An Argumentative Semantics for Paraconsistent Reasoning in Description Logic $\mathcal{ALC}$

Xiaowang Zhang[1,2], Zhihu Zhang[1] and Zuoquan Lin[1]

[1] School of Mathematical Sciences, Peking University, Bejing 100871, China
[2] School of Mathematical Sciences, Anhui University, Hefei 230039, China
{zxw,zhzhang,lzq}@is.pku.edu.cn

Abstract. It is well known that description logics cannot tolerate the incomplete or inconsistent data. Recently, inconsistency handling in description logics becomes more and more important. In this paper, we present an argumentative semantics for paraconsistent reasoning in inconsistent ontologies. An argumentative framework based on argument trees is provided to model argumentation in description logic $\mathcal{ALC}$. Furthermore, two basic problems, namely, satisfiability of concepts and query entailment problems, are discussed under our argumentative semantics in description logic $\mathcal{ALC}$.

1 Introduction

Description logics (or DLs) are considered as a kind of very important knowledge representation formalism unifying and giving a logical basis to the well known traditions of Frame-based systems, Semantic web and KL-ONE-like languages, Object-Oriented representations, Semantic data models and Type systems. Merging ontologies and ontology evolution may cause the occurrence of the inconsistency. In practical reasoning, it is common to have "too much" information about some situations. According the fact *ex contradictione quodlibet* "$\frac{\alpha,\neg\alpha}{\beta}$" in classical logics, any axiom can be entailed from an inconsistent ontology called *explosive entailment*. In general, conclusions drawn from inconsistent ontologies may be completely meaningless.

In recent years, handling inconsistency becomes more and more important in DLs. The main idea of handling inconsistency is to avoid the explosive entailment and obtain some valuable information from inconsistent ontologies. Roughly speaking, there are two fundamentally different approaches to dealing with inconsistency in DLs. The first is based on the assumption that inconsistencies indicate erroneous data which are to be repaired in order to obtain consistent knowledge bases, e.g., selecting consistent subsets for the reasoning process [1,2,3]. Unfortunately, based on this approach, we may lose useful information so that we might not obtain more believable conclusions from those inconsistent information. Another approach, called *paraconsistent approach*, is to tolerate inconsistency by applying a non-standard reasoning method, such as beyond classical semantics, to obtain meaningful answers [4,5]. In the paraconsistent approach, inconsistencies are treated as a natural phenomenon in realistic data and they are tolerated in reasoning. In general, the idea of paraconsistent approach in DLs is introducing Belnap's four-valued semantics [6] for DLs (see [5]). A more prominent one of them is 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). To a certain extent four-valued semantics can be employed to handle inconsistency, however, the capability of reasoning is rather weaker. For instance, four-valued DLs do not hold three basic inference rules such as *modus ponens* (*MP*), *disjunctive syllogism* (*DS*), *modus tollens* (*MT*) or *intuitive equivalence*. In order to strengthen the ability of paraconsistent reasoning, Zhang et al [7,8] introduce quasi-classical (QC for short) semantics into DLs while tautologies cannot be driven by quasi-classical entailment from some ontology because the QC logics are relevant logics (see [8]). Besides these two approaches, there are some "*hybrid*" approaches to handling inconsistency based on formalisms [9] in other logics. The main idea of these hybrid approaches to handling inconsistency is reasoning with consistent subsets of a knowledge base essentially. However, a problem of those hybrid approaches is generally short of justice enough since inferences are drawn from parts of knowledge bases.

Though there are some different characteristics among those existing approaches, the common problem is that inconsistencies are not analyzed in detail further but either isolated or ignored in reasoning. In philosophy, there is a classical theory called *argumentation theory (or argumentation)*, which embraces the arts and sciences of civil debate, dialogue, conversation and persuasion. Dung researches the fundamental mechanism, humans use in argumentation, and explores ways in computer science (see [10]). Elvang and Hunter [11] have presented an argumentative framework which is different from Dung's argumentative framework to resolve inconsistency in propositional logic. In recent years, a number of logic-based proposals for formalizations of argumentation theory presented in [12,13] show that the approaches based on argumentation theory in logics have more predominant properties in handling inconsistency. Moreover, the approach is increasingly applied in ontologies engineering (see [14]).

In this paper, we introduce argumentation theory into description logic (DL for short) $\mathcal{ALC}$ in order to handle inconsistency. We propose an argumentative framework based on Elvang and Hunter's argumentative framework to model argumentative description logic $\mathcal{ALC}$. The main innovations and contributions of this paper can be summarized as follows:

- defining semantically the notion called *quasi-classical negation* to characterize the semantic reverse of axioms in DLs;
- introducing the argumentation theory for $\mathcal{ALC}$ by defining some basic notions, namely, *argument, undercut, maximal conservative undercut* and *canonical undercut* etc;
- presenting an argumentative framework, which are composed of canonical undercuts, is developed to demonstrate the structure of arguments in ontologies;
- providing an argumentative semantics based on binary argumentation for $\mathcal{ALC}$ and discussing two basic problems, namely, satisfiability of concepts and inference problems based on our argumentative semantics in $\mathcal{ALC}$.

This paper is structured as follows. Section 2 reviews briefly the syntax and semantics of $\mathcal{ALC}$. Section 3 presents argumentation theory for $\mathcal{ALC}$. Section 4 develops argumentative framework. Section 5 proposes argumentative semantics for $\mathcal{ALC}$. Section 6 discusses reasoning problems under argumentative semantics. Section 7 compares our work with four-valued DLs. Section 8 concludes and discusses our future work. Due to the space limitation, proofs are sketched out but they are available in a technical report[1]

2 Preliminaries

In the section, we briefly review syntax and semantics of the DL $\mathcal{ALC}$, but we basically assume that the reader is familiar with DLs. For comprehensive background reading, please refer to [15].

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

Concepts description in $\mathcal{ALC}$ can be formed according to the following syntax rule:

$$C, D \rightarrow A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

Let C, D be concepts, R a role and a, b individuals. In DL $\mathcal{ALC}$, an ontology consists of a set of assertions, called the $ABox$ of the ontology, and a set of inclusion axioms, called the $TBox$ of the ontology. $Assertions$ are of the form $C(a)$ or $R(a, b)$. $Inclusion\ axioms$ are of the form $C \sqsubseteq D$. 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.

Let an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$. The formal definition of the (model-theoretic) semantics of $\mathcal{ALC}$ is given by means of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$which assigns to every an individual an member of $\Delta^{\mathcal{I}}$, to every atomic concept A a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to concept descriptions by the following inductive definitions: $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} = \emptyset$, $(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash A^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y, (x, y) \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$, $(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y, (x, y) \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$.

An interpretation $\mathcal{I}$ *satisfies* a concept axiom $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, denoted by $\mathcal{I} \models C(a)$. An interpretation $\mathcal{I}$ *satisfies* a role axiom $R(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, denoted by $\mathcal{I} \models R(a, b)$. An interpretation $\mathcal{I}$ *satisfies* an inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, denoted by $\mathcal{I} \models C \sqsubseteq D$. An interpretation $\mathcal{I}$ is a $model$ of $\mathcal{A}$ if $\mathcal{I}$ satisfies every assertion $C(a)$ or , $R(a, b) \in \mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$. An interpretation $\mathcal{I}$ is a $model$ of $\mathcal{T}$ if $\mathcal{I}$ satisfies every inclusion axiom $C \sqsubseteq D \in \mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$. An interpretation $\mathcal{I}$ *satisfies* an $\mathcal{ALC}$ ontology $\mathcal{O}$ (i.e. is a $model$ of the ontology) iff it satisfies both the ABox and the TBox of it, denoted by $\mathcal{I} \models \mathcal{O}$.

A TBox $\mathcal{T}$ is *consistent* if there is a model of $\mathcal{T}$. An ABox $\mathcal{A}$ is *consistent* if there is a model of $\mathcal{A}$. An ABox $\mathcal{A}$ is *consistent* w.r.t. $\mathcal{T}$ if there is a model of $\mathcal{I}$ which is a model of $\mathcal{T}$. An ontology $\mathcal{O}$ is *consistent* if there is a model of $\mathcal{A}$ and $\mathcal{T}$, and otherwise *unsatisfiable*. A set of formulae $\mathcal{S}$ $entails$ a formula α, denoted by $\mathcal{S} \models \alpha$, if for every model $\mathcal{I}$ of $\mathcal{S}$ then $\mathcal{I}$ is a model of α. $\{\beta\} \models \alpha$ is sometimes denoted by $\beta \models \alpha$. A set of formulae $\mathcal{S}_1$ entails a set of formulae $\mathcal{S}_2$, denoted by $\mathcal{S}_1 \models \mathcal{S}_2$, if $\mathcal{S} \models \alpha$ for any

[1] `http://www.is.pku.edu.cn/~zxw/publication/TRAALC.pdf`

member α of $\mathcal{S}_2$. An ABox $\mathcal{A}$ *entails* a query α if $\mathcal{S} = \mathcal{A}$, denoted by $\mathcal{A} \models \alpha$. A TBox $\mathcal{T}$ *entails* a query α, denoted by $\mathcal{T} \models \alpha$. Analogously, $\mathcal{O}$ entails α, denoted by $\mathcal{O} \models \alpha$ if $\mathcal{A} \models \alpha$ and $\mathcal{T} \models \alpha$.

3 Argumentation for Description Logic $\mathcal{ALC}$

In this section, we introduce basic definitions of argumentation for DL $\mathcal{ALC}$ and bring some properties.

Let α and β be axioms and $\mathcal{S}$ be a set of axioms in $\mathcal{ALC}$. In DLs, there are two logical connectives "*conjunction*" ($\wedge$) and "*disjunction*" ($\vee$) defined as follows:
(1) $\mathcal{S} \models \alpha \wedge \beta$ iff $\mathcal{S} \models \alpha$ and $\mathcal{S} \models \beta$;
(2) $\mathcal{S} \models \alpha \vee \beta$ iff $\mathcal{S} \models \alpha$ or $\mathcal{S} \models \beta$.

Compared with concept conjunction connective $\wedge$ and disjunction connective $\vee$ connecting with two concepts, logical connectives connect with two axioms. The following lemma shows the relationship between logic connectives and syntax connectives ("$\sqcap$", "$\sqcup$").

Lemma 1 *Given concepts C, D, an individual a and a set of axioms S, then*
(1) $\mathcal{S} \models C(a) \wedge D(a)$ *iff* $\mathcal{S} \models C \sqcap D(a)$*;*
(2) $\mathcal{S} \models C(a) \vee D(a)$ *iff* $\mathcal{S} \models C \sqcup D(a)$.

Proof (skeleton). It easily proves the two properties by the definitions of the syntax connectives and the logical connectives in DLs.

If α, β be axioms in DLs, then $\alpha \wedge \beta$ (or $\alpha \vee \beta$) is called *conjunction of axioms* (or *disjunction of axioms*).

Definition 1 *Given a set of formulae $\mathcal{S}$, concepts C, D, an individual a and two axioms α, β, a negative transformation written by $\sim$ on axioms or disjunction (conjunction) of axioms called* quasi-negation *is defined as follows:*

- $\mathcal{S} \models \sim C(a)$ *iff* $\mathcal{S} \models \neg C(a)$*;*
- $\mathcal{S} \models \sim C \sqsubseteq D$ *iff* $\mathcal{S} \models C \sqcap \neg D(\iota)$ *for some individual ι in $\mathcal{S}$;*
- $\mathcal{S} \models \sim (\alpha \wedge \beta)$ *iff* $\mathcal{S} \models \sim \alpha \vee \sim \beta$*;*
- $\mathcal{S} \models \sim (\alpha \vee \beta)$ *iff* $\mathcal{S} \models \sim \alpha \wedge \sim \beta$.

Axioms, conjunction of axioms, disjunction of axioms and their quasi-negations (QN for short) are called *extended axioms*. In this paper, extended axioms are not members of ontologies but are taken as the consequent of arguments presented in the following definition.

Definition 2 *Let Φ be a set of axioms and α be an extended axiom in $\mathcal{ALC}$. An $argument$ is a pair $\langle \Phi, \alpha \rangle$, denoted by $\mathbf{A} = \langle \Phi, \alpha \rangle$, such that:*
(1) *Φ is consistent;*
(2) $\Phi \models \alpha$*;*
(3) *no $\Phi' \subset \Phi$ satisfies $\Phi' \models \alpha$. We say that $\langle \Phi, \alpha \rangle$ is an $argument$ for α.*
We call α the $consequent$ of the argument and Φ the $support$ of the argument, denoted by $Support(\langle \Phi, \alpha \rangle) = \Phi$ and $Consequence(\langle \Phi, \alpha \rangle) = \alpha$.

Example 1 *Let an ontology* $\mathcal{O} = (\{Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly, Penguin \sqsubseteq \neg Fly, Swallow \sqsubseteq Bird, Swallow \sqsubseteq \forall HasFood.\neg Fish, Penguin \sqsubseteq \exists HasFood.Fish, Swallow \sqsubseteq \neg Penguin\}, \{Penguin(Tweety), \neg Fly(Tweety)\})$.
There are some arguments as follows:

$$
\begin{aligned}
\mathbf{A}_1 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, Fly(Tweety)\rangle \\
\mathbf{A}_2 &= \langle\{\neg Fly(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, \neg Penguin(Tweety)\rangle \\
\mathbf{A}_3 &= \langle\{\neg Fly(Tweety), Penguin(Tweety), Penguin \sqsubseteq Bird\}, Bird \sqcap \neg Fly(Tweety)\rangle \\
\mathbf{A}_4 &= \langle\{\neg Fly(Tweety), Penguin(Tweety), Bird \sqsubseteq Fly\}, Penguin \sqcap \neg Bird(Tweety)\rangle \\
\mathbf{A}_5 &= \langle\{\neg Fly(Tweety)\}, \neg Fly(Tweety)\rangle \\
\mathbf{A}_6 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq \neg Fly\}, \neg Fly(Tweety)\rangle \\
\mathbf{A}_7 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, Penguin \sqcap Fly(Tweety)\rangle \\
\mathbf{A}_8 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq \exists HasFood.Fish\}, \exists HasFood.Fish(Tweety)\rangle \\
\mathbf{A}_9 &= \langle\{Penguin(Tweety), Swallow \sqsubseteq \neg Penguin\}, \neg Swallow(Tweety)\rangle \\
\mathbf{A}_{10} &= \langle\{\neg Fly(Tweety), Bird \sqsubseteq Fly\}, \neg Bird(Tweety)\rangle
\end{aligned}
$$

An argument $\langle\Phi, \alpha\rangle$ is $more\ conservative$ than an argument $\langle\Psi, \beta\rangle$ iff $\Phi \subseteq \Psi$ and $\beta \models \alpha$, written by $\langle\Psi, \beta\rangle \preceq_c \langle\Phi, \alpha\rangle$. If $\alpha \not\models \beta$ then $\langle\Phi, \alpha\rangle$ is called $strictly\ more\ conservative$ than an argument $\langle\Psi, \beta\rangle$, written by $\langle\Psi, \beta\rangle \prec_c \langle\Phi, \alpha\rangle$. In short, we define a pre-order relationship on arguments using conservative relationship. For instance, the argument $\mathbf{A}_4 \prec_c \mathbf{A}_{10}$ in Example 1.

It is clear to conclude that $\mathbf{A} \preceq_c \langle\emptyset, \top\rangle$ for any argument $\mathbf{A}$ because the empty ontology is consistent. In this paper, we mainly consider non-empty finite ontologies.

Definition 3 *An argument* $\langle\Psi, \beta\rangle$ *is a* $defeater$ *of an argument* $\langle\Phi, \alpha\rangle$ *such that* $\beta \models \sim (\phi_1 \wedge \ldots \wedge \phi_n)$ *for some* $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. *An* $undercut$ *for an argument* $\langle\Phi, \alpha\rangle$ *is an argument* $\langle\Psi, \sim (\phi_1 \wedge \ldots \wedge \phi_n)\rangle$ *where* $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$.

In Example 1, the argument $\mathbf{A}_5$ is a defeater of the argument $\mathbf{A}_1$ and $\mathbf{A}_7$; the argument $\mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4$ are undercuts for the argument $\mathbf{A}_1$.

Proposition 1 *If* $\mathbf{A}'$ *is a defeater for* $\mathbf{A}$ *then there exists an undercut* $\mathbf{A}''$ *for* $\mathbf{A}$ *where* $\mathbf{A}' \preceq_c \mathbf{A}''$.

Proof (skeleton). $\mathbf{A}'$ is a defeater for $\mathbf{A} \Rightarrow$ there exists a subset $\{\phi_1, \ldots, \phi_m\}$ of $Support(\mathbf{A})$ such that $Consequnece(\mathbf{A}') \models \sim (\phi_1 \wedge \cdots \wedge \phi_m) \Rightarrow$ there exists a minimal subset Ψ of $Support(\mathbf{A}')$ such that $\Psi \models \sim (\phi_1 \wedge \cdots \wedge \phi_m) \Rightarrow \mathbf{A}'' = \langle\Psi, \sim (\phi_1 \wedge \cdots \wedge \phi_m)\rangle$ is an undercut for $\mathbf{A}$ and $\mathbf{A}' \preceq_c \mathbf{A}''$.

Two arguments $\langle\Phi, \alpha\rangle$ and $\langle\Psi, \beta\rangle$ are $equivalent$ iff Φ is logically equivalent to Ψ and α is logically equivalent to β. Clearly, if two arguments are equivalent then either of each is more conservative than the other or neither is. $\mathbf{A}'$ is a $maximally\ conservative\ defeater$ of $\mathbf{A}$ iff $\mathbf{A}'$ is a defeater of $\mathbf{A}$ for any defeater $\mathbf{A}''$ for $\mathbf{A}$ such that $\mathbf{A}'' \prec_c \mathbf{A}'$. A $maximally\ conservative\ undercut$ of $\langle\Phi, \alpha\rangle$ is defined analogously. An argument $\mathbf{A}'$ is a $canonical\ undercut$ of $\mathbf{A}$ iff $\mathbf{A}'$ is a maximally conservative undercut of $\mathbf{A}$.

In Example 1, the arguments $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$ are canonical undercuts of $\mathbf{A}_1$. The argument $\mathbf{A}_2$ is a canonical undercut of $\mathbf{A}_8$.

Proposition 2 *$\langle \Psi, \sim (\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is a canonical undercut for an argument $\langle \Phi, \alpha\rangle$ iff it is an undercut for $\langle \Phi, \alpha\rangle$ and $\{\phi_1, \ldots, \phi_n\}$ is the enumeration of Φ.*

Proof (skeleton). ($\Rightarrow$) Suppose $\mathbf{A}' = \langle \Psi, \sim (\phi_1 \wedge \cdots \wedge \phi_n)\rangle$ is not a maximal conservative undercut for $\mathbf{A} = \langle \Phi, \alpha\rangle \Rightarrow$ there exists undercut $\mathbf{A}''$ such that $\mathbf{A}' \prec_c \mathbf{A}''$ $\Rightarrow Consequence(\mathbf{A}'') \models \sim (\phi_1 \wedge \cdots \wedge \phi_n) \Rightarrow$ there exists a new axiom ϕ which is different from $\phi_i (1 \leq i \leq n)$ such that $Consequence(\mathbf{A}'') \models \sim (\phi_1 \wedge \cdots \wedge \phi_n \wedge \phi)$ while it conflicts this fact that $\{\phi_1, \ldots, \phi_n\}$ is the canonical enumeration of Φ. ($\Leftarrow$) It follows similarly.

Proposition 3 *Given an argument $\mathbf{A}$, if $\mathbf{A}', \mathbf{A}''$ are different canonical undercuts for $\mathbf{A}$ then the following claims hold: $\mathbf{A}' \not\preceq_c \mathbf{A}''$ and $\mathbf{A}'' \not\preceq_c \mathbf{A}'$; $Consequence(\mathbf{A}') = Consequence(\mathbf{A}'')$.*

Proof (skeleton). The first property follows the definition of canonical undercut which is a maximal conservative undercut. The second property follows Proposition 2 because the canonical enumeration of a set of formulae is single.

Example 2 *Given $\mathcal{O} = (\emptyset, \{C(a), D(a), \neg C(a), \neg D(a)\})$, both the following $\mathbf{A}_{11} = \langle\{\neg C(a)\}, \sim (C \sqcap D)(a)\rangle$ and $\mathbf{A}_{12} = \langle\{\neg D(a)\}, \sim (C \sqcap D)(a)\rangle$ are canonical undercuts for $\mathbf{A}_{13} = \langle\{C(a), D(a)\}, C \sqcap D(a)\rangle$, but neither is more conservative than the other.*

In Example 2, $\mathbf{A}_{11}$ and $\mathbf{A}_{12}$ are different canonical undercuts for $\langle\{C(a), D(a)\}, C \sqcap D(a)\rangle$. $Support(\mathbf{A}_{11})$ is different from $Support(\mathbf{A}_{12})$ while $Consequence(\mathbf{A}_{11})$ is the same as $Consequence(\mathbf{A}_{12})$.

Therefore, by Proposition 3 Item 2, we denote $\langle \Psi, \diamond\rangle$ denotes a canonical undercut of $\langle \Phi, \beta\rangle$. We easily conclude that $\diamond$ is $\sim (\phi_1 \wedge \cdots \wedge \phi_n)$ where $< \phi_1, \ldots, \phi_n >$ is the canonical enumeration for Φ.

Proposition 4 *For each defeater $\mathbf{A}'$ for an argument $\mathbf{A}$, there exists a canonical undercut $\mathbf{A}''$ for $\mathbf{A}$ such that $\mathbf{A}' \preceq_c \mathbf{A}''$.*

Proof (skeleton). There exists a defeater $\mathbf{A}'$ for an argument $\mathbf{A} \Rightarrow$ there exists an undercut $\mathbf{A}''$ for $\mathbf{A}$ such that $\mathbf{A}' \preceq_c \mathbf{A}''$ by Proposition 1 $\Rightarrow$ if $\mathbf{A}''$ is not a maximal conservative undercut for $\mathbf{A}$ then there exists $\mathbf{A}'' \preceq_c \mathbf{A}^3$ for $\mathbf{A}$ till finding a maximal conservative undercut $\mathbf{A}^m$ for $\mathbf{A}$, i.e., $\mathbf{A}^n$ is a canonical undercut for $\mathbf{A}$.

4 Argumentative Framework

Definition 4 *Given an axiom α, an $argument\ tree$ for α is a tree where the nodes are arguments such that*
(1) *the root is an argument for α;*
(2) *for no node $\langle \Phi, \beta\rangle$ with ancestor nodes $\langle \Phi_1, \beta_1\rangle \ldots \langle \Phi_n, \beta_n\rangle$ is Φ a subset of $\Phi_1 \cup \ldots \cup \Phi_n$;*
(3) *the children nodes of a node $\mathbf{A}$ consist of all canonical undercuts for $\mathbf{A}$, which obeys* (2).

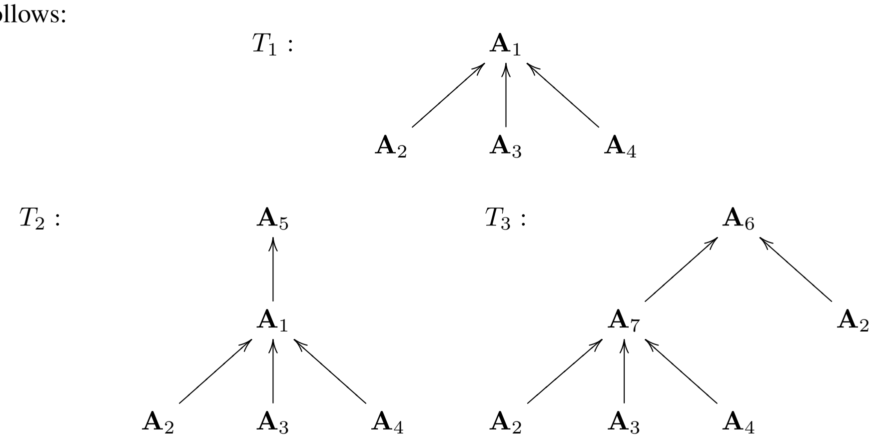
Proposition 5 *Any argument tree of an axiom α in a finite $\mathcal{O}$ is finite*[2].

[2] A tree is finite iff it has a finite number of branches and a finite depth.

Proof (skeleton). Since $\mathcal{O}$ is finite, the number of subsets of $\mathcal{O}$ is finite. In an argument tree, no branch can be infinite by Condition 2 of Definition4. Also, the number of canonical undercuts is finite by its definition. The branching factor in an argument tree is finite by Condition 3 of Definition 4.

Definition 5 *An $argumentative\ framework$ of an axiom α is a pair of sets $\langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{P}$ is the set of argument trees of α and $\mathcal{C}$ is the set of argument trees for $\sim \alpha$.*

In Example 1, we obtain the argumentative framework $\langle \mathcal{P}, \mathcal{C} \rangle$ of $Fly(Tweety)$ as follows:



Here T_1 is an argument tree of $Fly(Tweety)$ and T_2, T_3 are argument trees of $\neg Fly(Tweety)$. So the argumentative framework of $Fly(Tweety)$ is $\langle \langle \{T_1\}, \{T_2, T_3\} \rangle \rangle$.

Proposition 6 *Given an argumentative framework $\langle \mathcal{P}, \mathcal{C} \rangle$, if $\mathcal{O}$ is consistent, then each argument tree in $\mathcal{P}$ has exactly one node and $\mathcal{C}$ is the empty set.*

Proof (skeleton). Given any axiom α, we only show that $\mathcal{C}$ of α is the empty set. Suppose that $\mathcal{C} \neq \emptyset$, i.e., there exists a successful argument tree T of $\sim \alpha$. The root of T is a defeater of α. So there exists a canonical undercut $\mathbf{A}$ for α by Proposition 4. That is to say, there exists an argument tree of α that contains at least two nodes. It conflicts with Proposition 6.

5 Argumentative Semantics

If $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$ are three arguments such that $\mathbf{A}_1$ is undercut by $\mathbf{A}_2$ and $\mathbf{A}_2$ is undercut by $\mathbf{A}_3$ then $\mathbf{A}_3$ is called a *defence* for $\mathbf{A}_1$. We define the "*defend*" relation as the transitive closure of "*being a defence*". An argument tree is said to be *successful* iff every leaf defends the root node. The *categorizer* is a function, denoted by $\mathbf{c}$, from the set of argument trees to $\{0, 1\}$ such that $\mathbf{c}(T) = 1$ iff T is successful. The *categorization* of a set of trees is the collections of their values by categorizer function on them. The *accumulator* of a query α is a function, denoted by $\mathbf{a}$, from categorizations to the set $\{(1, 1), (1, 0), (0, 1), (0, 0)\}$. Let $\langle X, Y \rangle$ be a categorization of argumentative framework of an axiom α, then $\mathbf{a}(\langle X, Y \rangle) = (w(X), w(Y))$ where $w(Z) = 1$ iff $1 \in Z$.

Definition 6 *The* $valuation$ *is a function, denoted by* $\mathbf{v}$*, from a set of axioms to a set* $\{$*Both(B),True(T),False(F),Unknown(U)*$\}$*, defined as follows:*
(1) $\mathbf{v}(\alpha) = B$ *iff* $\mathbf{a}(\langle X, Y \rangle) = (1, 1)$*;*
(2) $\mathbf{v}(\alpha) = T$ *iff* $\mathbf{a}(\langle X, Y \rangle) = (1, 0)$*;*
(3) $\mathbf{v}(\alpha) = F$ *iff* $\mathbf{a}(\langle X, Y \rangle) = (0, 1)$*;*
(4) $\mathbf{v}(\alpha) = U$ *iff* $\mathbf{a}(\langle X, Y \rangle) = (0, 0)$*;*
where $\langle X, Y \rangle$ *is a categorization of argumentative framework of* α *in* $\mathcal{O}$*.*

Definition 7 *Let* $\mathcal{O}$ *be an ontology and* α *be an axiom in* $\mathcal{ALC}$*.* $\mathcal{O}$ *argumentatively* $entails$ α*, denoted by* $\mathcal{O} \models_a \alpha$*, iff there exists a successful argument tree of* α*.*

Theorem 1 *Given an ontology* $\mathcal{O}$ *and an axiom* α*, then the following properties are equivalent:*
(1) $\mathcal{O} \models_a \alpha$*;*
(2) $\mathbf{v}(\alpha) \in \{B, T\}$*;*
(3) *there exists an argument tree* T *of* α *such that* $\mathbf{c}(T) = 1$.

Proof (skeleton). $\mathcal{O} \models_a \alpha \Leftrightarrow$ there exists a successful tree T of α in $\mathcal{O}$ by Definition 7 $\Leftrightarrow \mathbf{c}(T) = 1$ by the definitions of accumulator $\Leftrightarrow \mathbf{v}(\alpha) \in \{B, T\}$ by definition of categorizer and valuation function.

Theorem 2 *Given a consistent ontology* $\mathcal{O}$ *and an axiom* α*, then* $\mathcal{O} \models_a \alpha$ *iff* $\mathcal{O} \models \alpha$*.*

Proof (skeleton). Because $\mathcal{O}$ is consistent, $\mathcal{O} \models \alpha \Leftrightarrow$ the argumentative framework $\langle \mathcal{P}, \mathcal{C} \rangle$ of α whose each argument tree in $\mathcal{P} \neq \emptyset$ has exactly one node and $\mathcal{C}$ is the empty set by Proposition 6 $\Leftrightarrow$ there exists a successful argument tree of α because its contains only one node $\Leftrightarrow \mathcal{O} \models_a \alpha$.

However, an ontology $\mathcal{O}$ is inconsistent, then $\models_a$ is weaker than $\models$. Clearly, $\mathcal{O} \models_a \sim \alpha$ doesn't possibly hold if $\mathcal{O} \not\models_a \alpha$ for any axiom α. $\mathcal{O} \models_a \beta$ is not inferred from $\mathcal{O} \models_a \alpha$ and $\mathcal{O} \models_a \sim \alpha \vee \beta$. Since $\{\alpha, \sim \alpha\} \not\models_a \beta$ for any query β, we conclude the following theorem.

We say description logics with argumentative semantics ***argumentative description logics*** (***ADLs*** for short). It easily shows the following property holds.

Theorem 3 *Argumentative description logic* $\mathcal{ALC}$ *is a paraconsistent logic.*

6 Reasoning in Argumentative Description Logic $\mathcal{ALC}$

In this section, we mainly discuss two basic reasoning problems argumentative satisfiability and queries argumentative entailment problem in argumentative $\mathcal{ALC}$ under argumentative semantics.

Definition 8 *A concept* C *is* $argumentatively\ satisfiable$ *if there exists a successful argument tree* T *of* $C(a)$ *for some individual* a*; and* $argumentatively\ unsatisfiable$ *otherwise. An ontology* $\mathcal{O}$ *is* $argumentatively\ consistent$ *iff each concept in* $\mathcal{O}$ *is argumentatively satisfiable; and* $argumentatively\ inconsistent$ *otherwise.*

The following theorem shows the relationship between satisfiability and argumentative satisfiability of concepts.

Theorem 4 *If a concept C is satisfiable in an ontology $\mathcal{O}$ then the concept C is argumentatively satisfiable in $\mathcal{O}$.*

Proof (skeleton). C is satisfiable in $\mathcal{O}$ $\Rightarrow$ there exists a model $\mathcal{I}$ of C $\Rightarrow$ by definition of satisfiability of concept in DL $\Rightarrow$ there exists an individual a such that $a \in C^{\mathcal{I}}$ and $a \notin (\neg C)^{\mathcal{I}}$ $\Rightarrow$ there exists an argument tree of $C(a)$ which contains only one node $\Rightarrow$ there exists a successful argument tree of $C(a)$ $\Rightarrow$ C is argumentatively satisfiable in $\mathcal{O}$.

In Example 1, the concept $\neg Fly$ is argumentatively satisfiable and other concepts are not argumentatively satisfiable. and the ontology $\mathcal{O}$ is neither consistent nor argumentatively consistent.

There are two basic query problems, namely instance checking and subsumption in an ontology $\mathcal{O}$ under argumentative semantics.

- *instance checking*: given a concept C and an individual a, a is an instance of concept under argumentative semantics iff $\mathcal{O} \models_a C(a)$.
- *subsumption*: a concept C is subsumed by a concept D under argumentative semantics iff $\mathcal{O} \models_a C \sqsubseteq D$.

Two problems can be taken as argumentative entailment problems because a query is a given axiom. By Theorem 2, two problems of instance checking and subsumption in a consistent ontology is equivalent to the problems under argumentative semantics. When an ontology $\mathcal{O}$ is inconsistent, some meaningful information can be mined after reasoning with queries under argumentative semantics by analyzing and evaluating inconsistency occurring in given ontologies. Compared with "*satisfiability* " of queries in ontologies with classical entailment, argumentative entailment considers "*justifiability*" of queries in ontologies.

In Example 1, Since the accumulator of $Fly(Tweety)$ is $(0,1)$, $\mathbf{v}(\neg Fly(Tweey))$ is "T". Therefore, $\mathcal{O} \models_a \neg Fly(Tweey)$ and $\mathcal{O} \not\models_a Fly(Tweey)$. Though $\mathcal{O} \models Fly(Tweety)$ and $\mathcal{O} \models \neg Fly(Tweety)$, it shows that $\neg Fly(Tweety)$ is "*justifiable*" in $\mathcal{O}$ because the fact $\neg Fly(Tweety)$ has other facts support it, while $Fly(Tweety)$ is "*unjustifiable*". In short, reasoning with argumentative semantics can justify a query by analyzing other facts for and against the query in a given ontology, distinguishing from answering the query by "*true*" or "*false*".

7 Related Work

In this section, we mainly compare argumentative semantics for DLs with four-valued semantics for DLs, $\mathcal{ALC}4$ (see [5]). The similarities and differences between argumentative DLs and four-valued DLs resemble those between argumentative logic and Belnap's four-valued logics respectively.

At first, we consider their similarities: (1) they are paraconsistent DLs; that is to say, they don't hold the *fact ex contradictione quodlibet*; and (2) the range of their valuation function are $\{B, T, F, U\}$. Argumentative entailment $\models_a$ and four-valued entailment $\models_4$ are based on the subset $\{B, T\} \subseteq \{B, T, F, U\}$.

There are many differences between argumentative DLs and four-valued DLs as follows: (1) argumentative DLs directly employs the reasoning system of DLs to implement paraconsistent reasoning compared with four-valued DLs; (2) the scope of reasoning in argumentative DL is consistent subsets of an inconsistent ontology while the scope of reasoning in four-valued DLs is a whole ontology; (3) there are some concepts which are not argumentatively satisfiable while all concepts in four-valued DLs are satisfiable; and (4) argumentation semantics is more outstanding than four-valued semantics when an ontology contains a lot of inconsistent information.

8 Conclusions

The primary aim of this paper is to extend logic-based proposals for argumentation with techniques for argumentative DLs. The most difficult problem of argumentative DLs is that there exist many differences between DLs and other logics. We don't directly define the form of canonical undercuts in DLs by applying analogously the form of them in propositional logic or first-order logic because the connectives $\neg, \sqcap, \sqcup$ in syntax of DLs aren't logical connectives. In this paper, the main innovations in defining the form of canonical undercut are summarized as follows: (1) compared to defining argument for propositional logic or first-order logic in syntax, the notion of argument for description logic is defined in semantics. An argument for DLs contains two part called Support and Consequent where Support logically entails Consequent; (2) two logical connectives $\wedge, \vee$ are used to express the logical relationship between two axioms; and (3) a negation transformation $\sim$ on axioms or conjunctions (disjunctions) of axioms called quasi-negation is introduced to express the counterpart of these axioms or conjunctions (disjunctions) of axioms.

Argumentative framework for $\mathcal{ALC}$ based on argument trees is inherited from Elvang and Hunter's argumentative framework [11]. Binary argumentation is developed under the aim of capturing the simple form of argumentation in our framework because binary argumentation and four-valued logic have the same form of expression. Argumentative entailments based on the set of values $\{B, T\}$ holds the paraconsistent property. In the end of the paper, we discuss serval basic problems that argumentative satisfiability of concepts, argumentative consistency of ontologies, instance checking and subsumption under argumentative semantics. Though argumentative semantics can handle inconsistency in $\mathcal{ALC}$, the main work of implementing paraconsistent reasoning with argumentative semantics is searching arguments. The problem of searching arguments is difficult and has higher complexity. Efstathiou and Hunter [16] search for arguments from propositional knowledge by applying a graph approach. However, the problem of searching arguments for a query in a given ontology is still an open problem. Finding an efficient approach to search arguments in DLs will be our future work.

Acknowledgements

The authors would like to thank Yue Ma, Kedian Mu, Guilin Qi, Shuang Ren and Guohui Xiao for helpful comments and discussions. This work is supported by the major program of the National Natural Science Foundation of China (NSFC).

References

1. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proc. of IJCAI'03, Acapulco, Mexico, August 9-15, 2003, Morgan Kaufmann (2003) 355–362
2. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: Proc. of IJCAI'05, Edinburgh, Scotland, UK, July 30-August 5, 2005, Professional Book Center (2005) 454–459
3. Qi, G., Haase, P., Huang, Z., Ji, Q., Pan, J.Z., Völker, J.: A kernel revision operator for terminologies - algorithms and evaluation. In: Proc. of ISWC'08, Karlsruhe, Germany, October 26-30, 2008. Volume 5318 of Lecture Notes in Computer Science., Springer (2008) 419–434
4. Patel-Schneider, P.F.: A four-valued semantics for terminological logics. Artif. Intell. **38**(3) (1989) 319–351
5. Ma, Y., Hitzler, P., Lin, Z.: Algorithms for paraconsistent reasoning with OWL. In: Proc. of ESWC'07, Innsbruck, Austria, June 3-7, 2007. Volume 4519 of Lecture Notes in Computer Science., Springer (2007) 399–413
6. Belnap, N.D.: A useful four-valued logic. Modern uses of multiple-valued logics (1977) 7–73
7. Zhang, X., Lin, Z.: Paraconsistent reasoning with quasi-classical semantic in ALC. In: Proc. of RR'08, Karlsruhe, Germany, October 31-November 1, 2008. Volume 5341 of Lecture Notes in Computer Science., Springer (2008) 222–229
8. Zhang, X., Xiao, G., Lin, Z.: A tableau algorithm for handling inconsistency in OWL. In: Proc. of ESWC'09, Heraklion, Crete, Greece, May 31-June 4, 2009. Volume 5554 of Lecture Notes in Computer Science., Springer (2009) 399–413
9. Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsisent beliefs without revising? In: Proc. of IJCAI'95, Montral, Qubec, Canada, August 20-25 1995. (1995) 1449–1457
10. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2) (1995) 321–358
11. Elvang-Gøransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. Data Knowl. Eng. **16**(2) (1995) 125–145
12. Hunter, A.: Making argumentation more believable. In: Proc. of AAAI'04, San Jose, California, USA, July 25-29, 2004, AAAI Press / The MIT Press (2004) 269–274
13. Hunter, A.: Reasoning about the appropriateness of proponents for arguments. In: Proc. of AAAI'08, Chicago, Illinois, USA, July 13-17, 2008, AAAI Press (2008) 89–94
14. Tempich, C., Simperl, E., Luczak, M., Studer, R., Pinto, H.S.: Argumentation-based ontology engineering. IEEE Intelligent Systems **22**(6) (2007) 52–59
15. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. 2 edn. Cambridge University Press (2007)
16. Efstathiou, V., Hunter, A.: Focused search for arguments from propositional knowledge. In: Proc. of COMMA'08, Toulouse, France, May 28-30, 2008. Volume 172 of Frontiers in Artificial Intelligence and Applications., IOS Press (2008) 159–170