$,
- $C^I(a) = \ddot{\bot}$, iff $a^I \notin \text{proj}^+(C^I)$ and $a^I \notin \text{proj}^-(C^I)$;
- $R^I(a, b) = \ddot{\top}$, iff $(a^I, b^I) \in proj^+(R^I)$ and $(a^I, b^I) \in proj^-(R^I)$,
- $R^I(a, b) = f$, iff $(a^I, b^I) \notin proj^+(R^I)$ and $(a^I, b^I) \in proj^-(R^I)$,
- $R^I(a, b) = t$, iff $(a^I, b^I) \in proj^+(R^I)$ and $(a^I, b^I) \notin proj^-(R^I)$,

Table 3. Semantics of $\mathcal{ALC}4$ Concepts

Constructor Syntax	Semantics
A	$A^I = \langle P, N\rangle$, where $P, N \subseteq \Delta^I$
R	$R^I = \langle P_1 \times P_2, N_1 \times N_2\rangle$, where $P_1 \times P_2, N_1 \times N_2 \subseteq \Delta^I \times \Delta^I$
o	$o^I \in \Delta^I$
$\top$	$\langle \Delta^I, \emptyset\rangle$
$\bot$	$\langle \emptyset, \Delta^I\rangle$
$C_1 \sqcap C_2$	$\langle P_1 \cap P_2, N_1 \cup N_2\rangle$, if $C_i^I = \langle P_i, N_i\rangle$ for $i = 1, 2$
$C_1 \sqcup C_2$	$\langle P_1 \cup P_2, N_1 \cap N_2\rangle$, if $C_i^I = \langle P_i, N_i\rangle$ for $i = 1, 2$
$\neg C$	$(\neg C)^I = \langle N, P\rangle$, if $C^I = \langle P, N\rangle$
$\exists R.C$	$\langle \{x \mid \exists y, (x, y) \in \text{proj}^+(R^I)$ and $y \in \text{proj}^+(C^I)\}$, $\{x \mid \forall y, (x, y) \in \text{proj}^+(R^I)$ implies $y \in \text{proj}^-(C^I)\}\rangle$
$\forall R.C$	$\langle \{x \mid \forall y, (x, y) \in \text{proj}^+(R^I)$ implies $y \in \text{proj}^+(C^I)\}$, $\{x \mid \exists y, (x, y) \in \text{proj}^+(R^I)$ and $y \in \text{proj}^-(C^I)\}\rangle$

– $R^I(a, b) = \ddot{\bot}$, iff $(a^I, b^I) \notin proj^+(R^I)$ and $(a^I, b^I) \notin proj^-(R^I)$.

When defining the semantics as we just did, we ensure that a number of useful equivalences from classical logic hold, as follows.

Proposition 2 *Let C, D be concepts, and $\top, \bot$ are top concept and empty concept in DLs, respectively. For any four-valued interpretation I,*

$$(C \sqcap \top)^I = C^I, (C \sqcup \top)^I = \top^I,$$

$$(C \sqcap \bot)^I = \bot^I, (C \sqcup \bot)^I = C^I.$$

Proof. For any given interpretation $I = (\Delta^I, \cdot^I)$, $\top^I = \langle \Delta^I, \emptyset\rangle, \bot^I = \langle \emptyset, \Delta^I\rangle$. Without loss of generality suppose $C^I = \langle P, N\rangle$. By table 3:

$$\begin{aligned}
(C \sqcap \top)^I &= \langle P \cap \Delta^I, N \cup \emptyset\rangle = \langle P, N\rangle = C^I \\
(C \sqcup \top)^I &= \langle P \cup \Delta^I, N \cap \emptyset\rangle = \langle \Delta^I, \emptyset\rangle = \top^I \\
(C \sqcap \bot)^I &= \langle P \cap \emptyset, N \cup \Delta^I\rangle = \langle \emptyset, \Delta^I\rangle = \bot^I \\
(C \sqcup \bot)^I &= \langle P \cup \emptyset, N \cap \Delta^I\rangle = \langle P, N\rangle = C^I.
\end{aligned}$$

□

Proposition 3 *Let C, D be concepts, R be an object role name or a datatype role name. For any four-valued interpretation I,*

$$\begin{aligned}
&(\neg\neg C)^I = C^I, (\neg\top)^I = \bot^I, (\neg\bot)^I = \top^I, \\
&(\neg(C \sqcup D))^I = (\neg C \sqcap \neg D)^I, (\neg(C \sqcap D))^I = (\neg C \sqcup \neg D)^I, \\
&(\neg(\forall R.C))^I = (\exists R.\neg C)^I, (\neg(\exists R.C))^I = (\forall R.\neg C)^I.
\end{aligned}$$

Table 4. Semantics of inclusion axioms in $\mathcal{ALC}4$

Axiom Name	Syntax	Semantics
material inclusion	$C_1 \mapsto C_2$	$\Delta^I \setminus \text{proj}^-(C_1^I) \subseteq \text{proj}^+(C_2^I)$
internal inclusion	$C_1 \sqsubset C_2$	$\text{proj}^+(C_1^I) \subseteq \text{proj}^+(C_2^I)$
strong inclusion	$C_1 \rightarrow C_2$	$\text{proj}^+(C_1^I) \subseteq \text{proj}^+(C_2^I)$ and $\text{proj}^-(C_2^I) \subseteq \text{proj}^-(C_1^I)$
individual assertion	$C(a)$	$a^I \in \text{proj}^+(C^I)$

Proof. For any interpretation $I = (\Delta^I, \cdot^I)$,$\top^I = \langle \Delta^I, \emptyset \rangle, \bot^I = \langle \emptyset, \Delta^I \rangle$. Without loss of generality suppose $C^I = \langle P, N \rangle, D^I = \langle P', N' \rangle$. By table 3, the first three formulae hold obviously. Since

$$(\neg(C \sqcup D))^I = \neg\langle P \cup P', N \cap N' \rangle = \langle N \cap N', P \cup P' \rangle = (\neg C \sqcap \neg D)^I$$

So $(\neg(C \sqcup D))^I = (\neg C \sqcap \neg D)^I$ holds. $(\neg(C \sqcap D))^I = (\neg C \sqcup \neg D)^I$ follows in the same way.

Note that $proj^+(C^I) = proj^-((\neg C)^I) = P, proj^-(C^I) = proj^+((\neg C)^I) = N$. By definition 3:

$$\begin{aligned}
(\neg(\forall R.C))^I &= \neg\langle \{x \mid \forall y, (x,y) \in proj^+(R^I) \Rightarrow y \in proj^+(C^I)\}, \\
&\quad \{x \mid \exists y, (x,y) \in proj^+(R^I) \wedge y \in proj^-(C^I)\}\rangle \\
&= \langle \{x \mid \exists y, (x,y) \in proj^+(R^I) \wedge y \in proj^-(C^I)\}, \\
&\quad \{x \mid \forall y, (x,y) \in proj^+(R^I) \Rightarrow y \in proj^+(C^I)\}\rangle \\
&= \langle \{x \mid \exists y, (x,y) \in proj^+(R^I) \wedge y \in proj^+((\neg C)^I)\}, \\
&\quad \{x \mid \forall y, (x,y) \in proj^+(R^I) \Rightarrow y \in proj^-((\neg C)^I)\}\rangle \\
&= (\exists R.\neg C)^I
\end{aligned}$$

Therefore, $(\neg(\forall R.C))^I = (\exists R.\neg C)^I$. Similarly, $(\neg(\exists R.C))^I = (\forall R.\neg C)^I$ holds. □

We now come to the semantics of the three different types of inclusion axioms. It is formally defined in Table 4 (together with the semantics of concept assertions). We say that a four-valued interpretation I satisfies a four-valued ontology $\mathcal{O}$ (i.e. is a model of it) iff it satisfies each assertion and each inclusion axiom in $\mathcal{O}$. An ontology $\mathcal{O}$ is satisfiable (unsatisfiable) iff there exists (does not exist) such a model.

With the formal definitions out of the way, it remains to address the intuitions underlying the different inclusion axioms. These intuitions are evidenced by the formal properties of the underlying implications as discussed in Section 2.2 as well as the behaviour of the implications in practice. We actually foresee a possible workflow for handling inconsistent ontologies, as follows. In a first step, inclusion axioms are classified into the three types of four-valued inclusion axioms available. Then four-valued reasoning is performed based on the classification, in order to arrive at a meaningful 4-valued conclusion. The question, how such a classification can be performed, will not

be addressed in this paper. It constitutes a seperate substantial piece of work which is under investigation by the authors. A combination of automated detection and a user-interaction process may be the most workable solution, where the user-interaction process may be guided by the intuitive explanations which we will now give for the three types of inclusion. An example which displays the effects of the different inclusions is given in Section 6.

Strong inclusion respects the deduction theorem and contraposition reasoning. In a paraconsistent context, it is thus the inclusion to be used for universal truth, such as Square $\rightarrow$ FourEdged.

Internal inclusion propagates contradictory information forward, but not backward as it does not allow for contraposition reasoning. It can thus be characterized as a *brave* way of handling inconsistency. It should be used whenever it is important to infer the consequent even if the antecedent may be contradictory. To give an example, consider a robot fault diagnosis system and an axiom stating that oil leakage is indicative of a robot malfunction. Obviously, it is important to check on a possible malfunction even in case there is contradictory information about an oil leakage. In a paraconsistent context, the axiom is thus best modeled by means of internal inclusion, i.e. as OilLeakage $\sqsubset$ RobotMalfunction.

Material inclusion is *cautious* in the sense that contradictory information is not propagated. The intuition behind material inclusion becomes apparent by studying the truth table for material implication: $a \mapsto b$ indicates that the only way for b to be *not true* (i.e. to be f or $\ddot{\bot}$) is if a is false (i.e. it is f or $\ddot{\top}$). This kind of modelling becomes important if an inclusion has to be second-guessed e.g. after a merging of knowledge bases. Consider, for example, an ontology about marathon runs containing the axiom Healthy $\sqsubseteq$ ParticipatesInMarathon which is supposed to say that somebody (i.e. a person who has signed up for a run) participates in a marathon if he checkes out to be healthy. The axiom is reasonable if the domain is for the management of marathon participants' data only. Now imagine that this ontology is merged with other sports knowledge bases, e.g. a boxing domain. It is wrong to infer that every healthy boxer will participate in the marathon, so the original axiom will likely lead to contradictions. We propose to handle this kind of information by modelling the axiom as material inclusion, i.e. as Healthy $\mapsto$ ParticipatesInMarathon, which will indeed not infer participation from a positive health status. However, the weak form of contraposition reasoning featured by material inclusion results in the following situation: If an individual is not known to be contained in ParticipatesInMarathon, then it is known to be not Healthy, resulting in a possible contradiction on health status while avoiding contradiction in terms of marathon participation, which may be preferred in the domain. Material inclusion may thus propagate contradictory information backwards (to the antecedent), while internal inclusion may propagate contradictory information forward (to the consequent).

4 Transforming $\mathcal{ALC}4$ to $\mathcal{ALC}$

It is a pleasing property of $\mathcal{ALC}4$, that it can be translated easily into classical $\mathcal{ALC}$, such that paraconsistent reasoning can be simulated using standard $\mathcal{ALC}$ reasoning algorithms. We briefly present the translation, a preliminary report has appeared in [7].[2]

For any given concept C, its transformation $\overline{C}$ is the concept obtained from C by the following inductively defined transformation.

- If $C = A$ for A an atomic concept, then $\overline{C} = A^+$, where A^+ is a new concept;
- If $C = \neg A$ for A an atomic concept, then $\overline{C} = A^-$, where A^- is a new concept;
- If $C = \top$, then $\overline{C} = \top$;
- If $C = \bot$, then $\overline{C} = \bot$;
- If $C = E \sqcap D$ for concepts D, E, then $\overline{C} = \overline{E} \sqcap \overline{D}$;
- If $C = E \sqcup D$ for concepts D, E, then $\overline{C} = \overline{E} \sqcup \overline{D}$;
- If $C = \exists R.D$ for D a concept and R is a role, then $\overline{C} = \exists R.\overline{D}$;
- If $C = \forall R.D$ for D a concept and R is a role, then $\overline{C} = \forall R.\overline{D}$;
- If $C = \neg\neg D$ for a concept D, then $\overline{C} = \overline{D}$;
- If $C = \neg(E \sqcap D)$ for concepts D, E, then $\overline{C} = \overline{\neg E} \sqcup \overline{\neg D}$;
- If $C = \neg(E \sqcup D)$ for concepts D, E, then $\overline{C} = \overline{\neg E} \sqcap \overline{\neg D}$;
- If $C = \neg(\exists R.D)$ for D a concept and R is a role, then $\overline{C} = \forall R.\overline{\neg D}$;
- If $C = \neg(\forall R.D)$ for D a concept and R is a role, then $\overline{C} = \exists R.\overline{\neg D}$;

Based on this, axioms are transformed as follows, where C_1, C_2 are concepts.

- $\overline{C_1 \mapsto C_2} = \neg\overline{\neg C_1} \sqsubseteq \overline{C}_2$
- $\overline{C_1 \sqsubset C_2} = \overline{C_1} \sqsubseteq \overline{C}_2$;
- $\overline{C_1 \rightarrow C_2} = \{\overline{C_1} \sqsubseteq \overline{C}_2, \overline{\neg C_2} \sqsubseteq \overline{\neg C_1}\}$.
- $\overline{C(a)} = \overline{C}(a), \overline{R}(a,b) = R(a,b)$, where a, b are individuals, C a concept, R a role.

We note that the transformation algorithm is linear in the size of the ontology. The following theorem shows that paraconsistent reasoning can indeed be simulated on standard reasoners by means of the transformation just given. This implies that paraconsistent reasoning in our paradigm is not more expensive than classical reasoning.

For any ontology $\mathcal{O}$, $\overline{\mathcal{O}}$ is defined as the set $\{\overline{\alpha} \mid \alpha \text{ is an axiom of } \mathcal{O}\}$.

Theorem 4 *For any ontology $\mathcal{O}$ we have $\mathcal{O} \models_4 \alpha$ if and only if $\overline{\mathcal{O}} \models_2 \overline{\alpha}$, where $\models_2$ is the entailment in classical $\mathcal{ALC}$.*

In the rest of this section we prove theorem 4. For this, we first need some notations and definitions.

Definition 2 *(Decomposability) The four-valued semantics of $\mathcal{ALC}4$ can be decomposed into two-valued semantics of $\mathcal{ALC}$, iff for any concept C and object role R in an $\mathcal{ALC}4$ ontology $\mathcal{O}$, there are two concepts C_1,C_2 and two object roles R_1,R_2*

[2] In [7], it was actually spelled out for $\mathcal{SHOIN}$.

in a two-valued $\mathcal{ALC}$ ontology $\mathcal{O}'$ such that for any four-valued interpretation I of $\mathcal{O}$, there's a two-valued interpretation I' of $\mathcal{O}'$, such that

$$C^I = \langle P, N\rangle \ \textit{iff}\ C_1^{I'} = P, C_2^{I'} = N.$$
$$R^I = \langle P_1 \times P_2, N_1 \times N_2\rangle \ \textit{iff}\ R_1^{I'} = P_1 \times P_2, R_2^{I'} = \Delta^{I'} \times \Delta^{I'} \setminus N_1 \times N_2.$$

where P, N, P_1, P_2, N_1 and N_2 are subsets of the domain of I.

The decomposability of $\mathcal{ALC}4$ means that the four-valued semantics of concept C and role R can be divided into the two-valued semantics of two $\mathcal{ALC}$ concepts C_1, C_2 and roles R_1, R_2. This is an essential property of four-valued DLs such that theorem 4 holds.

To eliminate the symbol confusion between the original and the transformed languages, we denote the original language as $\mathcal{L}$, for which $\mathcal{L} = \{C, R, a \mid C$ is a concept name, R is a role name, a is an individual$\}$. $\mathcal{A}(\mathcal{L})$ is the atomic concept set of $\mathcal{L}$. And the transformed language is denoted as $\overline{\mathcal{L}} = \{\overline{C}, \overline{\neg C}, R^+, R^=, \overline{a} \mid C, R, a \in \mathcal{L}, \overline{C}, \overline{\neg C}$ are the concept transformations of C and $\neg C$ respectively, $R^+, R^=$ are two new roles for R. $\overline{a}$ is the renamed name of individual $a\}$. Especially, for atom A, we denote A^+ and A^- for atomic concept transformations of A and $\neg A$, respectively.

Definition 3 *(Classical Induced Interpretation) Let $I = (\Delta^I, \cdot^I)$ be an interpretation of $\mathcal{ALC}4$, and $\overline{\mathcal{O}}$ be the classical induced ontology of $\mathcal{O}$. $I's$* classical induced interpretation *$\overline{I} = (\Delta^{\overline{I}}, \cdot^{\overline{I}})$ is defined as follows:*

- *I and $\overline{I}$ have the same domain, i.e. $\Delta^{\overline{I}} = \Delta^I$;*
- *I and $\overline{I}$ interpret instance names in the same way, i.e. $\overline{a}^{\overline{I}} = a^I$;*
- *For any atomic concept A, if $A^I = \langle P, Q\rangle$, then $(A^+)^{\overline{I}} = P, (A^-)^{\overline{I}} = Q$;*
- *For any object or datatype role R, if $R^I = \langle P_1 \times P_2, N_1 \times N_2\rangle$, then $(R^+)^{\overline{I}} = P_1 \times P_2$, and $(R^=)^{\overline{I}} = \Delta^{\overline{I}} \times \Delta^{\overline{I}} \setminus N_1 \times N_2$.*
- *The semantics of complex concepts are obtained in the standard way.*

Definition 4 *(Four-valued Induced Interpretation) Let $\overline{I}$ be the interpretation of an $\mathcal{ALC}$ ontology $\mathcal{O}$, $\overline{I}$'s* four-valued induced interpretation *$I = (\Delta^I, \cdot^I)$ is defined as follows:*

- *I and $\overline{I}$ have the same domain, i.e. $\Delta^{\overline{I}} = \Delta^I$;*
- *I and $\overline{I}$ interpret instance names in the same way, i.e. $\overline{a}^{\overline{I}} = a^I$;*
- *For any primitive concept A, if $(A^+)^{\overline{I}} = P, (A^-)^{\overline{I}} = Q$, then $A^I = \langle P, Q\rangle$;*
- *For any object and datatype role R, if $(R^+)^{\overline{I}} = P_1 \times P_2$, and $(R^=)^{\overline{I}} = Q_1 \times Q_2$, then $R^I = \langle P_1 \times P_2, \Delta^I \times \Delta^I \setminus Q_1 \times Q_2\rangle$.*
- *The semantics of complex concepts are obtained according to table 3.*

Now we prove the decomposability of $\mathcal{ALC}4$.

Lemma 5 *$\mathcal{ALC}4$ can be decomposed to two-valued semantics of $\mathcal{ALC}$.*

Proof. Let $\mathcal{O}$ be an $\mathcal{ALC}4$ ontology and C be a concept. For any interpretation I, we prove by induction on concept structure that $C^I = \langle P, N\rangle$ iff $\overline{C}^{\overline{I}} = P, \overline{\neg C}^{\overline{I}} = N$, where $\overline{I}$ is the Classical Induced Interpretation of I.

Case: C is an atomic concept A is easy by definition 4, 3.
Case: $C = \neg D$. $\overline{C} = \overline{\neg D}, \overline{\neg C} = \overline{D}$,

- Suppose $C^I = \langle P, N\rangle$. Then $D^I = \langle N, P\rangle$. By induction assumption, we know $\overline{D}^{\overline{I}} = N, \overline{\neg D}^{\overline{I}} = P$. That is $\overline{\neg C}^{\overline{I}} = N, \overline{C}^{\overline{I}} = P$.
- Whereas, suppose $\overline{C}^{\overline{I}} = P, \overline{\neg C}^{\overline{I}} = N$. Then $\overline{D}^{\overline{I}} = N, \overline{\neg D}^{\overline{I}} = P$ By induction assumption, we know $D^I = \langle N, P\rangle$. Through the semantics of negation, we know $C^I = \langle P, N\rangle$.

Case: $C = D \sqcup E$. $\overline{C} = \overline{D} \sqcup \overline{E}$, and $\overline{\neg C} = \overline{\neg D} \sqcap \overline{\neg E}$,

- Suppose $C^I = \langle P, N\rangle$,$D^I = \langle P_1, N_1\rangle, E^I = \langle P_2, N_2\rangle$. Then $P_1 \cup P_2 = P, N_1 \cap N_2 = N$. By induction assumption, we know $\overline{D}^{\overline{I}} = P_1, \overline{\neg D}^{\overline{I}} = N_1, \overline{E}^{\overline{I}} = P_2$, and $\overline{\neg E}^{\overline{I}} = N_2$. therefore $\overline{C}^{\overline{I}} = \overline{D}^{\overline{I}} \cup \overline{E}^{\overline{I}} = P_1 \cup P_2 = P, \overline{\neg C}^{\overline{I}} = \overline{\neg D}^{\overline{I}} \cap \overline{\neg E}^{\overline{I}} = N_1 \cap N_2 = N$.
- Whereas, suppose $\overline{C}^{\overline{I}} = P, \overline{\neg C}^{\overline{I}} = N, \overline{D}^{\overline{I}} = P', \overline{\neg D}^{\overline{I}} = N'$, and $\overline{E}^{\overline{I}} = P'', \overline{\neg E}^{\overline{I}} = N''$. By the definition of semantics, $P = P' \cup P'', N = N' \cap N''$. By induction assumption, we know $D^I = \langle P', N'\rangle, E^I = \langle P'', N''\rangle$. Therefore $C^I = \langle P' \cup P'', N' \cap N''\rangle = \langle P, N\rangle$ by definition of semantics of $\mathcal{ALC}4$.

Case: $C = D \sqcap E$. the proposition holds likewise.
Case: $C = \forall R.D$. $\overline{C} = \forall R.\overline{D}$ and $\overline{\neg C} = \exists R.\overline{\neg D}$,

- Suppose $C^I = \langle P, N\rangle$,$D^I = \langle P_1, N_1\rangle$. By semantics definition, we know $P = \{x \mid \forall y, R(x, y) \Rightarrow y \in proj^+(D^I)\}, N = \{x \mid \exists y, R(x, y) \wedge y \in proj^-(D^I)\}$. By induction assumption, we know $\overline{D}^{\overline{I}} = P_1$ and $\overline{\neg D}^{\overline{I}} = N_1$. Therefore, $N_1 = proj^-(D^I)$. (Note that $P_1 = proj^+(D^I)$)

$$\begin{aligned}
\overline{C}^{\overline{I}} &= (\forall R.\overline{D})^{\overline{I}} = \{x \mid \forall y, R(x, y) \Rightarrow y \in (\overline{D})^{\overline{I}}\} \\
&= \{x \mid \forall y, R(x, y) \Rightarrow y \in P_1\} = P, \\
\overline{\neg C}^{\overline{I}} &= (\exists R.\overline{\neg D})^{\overline{I}} = \{x \mid \exists y, R(x, y) \wedge y \in (\overline{\neg D})^{\overline{I}}\} \\
&= \{x \mid \exists y, R(x, y) \wedge y \in N_1\} = N.
\end{aligned}$$

- Whereas, suppose $\overline{C}^{\overline{I}} = P, \overline{\neg C}^{\overline{I}} = N, \overline{D}^{\overline{I}} = P', \overline{\neg D}^{\overline{I}} = N'$. By the definition of semantics,

$$\begin{aligned}
P = \overline{C}^{\overline{I}} &= (\forall R.\overline{D})^{\overline{I}} = \{x \mid \forall y, R(x, y) \Rightarrow y \in P'\}, \\
N = \overline{\neg C}^{\overline{I}} &= (\exists R.\overline{\neg D})^{\overline{I}} = \{x \mid \exists y, R(x, y) \wedge y \in N'\}.
\end{aligned}$$

By induction assumption, we know $D^I = \langle P', N' \rangle$. Furthermore, by the semantics of $\mathcal{ALC}4$, we know

$$C^I = \langle \{x \mid \forall y, R(x,y) \Rightarrow y \in P'\}, \{x \mid \exists y, R(x,y) \wedge y \in N'\} \rangle = \langle P, N \rangle$$

Case: $C = \exists R.D$. the lemma holds likewise.

In all, let $C_1 = \overline{C}, C_2 = \overline{\neg C}$, we see that for any concept C, $C^I = \langle P, N \rangle$ iff $C_1^{\overline{I}} = P$ and $C_2^{\overline{I}} = N$.

For any role R, from definition 4 and 3, we can see that $R^I = \langle P_1 \times P_2, N_1 \times N_2 \rangle$ iff $\overline{R_1}^{\overline{I}} = P_1 \times P_2, \overline{R_2}^{\overline{I}} = \Delta^I \times \Delta^I \setminus N_1 \times N_2$. □

Now we turn to prove theorem 4.

Proof. (OF THEOREM 4) (Necessity) For any interpretation I of $\mathcal{O}$, let the interpretation $\overline{I}$ be I's classical induced interpretation. According to the relationship between $\overline{\mathcal{O}}$ and $\mathcal{O}$, for any $\overline{\mathcal{O}}$'s inclusion of the form $\neg\overline{\neg C} \sqsubseteq \overline{D} \in \overline{\mathcal{O}}, C \mapsto D \in \mathcal{O}$. Suppose $C^I = \langle P_1, N_1 \rangle, D^I = \langle P_2, N_2 \rangle$. By lemma 5, $\overline{\neg C}^{\overline{I}} = N_1, \overline{D}^{\overline{I}} = P_2$. Then, $(\neg\overline{\neg C})^{\overline{I}} = \Delta^{\overline{I}} \setminus N_1 = \Delta^I \setminus N_1$. I satisfies $C \mapsto D$, so $\Delta^I \setminus N_1 \subseteq P_2$. Therefore, $(\neg\overline{\neg C})^{\overline{I}} \subseteq \overline{D}^{\overline{I}}$, that is, $\overline{I}$ satisfies $\neg\overline{\neg C} \sqsubseteq \overline{D}$.

For any $\overline{\mathcal{O}}$'s inclusion of the form $\overline{C} \sqsubseteq \overline{D} \in \overline{\mathcal{O}}$, $\overline{\neg C} \sqsubseteq \overline{\neg D} \notin \overline{\mathcal{O}}, C \sqsubset D \in \mathcal{O}$. Suppose $C^I = \langle P_1, N_1 \rangle, D^I = \langle P_2, N_2 \rangle$. By lemma 5, $\overline{C}^{\overline{I}} = P_1, \overline{D}^{\overline{I}} = P_2$. I satisfies $C \sqsubset D$. Therefore, $P_1 = proj^+(C^I) \subseteq proj^+(D^I) = P_2$, that is $\overline{I}$ satisfies $\overline{C} \sqsubseteq \overline{D}$.

For any $\overline{O}$'s inclusion pair of the form $\{\overline{C} \sqsubseteq \overline{D}, \overline{\neg D} \sqsubseteq \overline{\neg C}\} \subseteq \overline{\mathcal{O}}, C \rightarrow D \in \mathcal{O}$. Suppose $C^I = \langle P_1, N_1 \rangle, D^I = \langle P_2, N_2 \rangle$. By lemma 5, $\overline{C}^{\overline{I}} = P_1, \overline{D}^{\overline{I}} = P_2, \overline{\neg C}^{\overline{I}} = N_1, \overline{\neg D}^{\overline{I}} = N_2$. I satisfies $C \rightarrow D$. Therefore, $P_1 = proj^+(C^I) \subseteq proj^+(D^I) = P_2, N_2 = proj^-(D^I) \subseteq proj^-(C^I) = N_1$, that is $\overline{I}$ satisfies $\{\overline{C} \sqsubseteq \overline{D}, \overline{\neg D} \sqsubseteq \overline{\neg C}\}$.

For any assertion of the form $\overline{a}{:}\overline{C}$. $a{:}C$ belongs to the $\mathcal{O}$. Suppose $C^I = \langle P, N \rangle$, $a^I = \delta_0 \in \Delta^I$, then $(\overline{C})^{\overline{I}} = P, \overline{a}^{\overline{I}} = \delta_0$. Because I satisfies $a{:}C$, $\delta_0 \in P$, that is $\overline{I}$ satisfies $\overline{a}{:}\overline{C}$. It is similar for $R(a,b)$ assertion.

(sufficiency) For any interpretation $\overline{I} = (\Delta^{\overline{I}}, \cdot^{\overline{I}})$ of $\overline{O}$, let I be the four-valued semantics of $\overline{I}$. Similarly, we can prove the proposition to be right. □

5 Resolution-Based Reasoning with $\mathcal{ALC}4$

There exist two fundamentally different approaches to reasoning with description logics. The first, historic approach is based on an adaptation of the tableaux method from first-order predicate logic (see [13]), and is implemented in most current reasoners. The second approach is based on resolution and has been realised in the KAON2 reasoner [12]. While the first method is best invoked by a preprocessing as spelled out in Section 4, the question arises whether anything can be gained by adjusting the resolution-based approach to a paraconsistent setting, thus avoiding the preprocessing step. In this section, we will spell out the adjustments necessary for doing this. We basically follow [12, Chapter 4], and indeed we assume that the reader is familiar with the KAON2-approach as we will not spell everything out in detail.

We first note that resolution relies heavily on the *tertium non datur*, and thus does not lend itself easily to a paraconsistent setting. In particular, resolution cannot be based on the negation present in paraconsistent logics, as in this case $A \vee B$ and $\neg A \vee C$ does not imply $B \vee C$. We thus start by introducing a second kind of negation, called the *strong negation*, denoted by $\sim$. In order to avoid confusion, we will refer to the standard negation as *paraconsistend negation*.

Definition 5 *The* strong negation $\sim$ *on* $\{\langle P, N\rangle \mid P, N \subseteq \Delta\}$ *is defined by*

$$\sim\langle P, N\rangle = \langle \Delta \setminus P, \Delta \setminus N\rangle.$$

The intuition behind strong negation is to reverse both the information of being true and of being false. Notice that we do not extend our four-valued DLs to have the strong negation as a concept constructor. We rather use it only to provide a resolution-based decision procedure for four-valued DLs.

Proposition 6 *For strong negation, the following hold for all concepts C, D and roles R. For any four-valued interpretation I,*

$$(\sim\sim C)^I = C^I, (\sim\top)^I = \bot^I, (\sim\bot)^I = \top^I, (\neg\sim C)^I = (\sim\neg C)^I$$
$$(\sim(C \sqcup D))^I = (\sim C \sqcap \sim D)^I, (\sim(C \sqcap D))^I = (\sim C \sqcup \sim D)^I,$$
$$(\sim(\forall R.C))^I = (\exists R.\sim C)^I, (\sim(\exists R.C))^I = (\forall R.\sim C)^I.$$

Proof. It is obvious by the definition 5.

A second issue which we have to address when adjusting resolution to the paraconsistent setting, is to obtain a representation of internal implication (i.e. of internal inclusion) in terms of clauses. We have already remarked in Section 2.2 that internal implication cannot be represented by means of the connectives conjunction, disjunction and paraconsistent negation. However, with strong negation a representation of $C \sqsubset D$ as $\sim C \sqcup D$ is possible. The representation is actually not logically equivalent, but it is equisatisfiable as shown in the following proposition, which suffices for setting up a resolution procedure.

Proposition 7 *For any four-valued interpretation I, $(C \sqsubset D)^I \in \{t, \top\}$ if and only if $(\sim C \sqcup D)^I \in \{t, \top\}$, that is $\top \sqsubset (\sim C \sqcup D)$ always holds.*

Proof. It is obvious by truth table.

We indeed have the following theorem.

Theorem 8 *Let O be a four-valued $\mathcal{ALC}4$ ontology, C, D be concepts, I be an interpretation and ι be a new individual not occurring in O. Then the following hold.*

1. *$(C \sqsubset D)^I \in \{t, \ddot{\top}\}$ if and only if $(\sim C \sqcup D)^I \in \{t, \ddot{\top}\}$.*
2. *$O \models_4 C(a)$ if and only if $O \cup \{\sim C(a)\}$ is four-valued unsatisfiable.*
3. *$O \models_4 C \mapsto D$ if and only if $O \cup \{\sim(\neg C \sqcup D)(\iota)\}$ is four-valued unsatisfiable.*
4. *$O \models_4 C \sqsubset D$ if and only if $O \cup \{(C \sqcap \sim D)(\iota)\}$ is four-valued unsatisfiable.*

5. *$O \models_4 C \rightarrow D$ if and only if $O \cup \{(C \sqcap \sim D)(\iota), (\neg D \sqcap \sim\neg C)(\iota)\}$ is four-valued unsatisfiable.*

Proof. – (Necessity of claim 1) For any four-valued model I of O, $C^I(a) \in \{t, \ddot{\top}\}$. So by the definition 5, $\sim C^I(a) \in \{f, \ddot{\bot}\}$. So $O \cup \{\sim C(a)\}$ is four-value unsatisfiable. (Sufficiency of claim 1) Suppose $O \cup \{\sim C(a)\}$ is four-value unsatisfiable. So for any four-valued model I of O, $\sim C^I(a) \in \{f, \ddot{\bot}\}$, which ensures that $C^I(a) \in \{t, \ddot{\top}\}$. That is, $O \models_4 C(a)$.

– (Necessity of claim 2) Suppose there's a four-valued model I of O which is a model of $\sim (\neg C \sqcup D)(\iota)$, that is, $\sim (\neg C \sqcup D)(\iota) \in \{t, \ddot{\top}\}$. By definition 5, $(\neg C \sqcup D)(\iota) \in \{f, \ddot{\bot}\}$. By table 3, $\neg C(\iota) \in \{f, \ddot{\bot}\}$ and $D(\iota) \in \{f, \ddot{\bot}\}$, so $C(\iota) \in \{t, \ddot{\bot}\}$, i.e, $\iota^I \notin proj^-(C^I)$ and $\iota^I \in proj^I(D)$. A contradiction with the fact that $O \models_4 C \mapsto D$. (Sufficiency of claim 2) For any four-valued model of O, $\sim (\neg C \sqcup D)(\iota) \in \{f, \ddot{\bot}\}$. By table 3 and definition 5, $\iota^I \in proj^-(C^I)$ or $\iota^I \in proj^+(D^I)$, that is, $C \mapsto D$ is satisfiable in I.

– (Necessity of claim 3) Suppose there's a four-valued model I of O which is a model of $(C \sqcap \sim D)(\iota)$, that is, $C(\iota) \in \{t, \ddot{\top}\}$ and $\sim D(\iota) \in \{t, \ddot{\top}\}$. By table 5, $D(\iota) \in \{f, \ddot{\bot}\}$. A contradiction with the fact that $O \models_4 C \mapsto D$. (Sufficiency of claim 2) For any four-valued model of O, $(C \sqcap \sim D)(\iota) \in \{f, \ddot{\bot}\}$. By definition 5 and table 3, $\iota^I \notin proj^+(C^I)$ or $\iota^I \in proj^+(D^I)$, that is, $C \sqsubset D$ is satisfiable in I.

– Claim 4 can be proved in the similar way.

5.1 Translating $\mathcal{ALC}4$ into Clauses

Resolution-based calculi operate on sets of clauses in normal form, so we introduce next clausal forms for $\mathcal{ALC}4$ expressions. We were inspired by [16]. We first define a negation normal form for $\mathcal{ALC}4$ concepts, which we call quasi-NNF.

Definition 6 *A concept C is a quasi-atom, if it is an atomic concept, or in form $\neg A$ where A is an atomic concept. A concept C is a quasi-literal, if it is a quasi-atomic concept, or in form $\sim L$ where L is an quasi-atomic concept. A concept C is in quasi-NNF, if the strong negation $\sim$ occurs only in front of quasi-literals.*

To give an example, let A, B, and C be atomic concepts. Then $(A \vee \sim\neg B) \sqcup \forall R.(\sim C)$ is in quasi-NNF. By propositions 3 and 6, the following is obvious.

Theorem 9 *All $\mathcal{ALC}4$ concepts can be transformed into equivalent expressions in quasi-NNF.*

We next define the Definitorial form of $\mathcal{ALC}4$ concepts, which is a technicality to control the size of clauses, and was also used in [12]. If C is a concept, then we set

$$\text{Def}(C) = \begin{cases} \{C\} & \text{if } C \text{ is a literal concept,} \\ \{\sim Q \sqcup C|_p\} \cup \text{Def}(C[Q]|_p) & \text{if } p \text{ is eligible for replacement in } C, \end{cases}$$

where $C|_p$ is the position p in concept C, as defined in [17, 12].

As an example, we have $\text{Def}(A \sqcap \exists R.(A \sqcap B)) = \{A \sqcap \exists R.Q, {\sim}Q \sqcup (A \sqcap B)\}$. Note that ${\sim}Q \sqcup (A \sqcap B)$ can be interpreted as internal inclusion $A \sqsubset B$, which allows us to use Q as a new name for $(A \sqcap B)$ in $\exists R.(A \sqcap B))$.

The following proposition is as expected.

Proposition 10 *For an $\mathcal{ALC}4$ concept C in quasi-NNF, $C(x)$ has information to be true for all individuals x if and only if all concepts $D_i(x)$ with $D_i \in$ Def(C) have the information to be true. Formally, $\{\top \sqsubset C\}$ is four-valued satisfiable iff $\{\top \sqsubset$ Def$(D_i) \mid D_i \in$ Def$(C)\}$ is.*

Proof. The proof is by induction on the number of recursive invocations of Def. The induction base is trivial, so we consider the induction step, which replaces the subconcept $C|_p$ in C with Q, resulting in $D = C[Q]_p$. For the necessity direction, let I be a model satisfying $\top \sqsubset C$. Obviously, an interpretation I', obtained by extending I with $Q^I = (C|_p)^I$, satisfies $\{\top \sqsubset D, \top \sqsubset\sim Q \sqcup C|_p\}$. For the sufficient direction, it is obvious by proposition 7.

We next translate the concepts into predicate logic. This is done by the standard translation as e.g. spelled out in [12] in terms of the function π_y – we just have to provide for the strong negation. This is done by allowing the strong negation to occur in the predicate logic formulae as well, and by translating strong negation in the same way as paraconsistent negation. We make one exception, namely for unversal restriction, where we set $\pi_y(\forall R.C, x) = \forall y.({\sim}R(x,y) \sqcup C(y))$.

Following the above transformations step by step, any $\mathcal{ALC}4$ concept can be translated into a set of first order predicate logic clauses (with strong negation) in polynomial size of the original concepts. We give an example.

Example 1 *The concept $\neg({\sim}A \sqcap \exists R.(\forall S.C))$ is translated as follows.*

$$\begin{aligned}
& \neg({\sim}A \sqcap \exists R.(\forall S.C)) \\
\textit{in quasi-NNF:} \quad & {\sim}\neg A \sqcup \forall R.(\exists S.\neg C) \\
\textit{in definitorial form:} \quad & \{{\sim}\neg A \sqcup \forall R.Q, {\sim}Q \sqcup \exists S.\neg C\} \\
\textit{in predicate logic:} \quad & \{{\sim}\neg A(x) \vee {\sim}R(x,y) \vee Q(y), {\sim}Q(x) \vee S(x, f(x)), \\
& \qquad\qquad {\sim}Q(x) \vee \neg C(f(x))\}
\end{aligned}$$

The obtained predicate logic formulae (with strong negation) can now be translated into clauses in the standard way, i.e. by first casting them into Skolem form, and then into conjunctive normal form by exhaustive application of well-known logical equivalences (see e.g. [17]), which are adjusted for strong negation in the obvious straightforward way.

If C is a concept (where the additional use of strong negation is allowed), then we denote by $\text{Cls}(C)$ the set of clauses which is obtained by the just mentioned transformation. These clauses are predicate logic formulae (with strong negation).

We finally translate an $\mathcal{ALC}4$ knowledge base KB into a set $\Xi(KB)$ of predicate logic clauses (with strong negation), as follows. The knowledge base $\Xi(KB)$ is the smallest set satisfying the following conditions:

Table 5. Clause Types

1	$\bigvee(\sim)(\neg)C_i(x) \vee R(x, f(x))$
2	$\bigvee(\sim)(\neg)C_i(x) \vee (\sim)(\neg)D(f(x))$
3	$\bigvee(\sim)(\neg)C_i(x)$
4	$\bigvee(\sim)(\neg)C_i(x) \vee R(x, y) \vee (\sim)(\neg)D(y)$
5	$(\sim)(\neg)C(a)$
6	$(\sim)(\neg)R(a, b)$

- For each $ABox$ axiom α in $ABox$, Cls$(\alpha) \subseteq \Xi(KB)$
- For each $TBox$ axiom $C \mapsto D$ in $TBox$, Cls$(\neg C \sqcup D) \subseteq \Xi(KB)$
- For each $TBox$ axiom $C \sqsubset D$ in $TBox$, Cls$(\sim C \sqcup D) \subseteq \Xi(KB)$
- For each $TBox$ axiom $C \rightarrow D$ in $TBox$, Cls$(\sim C \sqcup D, \sim\neg D \sqcup \neg C) \subseteq \Xi(KB)$

Theorem 11 *Let KB be an $\mathcal{ALC}4$ ontology. Then, the following hold.*

- *KB is satisfiable iff $\Xi(KB)$ is satisfiable.*
- *$\Xi(KB)$ can be computed in time polynomial in $|KB|$.*
- *Each clause in $\Xi(KB)$ is of one of the syntactic forms listed in Table 5. We refer to these clauses as* 4-valued clauses*.*

Proof. The first claim holds by definition 10. The other two can be proven in the same way of lemma 4.3.4 in [12].

5.2 Ordered Resolution with Selection Function $O4_{DL}$ for $\mathcal{ALC}4$

The KAON2 approach for $\mathcal{ALC}$ is based on a modification of the original resolution calculus, known as ordered resolution (see [12] for the necessary preliminaries). We will now define the corresponding notions of clause ordering and of selection function, which we require for this. We assume in the sequel that all 4-valued clauses are of the form described in theorem 11.

Given any fixed ordering $\succ$ on ground quasi-atoms which is total and well-founded, we can obtain an *ordering on sets of clauses* as follows.

1. Extend $\succ$ to an ordering $\succ_L$ on ground literals by setting $\sim A \succ_L A$ for any A, and $[\sim]A \succ_L [\sim]B$, if $A \succ B$.
2. Extend $\succ_L$ to an ordering $\succ_C$ on ground clauses by setting $\succ_C = (\succ_L)_{mul}$ to be the multi-set extension of $\succ_L$ (see [12] for formal definition). The intuition is that $C_1 \succ_C C_2$ iff the maximal quasi-literal in clause C_1 is greater then that in clause C_2 w.r.t. $\succ_L$.

By a slight abuse of notation, we use $\succ$ also for $\succ_L$ and $\succ_C$ where the meaning is clear from the context.

Example 2 *For* $\neg A \succ A \succ B \succ \neg B \succ C \succ \neg C \succ D$*, we obtain*

$$[\sim]\neg A \succ [\sim]A \succ [\sim]B \succ [\sim]\neg[\sim]B \succ [\sim]C \succ [\sim]\neg C \succ [\sim]D \qquad \textit{and}$$

$$\neg C \vee D \prec C \vee \sim D \prec \neg B \vee C \prec \sim A \vee \neg B \prec \neg A \vee B \prec \sim\neg A \vee D.$$

By a *selection function* we mean a mapping S that assigns to each clause C a (possibly empty) multiset $S(C)$ of literals with the prefix $\sim$ in C. For example, both $\{\sim\neg A\}$ and $\{\sim\neg A, \sim D\}$ can be selected in clause $\sim\neg A \vee \sim D \vee B \vee \neg C$.

An ordered resolution step with selection function can now be described by the inference rule

$$\frac{C \vee A \quad D \vee \sim B}{C\sigma \vee D\sigma}$$

where

- $\sigma = MGU(A, B)$ is the most general unifier of the quasi-atoms A, B, and C, D are quasi-clauses.
- $A\sigma$ is strictly maximal in $C\sigma \vee A\sigma$, and no literal is selected in $C\sigma \vee A\sigma$;
- $\sim B\sigma$ is either selected in $D\sigma \vee \sim B\sigma$, or it is maximal in $D\sigma \vee \sim B\sigma$ and no literal is selected in $D\sigma \vee \sim B\sigma$.

The corresponding ordered factorization rule is

$$\frac{C \vee A \vee B}{(C \vee A)\sigma},$$

where $\sigma = MGU(A, B)$ and $A\sigma$ is maximal in $C\sigma \vee A\sigma$ and nothing is selected in C.

The following theorem shows soundness and completeness of our resulting ordered resolution calculus $O4_{DL}$.

Theorem 12 *Let* N *be an* $\mathcal{ALC}4$ *knowledge base. Then* $\Xi(N) \vdash_{O4_{DL}} \square$ *iff* N *is four-valued unsatisfiable.*

Before we give the proof of this theorem, we first give some lemmas, whose corollary is the theorem. We begin with the propositional 4-valued clauses.

W.l.o.g., an $\mathcal{ALC}4$ clause D can be written in the form of $D = \bigvee_i A_i \vee \bigvee_j (\neg B_j) \vee \bigvee_k (\sim C_k)$, where A_i, B_j, and C_k are ground concepts or ground roles. D is true in a 4-valued interpretation I means that either there's A_{i_0} or $\neg B_{j_0}$ which is true in I, or there's C_{k_0} which is false in I.

Let $N, \prec, S$ be given. We define sets $I_C = \bigcup_{D \prec C} \Delta_D$ and Δ_C for all ground clauses C over the given signature inductively over

$$\Delta_C = \begin{cases} A & \text{if } C \in N, C = C' \vee A, A \succ C', C \text{ is false in } I_C, \text{ and} \\ & \quad \text{there's no selected literals in } C \\ \neg A & \text{if } C \in N, C = C' \vee \neg A, \neg A \succ C', C \text{ is false in } I_C, \text{ and} \\ & \quad \text{there's no selected literals in } C. \\ \emptyset & \text{if otherwise} \end{cases}$$

We say that C produces A ($\neg A$), if $\Delta_C = \{A\}$ ($\{\neg A\}$). And C is a *productive clause*, if C produces A or $\neg A$. The *candidate model* for N (wrt. $\prec$ and S) is given as $I_N^{\succ} :=$

$\bigcup_C \Delta_C$. The *counterexamples* in a candidate model $I_N^{\succ}$ are the clauses in N which are false in $I_N^{\succ}$. The *minimal counterexample* is the minimal element in the counterexample set w.r.t. given clause ordering.

Proposition 13 *Productive clauses are true in candidate model.*

The following theorem shows an important *Reduction Property* of $O4_{DL}$.

Lemma 1. *(Reduction Property Lemma) Let N be a set of clauses not containing the empty clause. Let C be the minimal counterexample in N for $I_N^{\succ}$. Then there exists an inference in $O4_{DL}$ from C, such that*

- *C is its main premise side and the premises that are productive clauses.*
- *its conclusion is a smaller counterexample for $I_N^{\succ}$ than C.*

Proof. Since N does not contain the empty clause, C must contain at least one quasi-literal. We have known that productive clause is true in candidate model $I_N^{\succ}$, so C is not a productive clause. So either C contains at least one selected quasi-literal, say $\sim A'$, or there's no selected quasi-literal and its maximal quasi-atom A occurs negatively, that is $\sim A \in C$. Thus C can be written as $\sim B \vee C'$, where $B \in I_N^{\succ}$, C' is false in $I_N^{\succ}$, and either B is a selected quasi-literal in C or else nothing is selected in C and B is maximal in C. Let $D = D' \vee B$ be the clause that produces B, so that no literal is selected in D, B is maximal in D, and D' is false in I_D (actually, there're theorems guarantee that D' is remain false in $I_N^{\succ}$). Therefore, ordered resolution with the productive clause D as the side premise and C as the main premise yields a resolvent $C' \vee D'$ that is strictly smaller than C, and is false in $I_N^{\succ}$. In short, ordered resolution yields a smaller counterexample than C.

Reduction property grantees the following *Model Existence Lemma* hold in propositional case:

Lemma 2. *(Model Existence Lemma) Let $\succ$ be a clause ordering. Suppose the propositional clause set N is saturated wrt. $\mathcal{O}4_{DL}$, and suppose that $\square \notin N$. Then the candidate model I is a model of N, that is $I \models_4 N$.*

Proof. Notice that if $I \not\models_4 N$, then there's a minimal counterexample C which does not equal to $\square$. By the reduction property, there's another strictly smaller counterexample in N since N is saturated. A contradiction with the suppose that C is minimal counterexample.

We finish the proof of completeness (Theorem 12) in the proposition case. Next we have to refer to a 4-valued lifting lemma to prove the completeness of $O4_{DL}$.

Lemma 3. *(Lifting Lemma) Let M be a set of clauses and K is the ground instances of clauses in M, denoted $K = G(M)$. If*

$$\frac{C_1, ..., C_n \quad C_0}{C}$$

is an inference in $O4_{DL}(K)$, then there exists clauses C'_i in M, a clauses C', and a ground substitution σ such that

$$\frac{C'_1, ..., C'_n \quad C'_0}{C'}$$

is an inference in $O4_{DL}(M), C_i = C'_i\sigma$, and $C = C'\sigma$.

An analogous lifting lemma holds for factorization.

Proof. It can be proved in the similar way as the classical lifting lemma.

Lemma 4. *Let N be a set of general clauses saturated under $\mathcal{O}4_{DL}$, i.e. $\mathcal{O}4_{DL}(N) \subseteq N$. Then there exists a selection function S_0 such that and G(N) is also saturated, i.e., $\mathcal{O}4_{DL}(G(N)) \subseteq G(N)$.*

Proof. This lemma holds by the lifting lemma.

Now we turn to theorem 12.

Proof. (OF THEOREM 12) By lemma 2, 4 and the lifting lemma, we can prove completeness theorem 12. Moreover, it is easy to check that $O4_{DL}$ is sound.

Decision Procedure for $\mathcal{ALC}4$ We can now select suitable parameters in order to arrive at a decision procedure based on $O4_{DL}$. This can be done as follows.

- The literal ordering $\succ$ is defined such that $R(x, f(x)) \succ \sim C(x)$ and $D(f(x)) \succ \sim C(x)$, for all function symbols f, and predicates R, C, and D.
- The selection function selects every binary literal which is preceeded by $\sim$.

The following theorem then holds.

Theorem 14 *For an $\mathcal{ALC}4$ knowledge base KB, saturating $\Xi(KB)$ by $\mathcal{O}4_{DL}$ decides satisfiability of KB and runs in time exponential in $|KB|$.*

Proof. This can be proved in the same way of similar theorem in [12].

6 Implementation

ParOWL[3] is a prototype implementation of our paraconsistent reasoning approach. It realises the algorithm from Section 4 by means of a command line tool based on KAON2. In order to allow the use of standard OWL syntax, the tool expects four input files as parameters, all of which are standard OWL documents. In the first file, class inclusion is interpreted as material inclusion, in the second as internal inclusion, and in the third as strong inclusion. The fourth file is expected to contain an ABox. ParOWL outputs an OWL file which contains the translation.

[3] http://logic.aifb.uni-karlsruhe.de/wiki/Paraconsistent_reasoning

Table 6. Paraconsistent reasoning examples

Ontology	Bird	FlyAnimal	Penguin	notBird	notFlyAnimal	notPenguin
O_1	$\emptyset$	$\emptyset$	tweety	$\emptyset$	$\emptyset$	$\emptyset$
O_2	tweety	tweety	tweety	$\emptyset$	tweety	$\emptyset$
O_3	tweety	tweety	tweety	tweety	tweety	tweety
O_4	tweety	$\emptyset$	tweety	$\emptyset$	tweety	$\emptyset$

To display the usage of the different inclusion axioms in ParOWL, consider the following example ontology O, which consists of the axioms Bird $\sqsubseteq$ FlyAnimal, Penguin $\sqsubseteq$ Bird, Penguin $\sqsubseteq \neg$FlyAnimal, and Penguin(tweety), which is obviously inconsistent. We compare the following different four-valued ontologies which can be derived from O: For O_1 all inclusions are material, for O_2 all inclusions are internal, for O_3 all inclusions are strong. For O_4, the first inclusion is material, the second is internal, and the third is strong. Table 6 shows the extensions of the concepts in these ontologies as computed with ParOWL. The desired result obviously is O_4, and indeed the choice of inclusion axioms in this case follows the intuitions laid out in Section 3.

7 Conclusions and Further Work

We have motivated and formally described an approach for paraconsistent reasoning with ontologies, which is based on the simultaneous use of different kinds of paraconsistent inclusion. We have provided guidelines for the use of these different inclusions. We have provided algorithms for implementing our approach and presented a publicly available tool which realises it.

Concerning the two algorithms provided in Sections 4 and 5, it appears at hindsight that the direct approach from Section 5 does not appear to provide a more efficient solution. It is rather apparent that all the benefits from the KAON2 system – like the ability to handle large ABoxes – can also be achieved by invoking KAON2 after employing the transformation algorithm from Section 4 in a preprocessing manner using ParOWL.

In the literature, there are basically two other approaches to four-valued description logics, namely [4] and [5]. Our approach differes from theirs in two important aspects. The first is that we allow for simultaneous usage of different inclusions. The second is that it is possible to translate our logic into standard description logic such that established reasoners can be used. We thus benefit directly from the highly optimised systems currently available. The logics in [4, 5] have neither of these features.

Obviously, much work remains to be done to make our approach fit for practice. Besides the obvious task of providing a better implementation than just a prototype, we also have to address in more detail the question, which kinds of paraconsistent inclusion are to be chosen when translating paraconsistent ontologies to standard ontologies. We envision a combination of system recommendations with user interactions. Alternatively, inclusions could be weakened gradually from strong inclusion to weaker versions – probably involving even further notions of inclusion – until a reasonable answer to a query is found. These issues are currently under investigation by the authors.

References

1. Peter F. Patel-Schneider and Horrocks Ian. OWL web ontology language semantics and abstract syntax. *W3C Recommendation*, 10 February, 2004.
2. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 355–362. Morgan Kaufmann, 2003.
3. Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. A framework for handling inconsistency in changing ontologies. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 353–367. Springer, 2005.
4. Peter F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38:319–351, 1989.
5. Umberto Straccia. A sequent calculus for reasoning in four-valued description logics. In Didier Galmiche, editor, *TABLEAUX*, volume 1227 of *Lecture Notes in Computer Science*, pages 343–357. Springer, 1997.
6. Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 454–459. Professional Book Center, 2005.
7. Yue Ma, Zuoquan Lin, and Zhangang Lin. Inferring with inconsistent OWL DL ontology: a multi-valued logic approach. *Current Trends in Database Technology - EDBT 2006 Workshops*, 2006.
8. N. D. Belnap. A useful four-valued logic. *Modern uses of multiple-valued logics*, pages 7–73, 1977.
9. N. D. Belnap. How a computer should think. *Comtemporary Aspects of Philosophy: Proceedings of the Oxford International Symposium*, pages 30–56, 1977.
10. Ofer Arieli and Arnon Avron. The value of the four values. *Artif. Intell.*, 102(1):97–141, 1998.
11. Sylvie Coste-Marquis and Pierre Marquis. On the complexity of paraconsistent inference relations. In Leopoldo E. Bertossi, Anthony Hunter, and Torsten Schaub, editors, *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 151–190. Springer, 2005.
12. Boris Motik. Reasoning in description logics using resolution and deductive databases. *PhD theis, University Karlsruhe, Germany*, 2006.
13. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
14. Ian Horrocks and Peter F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *J. Web Sem.*, 1(4):345–357, 2004.
15. Ofer Arieli and Arnon Avron. Reasoning with logical bilattices. *Journal of Logic, Language and Information*, 5(1):25–63, 1996.
16. Norihiro Kamide. Foundations of paraconsistent resolution. *Fundamenta Informaticae*, 71, Number 4:419–441, 2006.
17. M. Fitting. *First-Order Logic and Automated Theorem Proving, 2nd Edition*. Texts in Computer Science. Springer, 1996.