

# Towards a Paradoxical Description Logic for the Semantic Web

Xiaowang Zhang<sup>1,2</sup>, Zuoquan Lin<sup>2</sup> and Kewen Wang<sup>1</sup>

<sup>1</sup>Institute for Integrated and Intelligent Systems, Griffith University, Australia

<sup>2</sup>School of Mathematical Sciences, Peking University, China  
{zxw,lzq}@is.pku.edu.cn, k.wang@griffith.edu.au

**Abstract.** As a vision for the future of the Web, the Semantic Web is an open, constantly changing and collaborative environment. Hence it is reasonable to expect that knowledge sources in the Semantic Web contain noise and inaccuracies. However, as the logical foundation of Ontology Web Language in the Semantic Web, description logics fail to tolerate inconsistent information. The study of inconsistency handling in description logics is an important issue in the Semantic Web. One major approach to inconsistency handling is based on so-called *paraconsistent reasoning*, in which standard semantics is refined so that inconsistencies can be tolerated. Four-valued description logics are not satisfactory for the Semantic Web in that its reasoning is a bit far from standard semantics. In this paper, we present a paraconsistent description logic called *paradoxical description logic*, which is based on a three-valued semantics. Compared to existing paraconsistent description logics, our approach is more suitable for dealing with inconsistent ontologies in that paraconsistent reasoning under our semantics provides a better approximation to the standard reasoning. An important result in this paper is that we propose a sound and complete tableau for paradoxical description logics.

## 1 Introduction

The Semantic Web<sup>1</sup>, which is conceived as a future generation of the World Wide Web (WWW), is an evolving development of the Web by defining the meaning (semantics) of information and services on the Web in order to make it possible to process and use the Web content by computer programs [1]. In an open, constantly changing and collaborative environment like the Semantic Web, knowledge bases and data in a real world will rarely be perfect. Many reasons, such as modeling errors, migration from other formalisms, merging ontologies and ontology evolution, potentially bring inconsistent knowledge. Then it is unrealistic to expect that real knowledge bases and data are always logically consistent [2,3]. Unfortunately, inconsistencies cause undesired problems in reasoning with classical logic. As long as inconsistencies are involved in reasoning, inference can be described as exploding, or trivialised. That is to say, anything, no matter whether it is meaningful, can follow from an inconsistent set of assumptions. Let  $T$  be a theory and  $\phi$  a formula.  $T \vdash \phi$  denotes that  $\phi$  is a consequence from  $T$  using classical

---

<sup>1</sup> <http://semanticweb.org/>

logic where  $\vdash$  is denoted the classical consequence relation. If  $\Gamma$  is inconsistent, then  $\Gamma \vdash \varphi$  and  $\Gamma \vdash \neg\varphi$  for some  $\varphi$  in the language. According to the inference rule of classical logic *ex falso quodlibet*,

$$\frac{\phi \quad \neg\phi}{\psi},$$

for any  $\psi$  in the language,  $\Gamma \vdash \psi$ . So inconsistencies cause classical logic to collapse. No useful inferences follow from an inconsistent theory, in the sense that inference is trivial [2]. Today, description logics (DLs) [4] have become a cornerstone of the Semantic Web for its use in the design of ontologies. For instance, the OWL-DL and OWL-Lite sub-languages of the W3C-endorsed Web Ontology Language (OWL)<sup>2</sup> are based on DLs. However, DLs, as a fragment of classical first-order logic (FOL), break down in the presence of inconsistent knowledge [2]. As a result, inconsistency handling in DLs has attracted much attention in the Semantic Web community and Artificial Intelligence (AI) community.

The study of inconsistency handling in AI has a long tradition and many works exist [5,6,7,8,9,10,11,12,13,14] etc. The issue of how to transfer these existing results in classical logic. As a result, several approaches to handle inconsistencies in DLs have been proposed, which can be roughly divided into two fundamentally different categories. The first one is based on the assumption that inconsistencies indicate erroneous data which are to be repaired in order to obtain a consistent ontology, e.g., by pinpointing the parts of an inconsistent ontology, which cause inconsistencies, and remove or weaken axioms in these parts to restore consistencies [15,16,17,18]. In these approaches, researchers hold a common view that knowledge bases should be completely free of inconsistencies, and thus try to eliminate inconsistencies from them immediately by any means possible. This view is regarded as “too simplistic for developing robust intelligent systems, and fails to use the benefits of inconsistent knowledge in intelligent activities, or to acknowledge the fact that living with inconsistencies seems to be unavoidable” in [2]. Inconsistencies in knowledge are the norm in the real world, and so should be formalized and used, rather than always rejected [2]. Another category of approaches, called *paraconsistent approaches*, is not to simply avoid inconsistencies but to tolerate them by applying a non-standard reasoning method to obtain meaningful answers [19,20,21,22,23]. In this kind of approaches, inconsistencies are treated as a natural phenomenon in realistic data and are tolerated in reasoning. The call for robust intelligent systems has led to an increased interest in inconsistency tolerance in AI. Compared with the former, the latter acknowledges and distinguishes the different epistemic statuses between “*the assertion is true*” and “*the assertion is true with conflict*”. So far, paraconsistent approaches to inconsistency handling in DLs, such as [19,20,21], are based on Belnap’s four-valued semantics [5]. The major idea is to use additional truth values *underdefined* (i.e. neither true nor false) and *overdefined* (i.e. *contradictory*, or, both true and false). To a certain extent *four-valued semantics* can help us to handle inconsistencies. However the four-valued semantics is relatively weak because some important inference rules are not valid in the four-valued semantics. These rules include: (1) the *law of excluded middle*:  $\models A \sqcup \neg A(a)$  where  $A$  is a concept name and  $a$

<sup>2</sup> <http://www.w3.org/TR/owl-features/>

is an individual. This rule captures the tautology in logics, and (2) some *intuitive equivalences* including:  $\mathcal{O} \models C \sqsubseteq D$  if and only if  $\mathcal{O} \models \neg C \sqcup D(a)$  for any individual  $a$  (where  $C$  and  $D$  are concepts). Such rules are employed to reduce reasoning tasks w.r.t. ABoxes and TBoxes to reasoning tasks w.r.t. only ABoxes [24]. These shortcomings are inherent limitations of four-valued logics in paraconsistent reasoning. Besnard and Hunter [8,?] proposed a paraconsistent approach for classical logic, known as *quasi-classical logic* (abbreviated *QC*). This paraconsistent logic has been introduced to DLs by Zhang et al [22,23], where two semantics *strong semantics* and *weak semantics* are introduced in DLs in order to make more useful inference rules valid. The *concept inclusions* under two semantics are defined differently in order to maintain some intuitive equivalences at the cost of effectiveness. However, the *excluded middle rule* is still disabled in QC DLs. Based on the *argumentation theory* in [9], Zhang et al [25] present a hybrid approach to deal with inconsistent ontologies in description logics. A drawback of the approach is that the complexity of the argumentative semantics is much higher than that of classical logic.

In short, existing approaches to paraconsistency for DLs employ a “nonstandard” semantics that is from “*relevance logics*”<sup>3</sup>. In such semantics, to tolerate inconsistencies, the cost is to weaken the inference power. For instance, their inference power even are weaker than that of classical logics in reasoning with consistent classical ontologies since they are relevant logics. However, DLs, as classical logics, are not relevant logics. Naturally, we expect that a paraconsistent approach to inconsistency handling in DLs can preserve properties of classical logics as many as possible. That is to say, paraconsistent DLs is expected to be more close to classical DLs. For instance, candidate paraconsistent DLs can satisfy as inference rules (or proof rules) of DLs as many as possible. The motivation of this paper is to present a “*brave-reasoning*”<sup>4</sup> version of DLs which is more close to DLs.

In this paper, we present a paraconsistent extension of DL *ALC*, called *paradoxical DL ALC* (written by  $\mathcal{ALC}_{LP}$ ), which is an extension of *ALC* with semantics of logics of paradox (LP) presented by Priest in [6,7]. We introduce a paraconsistent semantics for *ALC*, which is closer to the standard semantics than the four-valued one. We prove that the paradoxical entailment satisfies the *excluded middle rule* and *intuitive equivalence*. A signed tableau for  $\mathcal{ALC}_{LP}$  is proposed to implement paraconsistent reasoning in DL-based ontologies. Firstly, we introduce two symbols **T** (for *true*) and **F** (for *false*) in  $\mathcal{ALC}_{LP}$  which are employed to mark the states of axioms in a given situation. Secondly, signed expansion rules are provided to construct signed tableaux for ABoxes. Finally, closed conditions are modified to tolerate inconsistencies and capture paraconsistency in the signed tableau. Furthermore, we prove that the signed tableau is sound and complete with respect to the paradoxical entailment in  $\mathcal{ALC}_{LP}$ .

The rest of the paper is organized as follows: in the next section, we give a short introduction of the syntax, the semantics of LP and DL *ALC*. Then Section 3 introduces the paradoxical semantics for DL *ALC* and two basic inference problems. Section 4

<sup>3</sup> <http://plato.stanford.edu/entries/logic-relevance/>

<sup>4</sup> In general, there are two reasoning mechanisms: “*cautious reasoning*” and “*brave reasoning*”. Cautious reasoning is done w.r.t. a local ontology and its neighbours while brave reasoning is done w.r.t. the transitive closure.

presents a signed tableau algorithm for  $\mathcal{ALC}_{LP}$  to implement paraconsistent reasoning. Section 5 discusses related works. Finally, in Section 6, we conclude this paper and discuss some future work.

## 2 Preliminaries

In this section, we recall some basics of logics of paradox (LP) and DL  $\mathcal{ALC}$ .

### 2.1 Logics of Paradox

A *logic of paradox* (LP) is a three-valued logic, in which one additional value is used to represent conflict besides the two standard truth values. We refer the reader to [6,7,26,27] for further details.

A *paradoxical interpretation*  $\pi$  assigns to each atomic sentence  $\phi$  in  $\mathcal{L}$  one of the following three values:  $f$  (false and false only),  $t$  (true and true only) and  $B$  (both true and false). A sentence  $\phi$  is *true* under  $\pi$  if  $\pi(\phi) = t$  or  $\pi(\phi) = B$ ; and  $\phi$  is *false* under  $\pi$  if  $\pi(\phi) = f$  or  $\pi(\phi) = B$ . Under any paradoxical interpretation, truth values can be conventionally extended to non-atomic sentences as follows:

- (1) –  $\neg\phi$  is true if and only if  $\phi$  is false;  
–  $\neg\phi$  is false if and only if  $\phi$  is true;
- (2) –  $\phi \wedge \psi$  is true if and only if both  $\phi$  and  $\psi$  are true;  
–  $\phi \wedge \psi$  is false if and only if either  $\phi$  or  $\psi$  is false;
- (3) –  $\forall x.\phi(x)$  is true if and only if for all closed terms  $t$ ,  $\phi(t)$  is true;  
–  $\forall x.\phi(x)$  is false if and only if for some closed terms  $t$ ,  $\phi(t)$  is false.

Note that  $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$ ,  $\phi \rightarrow \psi \equiv \neg(\phi \wedge \neg\psi)$  and  $\exists x.\phi(x) \equiv \neg\forall x.\neg\phi(x)$  where  $\equiv$  is the *logical equivalence (relationship)*.

For the sake of simplicity, we assume that every object in the domain is named by a closed term. Similarly, other connectives and qualifiers can be easily defined via  $\neg$ ,  $\wedge$  and  $\forall$  as a standard way.

- $\pi(\phi) = t$  if  $\phi$  is true but not false under  $\pi$ ;
- $\pi(\phi) = f$  if  $\phi$  is false but not true under  $\pi$ ;
- $\pi(\phi) = B$  if  $\phi$  is both true and false under  $\pi$ .

Let  $\Gamma$  be a set of sentences and  $\psi$  a sentence. We say  $\Gamma$  *paradoxically entails*  $\psi$ , denoted by  $\Gamma \models_{LP} \psi$ , if and only if  $\psi$  is true in all paradoxical models of  $\Gamma$ , that is, for any paradoxical interpretation  $\pi$ , if every member of  $\Gamma$  is true under  $\pi$ , then  $\psi$  is also true under  $\pi$ .

**Example 1** Let  $\phi$  and  $\psi$  be two different atomic sentences. It is easy to see that  $\phi \wedge \neg\phi \models \psi$ , but  $\phi \wedge \neg\phi \not\models_{LP} \psi$ . This can be seen by noting that  $\phi \wedge \neg\phi$  is true (actually it is both true and false) but  $\psi$  is not true, under the paradoxical interpretation  $\pi$  with  $\pi(\phi) = B$  and  $\pi(\psi) = f$ .

By the above example, LP gets rid of trivial problem of classical logic, and as a paraconsistent logic, inconsistencies are indeed localized in the sense. Furthermore, the *law of excluded middle* is valid in LP. For instance, let  $\phi$  be an atomic sentence. It easily concludes that  $\models_{LP} \phi \vee \neg\phi$  since  $\phi \vee \neg\phi \equiv \neg(\neg\phi \wedge \phi)$ .

## 2.2 Description Logic $\mathcal{ALC}$

Description logics (DLs) is a well-known family of knowledge representation formalisms. For more comprehensive background reasoning, we refer the reader to Chapter 2 of the DL Handbook [4]. DLs can be seen as fragments of *first-order predicate logic*. They are different from their predecessors such as semantic networks and frames in that they are equipped with a formal, logic-based semantics. In DLs, elementary descriptions are *concept names* (unary predicates) and *role names* (binary predicates). Complex descriptions are built from them inductively using concept and role constructors provided by the particular DLs under consideration.

In this paper, we consider  $\mathcal{ALC}$  which is a simple yet relatively expressive DL.  $\mathcal{AL}$  is the abbreviation of attributive language and  $\mathcal{C}$  denotes “*complement*”. Let  $N_C$  and  $N_R$  be pairwise disjoint and countably infinite sets of *concept names* and *role names* respectively. Let  $N_I$  be an infinite set of *individual names*. We use the letters  $A$  and  $B$  for concept names, the letter  $R$  for role names, and the letters  $C$  and  $D$  for concepts.  $\top$  and  $\perp$  denote the *universal concept* and the *bottom concept* respectively. The set of  $\mathcal{ALC}$  concepts is the smallest set such that:

- (1) every concept name is a concept;
- (2) if  $C$  and  $D$  are concepts,  $R$  is a role name, then the following expressions are also concepts:  $\neg C$  (*full negation*),  $C \sqcap D$  (*concept conjunction*),  $C \sqcup D$  (*concept disjunction*),  $\forall R.C$  (*value restriction on role names*) and  $\exists R.C$  (*existential restriction on role names*).

For instance, the concept description  $Person \sqcap Female$  is an  $\mathcal{ALC}$ -concept describing those persons that are female. Suppose  $hasChild$  is a role name, the concept description  $Person \sqcap \forall hasChild.Female$  expresses those persons whose children are all female. The concept  $\forall hasChild.\perp \sqcap Person$  describes those persons who have no children.

An interpretation  $I = (\Delta^I, \cdot^I)$  consisting of a non-empty domain  $\Delta^I$  and a mapping  $\cdot^I$  which maps every concept to a subset of  $\Delta^I$  and every role to a subset of  $\Delta^I \times \Delta^I$ , for all concepts  $C, D$  and a role  $R$ , satisfies conditions in the following table.

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

Constructor	Syntax	Semantics
concept	$A$	$A^I \subseteq \Delta^I$
role	$R$	$R^I \subseteq \Delta^I \times \Delta^I$
individual	$o$	$o^I \in \Delta^I$
top concept	$\top$	$\Delta^I$
bottom concept	$\perp$	$\emptyset^I$
negation	$\neg C$	$\Delta^I \setminus C^I$
conjunction	$C_1 \sqcap C_2$	$C_1^I \cap C_2^I$
disjunction	$C_1 \sqcup C_2$	$C_1^I \cup C_2^I$
existential restriction	$\exists R.C$	$\{x \mid \exists y, (x, y) \in R^I \text{ and } y \in C^I\}$
value restriction	$\forall R.C$	$\{x \mid \forall y, (x, y) \in R^I \text{ implies } y \in C^I\}$

A *general concept inclusion axiom (GCI)* or a *terminology* is an inclusion statement of the forms  $C \sqsubseteq D$ , where  $C$  and  $D$  are two (possibly complex)  $\mathcal{ALC}$  concepts (concepts for short). It is the statement about how concepts are related to each other. We use  $C \equiv D$  as an abbreviation for the symmetrical pair of GCIs  $C \sqsubseteq D$  and  $D \sqsubseteq C$ , called a *concept definition*. An interpretation  $I$  satisfies a GCI  $C \sqsubseteq D$  if and only if  $C^I \subseteq D^I$ , and it satisfies a GCI  $C \equiv D$  if and only if  $C^I = D^I$ . A finite set of GCIs is called a *TBox*. An *acyclic TBox* is a finite set of concept definitions such that every concept name occurs at most once on the left-hand side of an axiom, and there is no cyclic dependency between the definitions.

We can also formulate statements about individuals. A *concept (role) assertion axiom* has the form  $C(a) (R(a, b))$ , where  $C$  is a concept description,  $R$  is a role name, and  $a, b$  are individual names. An *ABox* contains a finite set of concept axioms and role axioms. In the ABox, one describes a specific state of affairs of an application domain in terms of concept and roles.

To give a semantics to ABoxes, we need to extend interpretations to individual names. For each individual name  $a$ ,  $\cdot^I$  maps it to an element  $a^I \in \Delta^I$ . We assume the mapping  $\cdot^I$  satisfies the *unique name assumption (UNA)* in which  $a^I \neq b^I$  for distinct names  $a$  and  $b$ .

An interpretation  $I$  satisfies a concept axiom  $C(a)$  if and only if  $a^I \in C^I$ ;  $I$  satisfies a role axiom  $R(a, b)$  if and only if  $(a^I, b^I) \in R^I$ . An ontology  $\mathcal{O}$  consists of a TBox and an ABox, i.e., it is a set of GCIs and assertion axioms.

An interpretation  $I$  is a *model* of a DL (TBox or ABox) axiom if and only if it satisfies this axiom; and it is a model of an ontology  $\mathcal{O}$  if it satisfies every axiom in  $\mathcal{O}$ . A concept  $D$  *subsumes* a concept  $C$  w.r.t. a TBox  $\mathcal{T}$  if and only if each model of  $\mathcal{T}$  is a model of axiom  $C \sqsubseteq D$ . An ABox  $\mathcal{A}$  is consistent w.r.t. a TBox  $\mathcal{T}$  if and only if there exists a common model of  $\mathcal{T}$  and  $\mathcal{A}$ .

Given an ontology  $\mathcal{O}$  and a DL axiom  $\phi$ , we say  $\mathcal{O}$  *entails*  $\phi$ , denoted  $\mathcal{O} \models \phi$ , if and only if every model of  $\mathcal{O}$  is a model of  $\phi$ . A concept  $C$  is *satisfiable* if and only if there exists an individual  $a$  such that the ABox  $\{C(a)\}$  is consistent; and *unsatisfiable* otherwise. A concept  $C$  is *satisfiable* w.r.t. a TBox  $\mathcal{T}$  if and only if there exists a model  $I$  of  $\mathcal{T}$  such that  $C^I \neq \emptyset$  is consistent; and *unsatisfiable* otherwise.

Two basic reasoning problems, namely, *instance checking* (checking whether an individual is an instance of a given concept) and *subsumption checking* (checking whether a concept subsumes a given concept) can be reduced to the problem of consistency by the following lemma.

**Lemma 1 ([4])** *Let  $\mathcal{O}$  be an ontology,  $C, D$  concepts and  $a$  an individual in  $\mathcal{ALC}$ .*

- (1)  $\mathcal{O} \models C(a)$  if and only if  $\mathcal{O} \cup \{\neg C(a)\}$  is inconsistent.
- (2)  $\mathcal{O} \models C \sqsubseteq D$  if and only if  $\mathcal{O} \cup \{C \sqcap \neg D(\iota)\}$  is inconsistent where  $\iota$  is a new individual not occurring in  $\mathcal{O}$ .

The following lemma by Horrocks et al [24] shows that satisfiability, unsatisfiability and consistency of a concept w.r.t. an acyclic TBox can be reduced to the corresponding reasoning task w.r.t the empty TBox. This result is obtained by introducing a “universal” role  $U$ , that is, if  $y$  is reachable from  $x$  via a role path, then  $\langle x, y \rangle \in U^I$ .

**Lemma 2 ([24])** Let  $C, D$  be concepts,  $\mathcal{A}$  an ABox and  $\mathcal{T}$  an acyclic TBox in  $\mathcal{ALC}$ . Define

$$C_{\mathcal{T}} := \prod_{C_i \sqsubseteq D_i \in \mathcal{T}} \neg C_i \sqcup D_i.$$

Then the following properties hold:

- (1)  $C$  is satisfiable w.r.t.  $\mathcal{T}$  if and only if  $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$  is satisfiable;
- (2)  $D$  subsumes  $C$  w.r.t.  $\mathcal{T}$  if and only if  $C \sqcap \neg D \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$  is unsatisfiable;
- (3)  $\mathcal{A}$  is consistent w.r.t.  $\mathcal{T}$  if and only if  $\mathcal{A} \cup \{C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}(a) \mid a \in U_{\mathcal{A}}\}$  is consistent, where  $U_{\mathcal{A}}$  is a set of all individuals occurring in  $\mathcal{A}$ .

Since a reasoning problem w.r.t. an ABox and an acyclic TBox can be reduced to the same reasoning problem w.r.t. the empty TBox. So in this paper, we mainly consider reasoning problems w.r.t. ABoxes.

### 3 Paradoxical Description Logic $\mathcal{ALC}$

In this section, we present the syntax and semantics of the paradoxical description logic  $\mathcal{ALC}$ , named  $\mathcal{ALC}_{LP}$ .

#### 3.1 Syntax and Semantics

Syntactically,  $\mathcal{ALC}_{LP}$  hardly differs from  $\mathcal{ALC}$ . In  $\mathcal{ALC}_{LP}$ , complex concepts and assertions are defined exactly in the same way. In the following, let  $\mathcal{L}$  be the language of  $\mathcal{ALC}_{LP}$ .

Semantically, *paradoxical interpretations* map individuals to elements of the domain of the interpretation, as usual. However, as for concepts, to allow for reasoning with inconsistencies, a paradoxical interpretation over domain  $\Delta^I$  assigns to each concept  $C$  a pair  $\langle +C, -C \rangle$  of (not necessarily disjoint) subsets of  $\Delta^I$  and  $+C \cup -C = \Delta^I$ .

Intuitively,  $+C$  is the set of elements which are known to belong to the extension of  $C$ , while  $-C$  is the set of elements which are known to be not contained in the extension of  $C$ .  $+C$  and  $-C$  are not necessarily disjoint but mutual complementary w.r.t. the domain. In this case, we do not consider that *incomplete information* since it is not valuable for users but a statement for insufficient information. Under the semantics of  $\mathcal{ALC}_{LP}$ , three situations which may occur in terms of containment of an individual in a concept are considered as follows:

- we know it is contained;
- we know it is not contained;
- we have contradictory information, i.e., the individual is both contained in the concept and not contained in the concept.

There are several equivalent ways of how this intuition can be formalized, one of which is described in the following.

Formally, a paradoxical interpretation is a pair  $I = (\Delta^I, \cdot^I)$  with  $\Delta^I$  as domain, where  $\cdot^I$  is a function assigning elements of  $\Delta^I$  to individuals, and subsets of  $(\Delta^I)^2$

to concepts, so that the conditions in Table 2 are satisfied, where functions  $\text{proj}^+(\cdot)$  and  $\text{proj}^-(\cdot)$  are defined by  $\text{proj}^+\langle +C, -C \rangle = +C$  and  $\text{proj}^-\langle +C, -C \rangle = -C$ . Table 2 for role restrictions are designed in such a way that the semantic 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 to handle role restrictions, as it will allow for a straightforward translation from  $\mathcal{ALC}_{LP}$  to classical  $\mathcal{ALC}$ .

**Table 2.** Semantics of  $\mathcal{ALC}_{LP}$  Concepts

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

As for roles, actually we require only the positive part of the extension since  $\neg R$  is not a constructor in  $\mathcal{ALC}$ , i.e.,  $\text{proj}^+(R^I) \cap \text{proj}^-(R^I) = \emptyset$  for any  $\mathcal{ALC}$  role  $R$ . However,  $\neg R$  is a constructor in OWL 2<sup>5</sup>, such as  $\mathcal{EL}$  [28]. Thus, inconsistencies may occur in roles in OWL 2. For instance, an  $\mathcal{EL}$  ontology tells that *Mike* does not like *Jane*, i.e.,  $\neg \text{like}(\text{Mike}, \text{Jane})$  while it is inferred that *Mike* likes *Jane*, i.e.,  $\text{like}(\text{Mike}, \text{Jane})$ . That is,  $\text{proj}^+(R^I) \cap \text{proj}^-(R^I) = \{(\text{Mike}, \text{Jane})\}$ . This example shows that, to deal with inconsistency, an interpretation should assign a pair of sets to a role. For a given domain  $\Delta^I$  and a concept  $C$ , under the constraints  $+C \cap -C = \emptyset$ , paradoxical interpretations just become standard two-valued interpretations.

The correspondence between truth values from  $\{t, f, B\}$  and concept extensions can be observed easily: for an individual  $a \in \Delta^I$  and a concept name  $C$ , we have that

- $C^I(a) = t$ , if and only if  $a^I \in \text{proj}^+(C^I)$  and  $a^I \notin \text{proj}^-(C^I)$ ;
- $C^I(a) = f$ , if and only if  $a^I \notin \text{proj}^+(C^I)$  and  $a^I \in \text{proj}^-(C^I)$ ;
- $C^I(a) = B$ , if and only if  $a^I \in \text{proj}^+(C^I)$  and  $a^I \in \text{proj}^-(C^I)$ .

The paraconsistent semantics defined above ensures that a number of useful equivalences from classical DLs are still valid, including *double negation law* and *De Morgan Law*. The following theorem summarizes these desirable characteristics.

<sup>5</sup> <http://www.w3.org/TR/owl2-profiles/>

**Theorem 1** For any paradoxical interpretation  $I$ , concepts  $C, D$  and a role  $R$  in  $\mathcal{ALC}_{LP}$ , the following claims hold.

- |                                                       |                                                       |
|-------------------------------------------------------|-------------------------------------------------------|
| (1) $(C \sqcap \top)^I = C^I$                         | (2) $(C \sqcup \top)^I = \top^I$                      |
| (3) $(C \sqcap \perp)^I = \perp^I$                    | (4) $(C \sqcup \perp)^I = C^I$                        |
| (5) $(\neg \top)^I = \perp^I$                         | (6) $(\neg \perp)^I = \top^I$                         |
| (7) $(\neg(C \sqcup D))^I = (\neg C \sqcap \neg D)^I$ | (8) $(\neg(C \sqcap D))^I = (\neg C \sqcup \neg D)^I$ |
| (9) $(\neg(\forall R.C))^I = (\exists R.\neg C)^I$    | (10) $(\neg(\exists R.C))^I = (\forall R.\neg C)^I$   |
| (11) $(\neg\neg C)^I = C^I$                           | (12) $proj^+((C \sqcup \neg C)^I) = proj^+(\top^I)$   |

From (12) in Theorem 1, we can see that  $(C \sqcup \neg C)^I$  may be different from  $\top^I$  for some concepts  $C$ . For instance, if  $\Delta^I = \{a, b, c\}$  and  $C^I = \{\{a, b\}, \{b, c\}\}$  then  $(C \sqcup \neg C)^I = \{\{a, b, c\}, \{b\}\}$  while  $\top^I = \{\{a, b, c\}, \emptyset\}$ , i.e.,  $(C \sqcup \neg C)^I \neq \top^I$ . Similarly, we conclude that  $(C \sqcap \neg C)^I \neq \perp^I$ .

By Theorem 1, we can equivocally transform the negation of a complex concept into its *negation normal form* (NNF) which negation ( $\neg$ ) only occurs on front of concept names in it.

Having defined paradoxical interpretations, we now introduce the *paradoxical satisfaction* relation.

**Definition 1** The *paradoxical satisfaction*, denoted  $\models_{LP}$ , is a binary relation between a set of paradoxical interpretations and a set of axioms defined as follows:

For any paradoxical interpretation  $I$ ,

- (1)  $I \models_{LP} C(a)$  if and only if  $a^I \in +C$ , where  $C^I = \langle +C, -C \rangle$ ;
- (2)  $I \models_{LP} R(a, b)$  if and only if  $(a^I, b^I) \in +R$ , where  $R^I = \langle +R, -R \rangle$ ;
- (3)  $I \models_{LP} C_1 \sqsubseteq C_2$  if and only if  $(\Delta^I \setminus -C_1) \subseteq +C_2$ , for  $i = 1, 2$ ,  $C_i^I = \langle +C_i, -C_i \rangle$ .

where  $C, C_1, C_2$  are concepts,  $R$  a role and  $a$  an individual in  $\mathcal{ALC}$ .

Note that  $I$  paradoxically satisfies a concept assertion  $C(a)$  if and only if  $a^I$  is known to be a member of  $C^I$ ;  $I$  paradoxically satisfies a role assertion  $R(a, b)$  if and only if it is known that  $a^I$  is related to  $b$  by the binary relation  $R^I$ ; and,  $I$  paradoxically satisfies a concept inclusion  $C \sqsubseteq D$  if and only if every member of  $C^I$  is a member of  $D^I$ . Thus, concept  $C$  and its negation  $\neg C$  are taken as two different concepts under paradoxical satisfaction. Based on the paradoxical satisfaction of concept inclusions, when there is an inconsistency with one concept, it will not be propagated to other concepts through axioms. For instance, let  $\mathcal{T} = \{A \sqsubseteq B, A \sqsubseteq \neg B, B \sqsubseteq C\}$  be a TBox,  $I$  a paradoxical interpretation such that  $I \models_{LP} A \sqsubseteq B$ ,  $I \models_{LP} A \sqsubseteq \neg B$  and  $I \models_{LP} B \sqsubseteq C$ . Thus  $proj^+(B^I) \cap proj^-(B^I) = proj^+(A^I)$ . Then  $\Delta^I \setminus proj^-(A^I) \not\subseteq proj^+(C^I)$ . Therefore,  $I \not\models_{LP} A \sqsubseteq C$ . Intuitively, because concept  $B$  is unsatisfiable w.r.t.  $\mathcal{T}$ , i.e., inconsistencies would occur in any ontologies by integrating  $\mathcal{T}$  with some ABox. Fortunately, inconsistencies are not transmitted into more general concepts.

In addition, the following result asserts that the *intuitive equivalence* is valid in  $\mathcal{ALC}_{LP}$ .

**Theorem 2** Let  $C_1, C_2$  be concepts and  $I$  a paradoxical interpretation in  $\mathcal{ALC}_{LP}$ .

$I \models_{LP} C_1 \sqsubseteq C_2$  if and only if  $I \models_{LP} \neg C_1 \sqcup C_2(a)$  for every individual  $a$ .

Theorem 2 in company with Lemma 1 ensures that a reasoning problem w.r.t. an ABox and a TBox can be equivalently transformed into the same reasoning problem w.r.t. an ABox (i.e. the TBox is empty).

In  $\mathcal{ALC}_{LP}$ , a paradoxical interpretation  $I$  *paradoxically satisfies* an ABox  $\mathcal{A}$  (i.e., a paradoxical model of it), denoted by  $I \models_{LP} \mathcal{A}$ , if and only if it satisfies each assertion in  $\mathcal{A}$ ;  $I$  *paradoxically satisfies* a TBox  $\mathcal{T}$ , denoted by  $I \models_{LP} \mathcal{T}$ , if and only if it satisfies each inclusion axiom in  $\mathcal{T}$ ;  $I$  *paradoxically satisfies* an ontology  $\mathcal{O}$ , denoted by  $I \models_{LP} \mathcal{O}$ , if and only if it satisfies its ABox and its TBox. An ontology  $\mathcal{O}$  is *paradoxically consistent (inconsistent)* if and only if there exists (does not exist) such a paradoxical model. In the following, we cut out “*paradoxically*” or “*paradoxical*” from terms for simplicity if no confusion occurs.

The following theorem shows that any classical ontology is paradoxically consistent, that is, contradictions are tolerated under the semantics of  $\mathcal{ALC}_{LP}$ .

**Theorem 3** *There exists a paradoxical model of  $\mathcal{O}$  for any ontology  $\mathcal{O}$  in  $\mathcal{ALC}$ .*

Next, we introduce the *paradoxical entailment* relation between ontologies and axioms based on paradoxical models.

**Definition 2** *Let  $\mathcal{O}$  be an ontology and  $\phi$  be an axiom in  $\mathcal{ALC}_{LP}$ . We say  $\mathcal{O}$  **paradoxically entails**  $\phi$ , denoted  $\mathcal{O} \models_{LP} \phi$ , if and only if for every paradoxical model  $I$  of  $\mathcal{O}$ ,  $I \models_{LP} \phi$ .*

The following example shows that the paradoxical entailment does not satisfy the *ex contradictione quodlibet*, i.e.,  $\mathcal{ALC}_{LP}$  is paraconsistent.

**Example 2** *Let  $A$  be a concept name and  $a$  an individual in  $\mathcal{ALC}_{LP}$ . Then  $\{A(a), \neg A(a)\} \models_{LP} A(a)$  and  $\{A(a), \neg A(a)\} \models_{LP} \neg A(a)$ . Let  $I$  be a paradoxical interpretation such that  $A^I(a) = B$  and  $A'^I(a) = f$  where  $b$  is an individual and  $A'$  a concept name. That is,  $I$  is a paradoxical model of  $\{A(a), \neg A(a)\}$  but  $I$  is not a paradoxical model of  $A'(b)$ . Then  $\{A(a), \neg A(a)\} \not\models_{LP} A'(b)$ .*

In the following, we build the relationship between paradoxical interpretations and classical interpretations in  $\mathcal{ALC}_{LP}$  by introducing the definition of *isomorphic map*.

**Definition 3** *Let  $I$  be a paradoxical interpretation and  $I_c$  be a classical interpretation. The **isomorphic map** transforming  $I$  into  $I_c$  is defined as follows: for each axiom  $\phi$ ,*

$$\begin{cases} I_c \models \phi, & \text{if and only if } I \models_{LP} \phi; \\ I_c \not\models \phi, & \text{if and only if } I \not\models_{LP} \phi. \end{cases}$$

By Definition 3, we conclude that for any paradoxical interpretation  $I$ , there exists a classical interpretation  $I_c$  such that the isomorphic map (defined in Definition 3) can transform  $I$  into  $I_c$ .

It is a pleasing property that reasoning in  $\mathcal{ALC}_{LP}$  can be reduced easily into reasoning in the standard  $\mathcal{ALC}$ . Thus, reasoning algorithms in  $\mathcal{ALC}$  can be used to implement paraconsistent reasoning in  $\mathcal{ALC}_{LP}$ .

In general, the reasoning power of  $\mathcal{ALC}_{LP}$  is weaker than that of  $\mathcal{ALC}$ .

**Theorem 4** Let  $\mathcal{O}$  be an ontology and  $\phi$  an axiom in  $\mathcal{ALC}$ .

If  $\mathcal{O} \models_{LP} \phi$  then  $\mathcal{O} \models \phi$ ; but not vice versa.

However, when ontologies are classically consistent,  $\mathcal{ALC}_{LP}$  has the same inference power as  $\mathcal{ALC}$ .

**Theorem 5** Let  $\mathcal{O}$  be a classically consistent ontology and  $\phi$  an axiom in  $\mathcal{ALC}$ .

$\mathcal{O} \models_{LP} \phi$  if and only if  $\mathcal{O} \models \phi$

The next result shows that the set of tautologies in  $\mathcal{ALC}_{LP}$  is exactly the set of tautologies in  $\mathcal{ALC}$ .

**Theorem 6** Let  $\phi$  be an axiom in  $\mathcal{ALC}$ .

$\models_{LP} \phi$  if and only if  $\models \phi$

The *excluded middle rule* is also valid in  $\mathcal{ALC}_{LP}$ .

**Corollary 1**  $\models_{LP} C \sqcup \neg C(a)$  for any concept  $C$  and individual  $a$  in  $\mathcal{ALC}$ .

By Corollary 1, we conclude that the semantics of  $\mathcal{ALC}_{LP}$  is different from the semantics of relevance logic.

### 3.2 Inference Problems

The *instance checking* and *subsumption checking* are two important inference problems in  $\mathcal{ALC}$ . In this section, we look at these two problem in the setting of  $\mathcal{ALC}_{LP}$ .

First, we give their definition as follows:

- **instance checking**: an individual  $a$  is a *paradoxical instance* of a concept  $C$  w.r.t. an ontology  $\mathcal{O}$  if  $\mathcal{O} \models_{LP} C(a)$ ;
- **subsumption checking**: a concept  $C$  is *paradoxically subsumed* by a concept  $D$  w.r.t. a TBox  $\mathcal{T}$  if  $\mathcal{T} \models_{LP} C \sqsubseteq D$ .

An important result is that Lemma 2 still holds in  $\mathcal{ALC}_{LP}$ .

**Corollary 2** Let  $C, D$  be concepts,  $\mathcal{A}$  an ABox and  $\mathcal{T}$  an acyclic TBox in  $\mathcal{ALC}_{LP}$ . Then we have

- (1)  $C$  is *paradoxically satisfiable* w.r.t.  $\mathcal{T}$  if and only if  $C \sqcap C_{\mathcal{T}} \sqcap \forall U. C_{\mathcal{T}}$  is *paradoxically satisfiable*;
- (2)  $D$  *paradoxically subsumes*  $C$  w.r.t.  $\mathcal{T}$  if and only if  $C \sqcap \neg D \sqcap C_{\mathcal{T}} \sqcap \forall U. C_{\mathcal{T}}$  is *paradoxically unsatisfiable*;
- (3)  $\mathcal{A}$  is *paradoxically consistent* w.r.t.  $\mathcal{T}$  if and only if  $\mathcal{A} \cup \{C_{\mathcal{T}} \sqcap \forall U. C_{\mathcal{T}}(a) \mid a \in U_{\mathcal{A}}\}$  is *paradoxically consistent*, where  $U_{\mathcal{A}}$  is a set of all individuals occurring in  $\mathcal{A}$ .

By Corollary 2, the problem of subsumption checking can be reduced to the problem of instance checking and the problem of instance checking w.r.t. ABoxes and TBoxes can be reduced to the problem of instance checking w.r.t. only ABoxes. Thus, in the rest of the paper we will mainly consider these reasoning problems w.r.t. ABoxes.

**Example 3** Let  $\mathcal{T}$  be a TBox and let  $C$ ,  $D$ , and  $E$  be concepts in  $\mathcal{ALC}_{LP}$ . By Definition 1 and Definition 2, if  $D$  is classical satisfiable w.r.t.  $\mathcal{T}$ , that is, for any paradoxical interpretation  $I$  of  $\mathcal{T}$ ,  $\text{proj}^+(D^I) \cap \text{proj}^-(D^I) = \emptyset$ , then  $\{C \sqsubseteq D, D \sqsubseteq E\} \models_{LP} C \sqsubseteq E$ ; if  $D$  is classical unsatisfiable w.r.t.  $\mathcal{T}$ , we have  $\{C \sqsubseteq D, D \sqsubseteq E\} \not\models_{LP} C \sqsubseteq E$ .

From Example 3, we can see that concept inclusions do not satisfy the transitivity in  $\mathcal{ALC}_{LP}$  when TBoxes contain unsatisfiable concepts. This implies that inconsistencies are not transited in  $\mathcal{ALC}_{LP}$ . Because of this, the principle of reasoning with inconsistent ontologies here is reasonable in the real world since the negative influence arising from inconsistencies is minimized or localized to some extent.

## 4 Signed Tableau Algorithm

In this section, we discuss the problem of instance checking and provide a sound and complete tableau for  $\mathcal{ALC}_{LP}$  by extending the signed tableau for LP developed by Lin [27] to  $\mathcal{ALC}_{LP}$ . As explained before, we assume that the TBox of an ontology is empty.

Let  $\mathcal{A}$  be an ABox and  $\phi$  a concept assertion in  $\mathcal{ALC}_{LP}$ . Under the paradoxical semantics of  $\mathcal{ALC}_{LP}$ , an axiom can be both true and false and triviality of classical  $\mathcal{ALC}$  is destroyed. There are two approaches to destroying the triviality of proofs. One approach is to restrict the rule of *reductio ad absurdum*:  $\mathcal{A}$  paradoxically entails  $\phi$  if and only if we can infer a contradiction from  $\mathcal{A}$  and  $\neg\phi$  by using relevant information from  $\phi$ . With this idea of using relevant information from  $\phi$ , we can formulate the proof procedure by retaining the tableau rules for classical logic but modifying closedness conditions of the tableau to get rid of the triviality. The other approach to formulating the restricted *reduction* to *absurdity* is using *signed tableau* as we will present in the following.

We first introduce two symbols **T** (for *true*) and **F** (for *false*) in  $\mathcal{L}$ . A *signed axiom* has the form **T** $\phi$  (called **T**-axiom) or **F** $\phi$  (called **F**-axiom), where  $\phi \in \mathcal{L}$ . A paradoxical interpretation  $I$  is extended to a signed axiom as follows:

- $I \models_{LP} \mathbf{T}\phi$  if and only if  $I \models_{LP} \phi$ ;
- $I \models_{LP} \mathbf{F}\phi$  if and only if  $I \models_{LP} \neg\phi$ .

Let  $\mathcal{L}^* = \{\mathbf{T}\phi, \mathbf{F}\phi \mid \phi \in \mathcal{L}\}$ . In this section, we mainly consider our tableaux based on  $\mathcal{L}^*$ . Let  $\mathbf{T}\mathcal{A} = \{\mathbf{T}\psi \mid \psi \in \mathcal{A}\}$ . A *signed literal* is a signed axiom in which the objective axiom of symbol **T** or **F** is a literal. A *signed ABox* is a set of signed concept assertions and signed role assertions. A *signed tableau* is a tree whose branches are actually sequences of signed axioms. A signed axiom is in NNF if negation ( $\neg$ ) only occurs in front of concept names occurring it.

**Definition 4** Let  $\mathcal{A}$  be a signed ABox,  $C_1, C_2, C$  concepts,  $R$  a role and  $x, y, z$  individuals in  $\mathcal{ALC}_{LP}$ . The **signed expansion rules** are defined in Table 3, where **S** is the place-holder of **T** and **F**.

In Table 3, nine signed expansion rules are introduced to construct a signed tableau for a given ABox. The first four rules similar to expansion rules of classical tableau

**Table 3.** Signed Expansion Rules in  $\mathcal{ALC}$

$\Box_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{SC}_1 \sqcap \mathbf{C}_2(x)$ , but not both $\mathbf{SC}_1(x)$ and $\mathbf{SC}_2(x)$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}_1(x), \mathbf{SC}_2(x)\}$ .
$\sqcup_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{SC}_1 \sqcup \mathbf{C}_2(x)$ , but neither $\mathbf{SC}_i(x)$ for $i = 1, 2$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}_1(x)\}$ , $\mathcal{A}'' := \mathcal{A} \cup \{\mathbf{SC}_2(x)\}$ .
$\exists_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\exists R.C(x)$ , but there is no individual name $z$ such that $\mathbf{SC}(z)$ and $\mathbf{TR}(x, z)$ are in $\mathcal{A}$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}(y), \mathbf{TR}(x, y)\}$ where $y$ is an individual name not occurring in $\mathcal{A}$ .
$\forall_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\forall R.C(x)$ and $\mathbf{TR}(x, y)$ , but is does not contain $\mathbf{SC}(y)$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}(y)\}$ .
$\neg\neg_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\neg\neg C(x)$ , but not $\mathbf{SC}(x)$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}(x)\}$ .
$\neg\Box_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\neg(C_1 \sqcap C_2)(x)$ , but neither $\mathbf{S}\neg C_i(x)$ for $i = 1, 2$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{S}\neg C_1(x)\}$ , $\mathcal{A}'' := \mathcal{A} \cup \{\mathbf{S}\neg C_2(x)\}$ .
$\neg\sqcup_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\neg(C_1 \sqcup C_2)(x)$ , but not both $\mathbf{S}\neg C_1(x)$ and $\mathbf{S}\neg C_2(x)$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{SC}_1(x), \mathbf{SC}_2(x)\}$ .
$\neg\exists_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\neg\exists R.C(x)$ and $\mathbf{TR}(x, y)$ , but is does not contain $\mathbf{S}\neg C(y)$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{S}\neg C(y)\}$ .
$\neg\forall_S$ -rule	Condition: $\mathcal{A}$ contains $\mathbf{S}\neg\forall R.C(x)$ , but there is no individual name $z$ such that $\mathbf{S}\neg C(z)$ and $\mathbf{TR}(x, z)$ are in $\mathcal{A}$ . Action: $\mathcal{A}' := \mathcal{A} \cup \{\mathbf{S}\neg C(y), \mathbf{TR}(x, y)\}$ where $y$ is an individual name not occurring in $\mathcal{A}$ .

for  $\mathcal{ALC}$ -ABoxes are applied to expand the tableau. Furthermore, the last five rules are applied to transform axioms into equivalent ones in NNF by pushing negations inwards using a combination of *De Morgan's laws* and some equivalences in Theorem 1.

**Definition 5** Let  $\mathcal{A}$  be a signed ABox. A **signed tableau** for  $\mathcal{A}$  is defined inductively according to the signed expansion rules in Table 3.

A signed tableau can be obtained from an initial signed ABox by applying the nine rules in Table 3 repeatedly. We say that a signed axiom is *marked* when a signed expansion rule is used up in it; otherwise, a signed axiom is called *non-marked*. In the following, we define the complete signed tableau. Initially, we transform a given  $\mathcal{ALC}$  ABox  $\mathcal{A}$  and a query  $\phi$  into a signed  $\mathcal{ALC}$  ABox  $\mathcal{A}'$  by computing the union of  $\{\mathbf{T}\psi \mid \psi \in \mathcal{A}\}$  and  $\{\mathbf{F}(\phi)\}$ .

**Definition 6** A branch  $\mathbf{b}$  of a signed tableau  $\mathbb{T}$  for  $\mathcal{A}$  is **complete** if and only if all non-marked axioms in  $\mathbf{b}$  are signed literals, that is to say, every signed expansion rule that can be used to extend  $\mathbf{b}$  has been applied at least once. A signed tableau  $\mathbb{T}$  for  $\mathcal{A}$  is a **complete tableau** if and only if every branch of  $\mathbb{T}$  is complete.

In the following, we define the closed signed tableau by modifying the closed conditions.

**Definition 7** A branch  $\mathbf{b}$  of a signed tableau  $\mathbb{T}$  is **closed** if and only if one of the following two conditions is satisfied:

- (1)  $\mathbf{T}A(a) \in \mathbf{b}$  and  $\mathbf{F}A(a) \in \mathbf{b}$ ; or
- (2)  $\mathbf{F}A(a) \in \mathbf{b}$  and  $\mathbf{F}\neg A(a) \in \mathbf{b}$ ;

where  $A$  is a concept name and  $a$  is an individual.

A signed tableau  $\mathbb{T}$  is **closed** if and only if every branch of  $\mathbb{T}$  is closed.

By Definition 7, the condition  $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{b}$  of closedness in signed tableaux is no longer taken as one of closed conditions since the kind of closedness is brought about because of inconsistencies in ABoxes. In short, the closed condition of signed tableaux for  $\mathcal{ALC}$ -ABoxes is different from that of classical tableau for  $\mathcal{ALC}$ -ABoxes.

Compared to classical tableau for DLs, if there exists a branch containing  $\mathbf{T}A(a)$  and  $\mathbf{T}\neg A(a)$  is closed, then the tableau is *refutation complete* for classical  $\mathcal{ALC}$  in the sense that for any signed ABox  $\mathcal{A}$ ,  $\mathcal{A}$  is inconsistent if and only if there exists a signed tableau for  $\mathcal{A}$  such that every branch of the tableau is closed. Therefore, according to the rule of *reductio ad absurdum*,  $\mathcal{A}$  paradoxically entails  $\phi$  if and only if every branch of the signed tableau for  $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}\phi\}$  is closed.

**Definition 8**  $\mathbb{T}$  is a **signed tableau** for  $\mathbf{T}A$  and  $\mathbf{F}\phi$  if the root node of  $\mathbb{T}$  contains only the set  $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}\phi\}$ . We denote  $\mathcal{A} \vdash_{LP} \phi$  if and only if the signed tableau for  $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}\phi\}$  are closed.

As mentioned earlier, the first approach to obtaining a paraconsistent logic is by restricting the rule of *reductio ad absurdum*. In  $\mathcal{ALC}_{LP}$ , we only need to weaken the closure conditions as defined above. This leads to the following soundness and completeness theorem of signed tableau w.r.t. the semantics of  $\mathcal{ALC}_{LP}$ .

**Theorem 7** Let  $\mathcal{A}$  be an ABox,  $C$  a concept and  $a$  an individual in  $\mathcal{ALC}_{LP}$ .

$$\mathcal{A} \vdash_{LP} C(a) \text{ if and only if } \mathcal{A} \models_{LP} C(a).$$

The following example illustrates the soundness and completeness in Theorem 7.

**Example 4** In Example 2, let  $\mathcal{A} = \{A(a), \neg A(a)\}$ . It can be easily checked that  $\{A(a), \neg A(a)\} \vdash_{LP} A(a)$ , since the signed tableau for  $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}(\neg A(a))\}$  has only one branch  $\{\mathbf{T}A(a), \mathbf{T}\neg A(a), \mathbf{F}\neg A(a)\}$  which is closed. However,  $\{A(a), \neg A(a)\} \not\vdash_{LP} B(a)$ , since the signed tableau for  $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}\neg B(a)\}$  has only a branch  $\{\mathbf{T}A(a), \mathbf{T}\neg A(a), \mathbf{F}\neg B(a)\}$  which is not closed.

Unfortunately, the following example shows the fact that *disjunctive syllogism* is invalid in  $\mathcal{ALC}_{LP}$ .

**Example 5** Let  $A, B$  be concepts and  $a$  an individual. It is easy to check that  $\{A(a), \neg A \sqcup B(a)\} \not\vdash_{LP} B(a)$ , since the signed tableau for  $\{\mathbf{T}A(a), \mathbf{T}\neg A \sqcup B(a), \mathbf{F}\neg B(a)\}$  has only a branch  $\{\mathbf{T}A(a), \mathbf{T}B(a), \mathbf{F}\neg B(a)\}$  which is closed, while  $\{\mathbf{T}A(a), \mathbf{T}\neg A(a), \mathbf{F}\neg B(a)\}$  is not closed. Then,  $\{A(a), \neg A \sqcup B(a)\} \not\models_{LP} B(a)$  by Theorem 7.

By Theorem 5, *disjunctive syllogism* is still valid in classical consistent ontologies. No new inference can be drawn from those possible inconsistent knowledge by using *disjunctive syllogism*. Therefore, in this paper, the paraconsistent reasoning mechanism in  $\mathcal{ALC}$  is “*brave*” under certain conditions. In particular, when we do not know which knowledge causes inconsistencies in an inconsistent ontologies, our reasoning mechanism becomes “*cautious*”. Intuitively, inferences following from an ontology, no matter whether it is inconsistent, are rather rational.

## 5 Related Work

In this section, we mainly compare  $\mathcal{ALC}_{LP}$  with  $\mathcal{ALC4}$  presented in [21] which is paraconsistent by  $\mathcal{ALC}$  integrating with Belnap’s four-valued logics [5]. The similarities and differences between LP [6], three-valued logics [11] and four-valued logics still exist between  $\mathcal{ALC}_{LP}$  and  $\mathcal{ALC4}$ .

In  $\mathcal{ALC4}$ , a 4-interpretation over domain  $\Delta^I$  assigns a pair  $\langle P, N \rangle$  each concept  $C$ , where  $P$  and  $N$  are of (neither necessarily disjoint nor necessarily mutual complement) subsets of  $\Delta^I$ . The correspondence between truth values from  $\{t, f, B, U\}$  and concept extensions is the obvious one: for instances  $a \in \Delta^I$  and concept name  $C$  we have

- $C^I(a) = t$ , if and only if  $a^I \in \text{proj}^+(C^I)$  and  $a^I \notin \text{proj}^-(C^I)$ ;
- $C^I(a) = f$ , if and only if  $a^I \notin \text{proj}^+(C^I)$  and  $a^I \in \text{proj}^-(C^I)$ ;
- $C^I(a) = B$ , if and only if  $a^I \in \text{proj}^+(C^I)$  and  $a^I \in \text{proj}^-(C^I)$ ;
- $C^I(a) = U$ , if and only if  $a^I \notin \text{proj}^+(C^I)$  and  $a^I \notin \text{proj}^-(C^I)$ .

In  $\mathcal{ALC4}$ , a 4-model of an axiom is defined as follows.

**Definition 9 ([21])** *Let  $I$  be a 4-interpretation.  $\models_4$  denotes the four-valued satisfaction relationship in  $\mathcal{ALC4}$ .*

- concept assertion:  $I \models_4 C(a)$  iff  $a^I \in \text{proj}^+(C^I)$ ;
- material inclusion:  $I \models_4 C \mapsto D$  iff  $\text{proj}^-(C^I) \subseteq \text{proj}^+(D^I)$ ;
- internal inclusion:  $I \models_4 C \sqsubseteq D$  iff  $\text{proj}^+(C^I) \subseteq \text{proj}^+(D^I)$ ;
- strong inclusion:  $I \models_4 C \rightarrow D$  iff  $\text{proj}^+(C^I) \subseteq \text{proj}^+(D^I)$  and  $\text{proj}^-(D^I) \subseteq \text{proj}^-(C^I)$ .

where  $C, D$  are concepts and  $a$  is an individual in  $\mathcal{ALC4}$ .

**Definition 10 ([21])** *Let  $\mathcal{O}$  be an ontology and  $\phi$  be an axiom in  $\mathcal{ALC4}$ .  $\phi$  is a **4-valued entailment** of  $\mathcal{O}$ , denoted by  $\mathcal{O} \models_4 \phi$ , if and only if for any 4-model  $I$  of  $\mathcal{O}$ ,  $I$  is a 4-model of  $\phi$ .*

The following theorem directly follows from Definition 1, Definition 2, Definition 9 and Definition 10.

**Theorem 8** *Let  $\mathcal{O}$  be an ontology,  $C, D$  concepts and  $a$  an individual in  $\mathcal{ALC}$ . Then*

- (1) *If  $\mathcal{O} \models_4 C(a)$  then  $\mathcal{O} \models_{LP} C(a)$ .*
- (2) *If  $\mathcal{O} \models_4 C \mapsto D$  then  $\mathcal{O} \models_{LP} C \sqsubseteq D$*

Theorem 8 shows that  $\mathcal{ALC}_{LP}$  has stronger inference power than  $\mathcal{ALC}4$ .

On the other hand, compared with the semantics of  $\mathcal{ALC}4$ , the incomplete information can also be expressed in  $\mathcal{ALC}_{LP}$ . For instance, let  $\mathcal{A} = \{C(a), \neg C(a), C \sqcup D(a), \neg C \sqcup \neg D(a)\}$ . For any 4-model  $I_4$  of  $\mathcal{A}$ ,  $D^{I_4}(a) = U$ . In fact,  $\mathcal{A} \not\models_{LP} D(a)$  and  $\mathcal{A} \not\models_{LP} \neg D(a)$ .

Another paraconsistent DL is QC  $\mathcal{ALC}$ , which is proposed in [22] by adapting Besnard and Hunter's quasi-classical logics [8] to  $\mathcal{ALC}$ . Because the QC semantics is based on the four-valued semantics, the differences between the paradoxical semantics and the four-valued semantics still retain between the paradoxical semantics and the QC semantics. Beside those differences, there are at least three differences between paradoxical DLs and QC DLs as follows.

- The QC semantics is characterized by two satisfaction relationships, namely, QC strong satisfaction and QC weak satisfaction. The QC weak satisfaction is tolerating inconsistencies while the QC strong satisfaction is enhancing the inference power. Furthermore, the concept inclusions (subsumptions) are differently defined under the two satisfaction relationships in order to capture the intuitive equivalence between concept inclusions and disjunction of concepts in QC  $\mathcal{ALC}$ . Compared with the QC semantics, it is intuitive that the paradoxical semantics is characterized by only one satisfaction (paradoxical satisfaction) and the concept inclusions is defined by the paradoxical satisfaction.
- In QC DLs [23], the tableau algorithm for ABoxes is obtained by modifying the disjunction expansion rules and changing closed conditions however the tableau algorithm for ABoxes (signed tableau algorithm) is obtained by only modifying the closed conditions in paradoxical DL  $\mathcal{ALC}$ . Thus, the signed tableau algorithm is more suitable for implementing paraconsistent reasoning.
- The *excluded middle rule* is valid in  $\mathcal{ALC}_{LP}$  by Corollary 1. However, since QC  $\mathcal{ALC}$  inherits properties of  $\mathcal{ALC}4$ , the *excluded middle law* is invalid QC  $\mathcal{ALC}$ , that is,  $\not\models_Q A \sqcup A(a)$  where  $A$  is a concept name and  $a$  is an individual (see Example 2 in [22]).

In summary, the paradoxical semantics is more suitable for dealing with inconsistent ontologies in that paraconsistent reasoning under our semantics provides a better approximation to the standard reasoning.

## 6 Conclusion and Further Work

In this paper, we have presented a non-standard DL  $\mathcal{ALC}_{LP}$ , which is a paraconsistent variant of  $\mathcal{ALC}$  by adopting the semantics of LP.  $\mathcal{ALC}_{LP}$  can be applied to formalize reasoning in the presence of inconsistency while still preserving some important inference rules such as the *excluded middle rule*. Our new logic  $\mathcal{ALC}_{LP}$  is different from the 4-valued DL  $\mathcal{ALC}4$  in that the *excluded middle rule* is not valid in  $\mathcal{ALC}4$ . Several useful properties of our logic is shown. We have also introduced a tableau for  $\mathcal{ALC}_{LP}$  and shown that it is sound and complete. This tableau algorithm has been provided to implement instance checking in ABoxes. In the future, we will develop efficient implementation of  $\mathcal{ALC}_{LP}$  by using existing efficient reasoners. It is also interesting to investigate the extension of our approach to other DLs, such as expressive DL  $\mathcal{SHION}(\mathcal{D})$ ,

which is the logical foundation of the Web Ontology Language (OWL) DL in Semantic Web, and tractable DLs, such as  $\mathcal{EL}^{++}$ , Horn-DLs, and DL-Lite.

## Acknowledgment

The authors would like to gratefully acknowledge the helpful comments and suggestions of three anonymous reviewers, which have improved the presentation. This work was supported by the Australia Research Council (ARC) Discovery Projects DP0666107.

## References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* (May 2001)
2. Bertossi, L.E., Hunter, A., Schaub, T., eds.: Inconsistency Tolerance. In Bertossi, L.E., Hunter, A., Schaub, T., eds.: *Inconsistency Tolerance*. LNCS 3300, Springer (2005)
3. Schaffert, S., Bry, F., Besnard, P., Decker, H., Decker, S., Enguix, C.F., Herzig, A.: Paraconsistent reasoning for the semantic web. In: *Proc. of the 4th International Semantic Web Conf. Workshop on Uncertainty Reasoning for the Semantic Web (ISWC-URSW 2005)*, Ireland. (2005) 104–105
4. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press (2003)
5. Belnap, N.D.: A useful four-valued logic. *Modern uses of multiple-valued logics* (1977) 7–73
6. Priest, G.: Logic of paradox. *J. of Philosophical Logic* **8** (1979) 219–241
7. Priest, G.: Reasoning about truth. *Artif. Intell.* **39**(2) (1989) 231–244
8. Besnard, P., Hunter, A.: Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information. In: *Proc. of the 3rd European Conf. on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1995)*, Switzerland. LNCS 946, Springer (1995) 44–51
9. Elvang-Gøransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. *Data Knowl. Eng.* **16**(2) (1995) 125–145
10. Benferhat, S., Dubois, D., Lang, J., Prade, H., Saffiotti, A., Smets, P.: A general approach for inconsistency handling and merging information in prioritized knowledge bases. In: *Proc. of the 6th International Conf. on Principles of Knowledge Representation and Reasoning (KR 1998)*, Italy., Morgan Kaufmann (1998) 466–477
11. Konieczny, S., Marquis, P.: Three-valued logics for inconsistency handling. In: *Proc. of the 8th European Conf. of Logics in Artificial Intelligence (JELIA 2002)*, Italy. LNCS 2424, Springer (2002) 332–344
12. Lang, J., Marquis, P.: Resolving inconsistencies by variable forgetting. In: *Proc. of the 8th International Conf. on Principles of Knowledge Representation and Reasoning (KR2002)*, France., Morgan Kaufmann (2002) 239–250
13. Coste-Marquis, S., Marquis, P.: Recovering consistency by forgetting inconsistency. In: *Proc. of the 11th European Conf. on Logics in Artificial Intelligence (JELIA 2008)*, Germany. LNAI 5293, Springer (2008) 113–125
14. Arieli, O.: Distance-based paraconsistent logics. *Int. J. Approx. Reasoning* **48**(3) (2008) 766–783
15. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: *Proc. of the 19th International Joint Conf. on Artificial Intelligence (IJCAI 2005)*, UK., Professional Book Center (2005) 454–459

16. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: Proc. of the 4th International Semantic Web Conf.(ISWC 2005), Ireland., Springer (2005) 353–367
17. Qi, G., Liu, W., Bell, D.A.: A revision-based approach to handling inconsistency in description logics. *Artif. Intell. Rev.* **26**(1-2) (2006) 115–128
18. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Proc. of the 21st International Joint Conf. on Artificial Intelligence (IJCAI 2009), USA. (2009) 891–897
19. Patel-Schneider, P.F.: A four-valued semantics for terminological logics. *Artif. Intell.* **38**(3) (1989) 319–351
20. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proc. of the 18th International Joint Conf. on Artificial Intelligence(IJCAI 2003), Mexico., Morgan Kaufmann (2003) 355–362
21. Ma, Y., Hitzler, P., Lin, Z.: Algorithms for paraconsistent reasoning with OWL. In: Proc. of the 4th European Semantic Web Conf.(ESWC 2007), Austria. LNCS 4519, Springer (2007) 399–413
22. Zhang, X., Lin, Z.: Paraconsistent reasoning with quasi-classical semantic in ALC. In: Proc. of the 2nd International Conf. of Web Reasoning and Rule Systems (RR 2008), Germany. LNCS 5341, Springer (2008) 222–229
23. Zhang, X., Xiao, G., Lin, Z.: A tableau algorithm for handling inconsistency in OWL. In: Proc. of the 6th European Semantic Web Conf.(ESWC 2009), Greece. LNCS 5554, Springer (2009) 399–413
24. I.Horrocks, U.Sattler, S.Tobies: Reasoning with individuals for the description logic SHIQ. *CoRR cs.LO/0005017* (2000)
25. Zhang, X., Zhang, Z., Lin, Z.: An argumentative semantics for paraconsistent reasoning in description logic ALC. In: Proc. of the 22nd International Workshop on Description Logics (DL 2009), UK. CEUR Workshop Proceedings 477, CEUR-WS.org (2009)
26. Lin, Z.: Three-valued nonmonotonic logic. In: Proc. of the 23rd IEEE International Symposium on Multiple-Valued Logic (ISMVL 1993), USA., IEEE Computer Society (1993) 42–47
27. Lin, Z.: Tableau systems for paraconsistency and minimal inconsistency. *J. of Computer Sci. and Technol.* **13**(2) (1998) 174–188
28. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic EL. In: Proc. of the 20th International Workshop on Description Logics (DL 2007), Italy. CEUR Workshop Proceedings 250, CEUR-WS.org (2007)
29. Baader, F., Horrocks, I., Sattler, U.: Description Logics. In van Harmelen, F., Lifschitz, V., Porter, B., eds.: *Handbook of Knowledge Representation*. Elsevier (2007)
30. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. *Artif. Intell.* **48**(1) (1991) 1–26

## Appendix: Proof Sketches

### Proof of Theorem 1

It is easy to check these properties by the definitions of complex concepts in Table 2.

### Proof of Theorem 2

For any individual  $a$  in  $\mathcal{ALC}_{LP}$ ,  $a^I \in (\neg C_1 \sqcup C_2)^I$  if and only if  $a^I \in \text{proj}^+(\neg C_1)^I$  or  $a^I \in \text{proj}^+(C_2^I)$  by Definition 1 if and only if  $a^I \notin \text{proj}^+(\neg C_1)^I$  implies  $a^I \in \text{proj}^+(C_2^I)$ , that is,  $a^I \in \Delta^I \setminus \text{proj}^-(C_1^I)$  implies  $a^I \in \text{proj}^+(C_2^I)$  if and only if  $I \models_{LP}$

$\neg C_1 \sqcup C_2(a)$ .

**Proof of Theorem 3**

Let  $I$  be a paradoxical interpretation of  $\mathcal{O}$  such that  $\text{proj}^+(C^I) \cap \text{proj}^-(C^I) = \Delta^I$  and  $R^I = \langle \Delta^I \times \Delta^I, \emptyset \times \emptyset \rangle$  for any concept  $C$  and role  $R$  occurring in  $\mathcal{O}$ . It easily shows that for any axiom  $C(a)$ ,  $R(a, b)$  and  $C \sqsubseteq D$  of  $\mathcal{O}$ ,  $I \models_{LP} C(a)$ ,  $I \models_{LP} R(a, b)$  and  $I \models_{LP} C \sqsubseteq D$  respectively where  $C, D$  are concepts,  $R$  is a role and  $a, b$  are individual in  $\mathcal{ALC}_{LP}$ . Therefore,  $I$  is a paradoxical model of  $\mathcal{O}$ .

**Proof of Theorem 4**

If  $\mathcal{O} \models_{LP} \phi$  then there exists a paradoxical interpretation  $I$  of  $\mathcal{O} \cup \{\phi\}$ . A classical interpretation  $I_c$  is obtained by constructing isomorphically a map  $\nu : I \rightarrow I_c$  by Definition 3. It easily shows that  $I_c \models \mathcal{O} \cup \{\phi\}$  by induction on the numbers of connectives occurring in  $\phi$  and the definition of satisfiability. Therefore,  $\mathcal{O} \models \phi$ . On the other hand, let  $\mathcal{O} = \{A(a), \neg A(a)\}$ .  $\mathcal{O} \models B(a)$  while  $\mathcal{O} \not\models_{LP} B(a)$  by Example 2.

**Proof of Theorem 5**

For any paradoxical interpretation  $I$  of  $\mathcal{O}$ ,  $I \models_{LP} \phi$  if and only if  $I \models \neg \phi$  since  $\mathcal{O}$  is classical consistent, that is, a classical interpretation  $I_c$ , which is obtained by constructing isomorphically a map  $\nu : I \rightarrow I_c$  by Definition 3, is  $I$  in fact. Therefore,  $\mathcal{O} \models_{LPm} \phi$  if and only if  $\mathcal{O} \models \phi$ .

**Proof of Theorem 6**

( $\Rightarrow$ ) If  $\models_{LP} \phi$  then for any paradoxical interpretation  $I$  of  $\phi$ . A classical interpretation  $I_c$  is obtained by constructing isomorphically a map  $\nu : I \rightarrow I_c$  by Definition 3. It easily shows that  $I_c \models \phi$  by induction on the numbers of connectives occurring in  $\phi$  and the definition of satisfiability. Therefore,  $\models \phi$ .

( $\Leftarrow$ ) If  $\models \phi$  then for any classical interpretation  $I_c$  of  $\phi$ . A paradoxical interpretation  $I$  is obtained by constructing isomorphically a map  $\nu : I_c \rightarrow I$  by Definition 3. It easily shows that  $I \models_{LP} \phi$  by induction on the numbers of connectives occurring in  $\phi$  and the definition of satisfiability. Therefore,  $\models_{LP} \phi$ .

**Proof of Corollary 1**

$\models_{LP} C \sqcup \neg C(a)$  by Theorem 6 since  $\models C \sqcup \neg C(a)$ .

**Proof of Corollary 2**

The proof is similar to that of Lemma 2.

**Proof of Theorem 7**

Let  $\mathbb{T}$  be a signed tableau for  $\mathbf{TA} \cup \{\mathbf{FC}(a)\}$ . We prove this property by induction of the structure of  $C$ .

( $\Rightarrow$ ) Assume that  $\mathcal{A} \vdash_{LP} C(a)$ . We need to prove that  $\mathcal{A} \models_{LP} C(a)$ .

In basic step, we consider two cases, namely, case 1.  $C$  is a concept name and case 2.  $C$  a complex concept.

**Case 1:** let  $C$  be the form  $A$  where  $A$  is a concept name. We assume that  $\mathcal{A} \vdash_{LP} A(a)$ , that is,  $\mathbf{TA} \cup \{\mathbf{FA}(a)\}$  is closed. Supposed that  $I$  is a paradoxical model of  $\mathcal{A}$  and  $I \not\models_{LP} A(a)$ , i.e.,  $\mathbf{TA}^I(a) = f$ . It easily shows that by induction on the structure of

$\mathbb{T}$  there exists a branch  $\mathbf{b}$  with respect to  $I$  in  $\mathbb{T}$ . Thus,  $\{\mathbf{T}A(a), \mathbf{F}A(a)\} \not\subseteq \mathbf{b}$  since  $\mathbf{T}A^I(a) = f$  and  $\{\mathbf{F}A(a), \mathbf{F}\neg A(a)\} \not\subseteq \mathbf{b}$  since  $\mathbf{F}\neg A(a) \notin \mathbf{b}$ . Thus,  $\mathbf{b}$  isn't closed, that is,  $\mathbf{T}A \cup \{\mathbf{F}A(a)\}$  aren't closed which contradicts our assumption. Therefore,  $\mathcal{A} \models_{LP} A(a)$ .

**Case 2:** let  $C$  has one of forms of  $\neg A, A \sqcap B, A \sqcup B, \forall R.A$  and  $\exists R.A$  where  $A$  and  $B$  are concept names. It directly follows that  $\mathcal{A} \vdash_{LP} C(a)$  by the proof of Case 1 since  $\mathbf{F}\neg A^I(a) = t \Leftrightarrow \mathbf{F}A^I(a) = f$ ,  $\mathbf{F}(A \sqcap B)^I(a) = t \Leftrightarrow \mathbf{F}\neg A^I(a) = f$  or  $\mathbf{F}\neg B^I(a) = f$ ,  $\mathbf{F}(A \sqcup B)^I(a) = t \Leftrightarrow \mathbf{F}\neg A^I(a) = f$  and  $\mathbf{F}\neg B^I(a) = f$ ,  $\mathbf{F}(\forall R.A)^I(a) = t$  and  $\mathbf{T}R^I(a, b) = t \Leftrightarrow \mathbf{F}\neg A^I(b) = f$  and  $\mathbf{F}(\exists R.A)^I(a) = t \Leftrightarrow \mathbf{F}\neg A^I(\iota) = f$  and  $\mathbf{T}R^I(a, \iota) = t$  where  $\iota$  is a new individual no occurring in  $\mathbb{T}$  before by applying paradoxical expansion rules in Table 3.

In induction step, we assume that this property holds when  $C_1$  and  $C_2$  general concepts. It analogously shows that  $\mathcal{A} \models_{LP} C(a)$  when  $C$  has the form of  $\neg C_1, C_1 \sqcap C_2, C_1 \sqcup C_2, \forall R.C_1$  and  $\exists R.C_1$  by the proof of Case 2.

Therefore, if  $\mathcal{A} \vdash_{LP} C(a)$  then  $\mathcal{A} \models_{LP} C(a)$ .

( $\Leftarrow$ ) Assume that  $\mathcal{A} \models_{LP} C(a)$ . We need to prove that  $\mathcal{A} \vdash_{LP} C(a)$ .

In basic step, we consider two case, namely, case 1.  $C$  is a concept name and case 2.  $C$  a complex concept.

**Case 1:** let  $C$  be the form  $A$  where  $A$  is a concept name. We assume that  $\mathcal{A} \models_{LP} A(a)$ , that is, for any paradoxical model  $I$  of  $\mathcal{A}$ ,  $I \models_{LP} A(a)$ . Supposed that  $\mathbf{T}A \cup \{\mathbf{F}A(a)\}$  isn't closed, that is, there exists a branch  $\mathbf{b}$  in  $\mathbb{T}$  such that  $\{\mathbf{T}A(a), \mathbf{F}A(a)\} \not\subseteq \mathbf{b}$  and  $\{\mathbf{F}A(a), \mathbf{F}\neg A(a)\} \not\subseteq \mathbf{b}$ . Since  $\mathbf{F}\neg A(a) \notin \mathbf{b}$ ,  $\mathbf{T}A(a) \notin \mathbf{b}$ . Thus there exists a paradoxical model  $I'$  with respect to  $\mathbf{b}$  of  $\mathbb{T}$  such that  $\mathbf{T}A^{I'}(a) = f$ , then  $A^{I'}(a) = f$ . It easily shows that by induction on the structure of  $\mathbb{T}$  since  $A^{I'}(a) = f$ . Thus,  $I'$  isn't a paradoxical model of  $A(a)$  which contradicts our assumption. Therefore,  $\mathcal{A} \vdash_{LP} A(a)$ .

**Case 2:** let  $C$  be of forms  $\neg A, A \sqcap B, A \sqcup B, \forall R.A$  and  $\exists R.A$  where  $A$  and  $B$  are concept names. It directly follows that  $\mathcal{A} \vdash_{LP} C(a)$  by the proof of Case 1 since  $\mathbf{F}\neg A(a) \in \mathbf{b} \Leftrightarrow \mathbf{F}A(a) \in \mathbf{b}$ ,  $\mathbf{F}(A \sqcap B)(a) \in \mathbf{b} \Leftrightarrow \mathbf{F}A(a) \in \mathbf{b}$  or  $\mathbf{F}B(a) \in \mathbf{b}$ ,  $\mathbf{F}(A \sqcup B)(a) \in \mathbf{b} \Leftrightarrow \mathbf{F}A(a) \in \mathbf{b}$  and  $\mathbf{F}B(a) \in \mathbf{b}$ ,  $\mathbf{F}(\exists R.A)(a) \in \mathbf{b}$  and  $\mathbf{T}R(a, b) \in \mathbf{b} \Leftrightarrow \mathbf{F}A(b) \in \mathbf{b}$  and  $\mathbf{F}(\forall R.A)(a) \in \mathbf{b} \Leftrightarrow \mathbf{F}A(\iota) \in \mathbf{b}$  and  $\mathbf{T}R(a, \iota) \in \mathbf{b}$  where  $\iota$  is a new individual no occurring in  $\mathbb{T}$  before by applying paradoxical expansion rules in Table 3.

In induction step, we assume that this property holds when  $C_1$  and  $C_2$  general concepts. It analogously shows that  $\mathcal{A} \vdash_{LP} C(a)$  when  $C$  has the form of  $\neg C_1, C_1 \sqcap C_2, C_1 \sqcup C_2, \forall R.C_1$  and  $\exists R.C_1$  by the proof of Case 2.

Therefore, if  $\mathcal{A} \models_{LP} C(a)$  then  $\mathcal{A} \vdash_{LP} C(a)$ .