A Tableau Algorithm for Paraconsistent and Nonmonotonic Reasoning in Description Logic-Based System

Xiaowang Zhang[1], Zuoquan Lin[1], and Kewen Wang[2]

[1] School of Mathematical Sciences, Peking University, China
[2] School of Information and Communication Technology, Griffith University, Australia

Abstract. This paper proposes a paraconsistent and nonmonotonic extension of description logic by planting a nonmonotonic mechanism called *minimal inconsistency* in paradoxical description logics, which is a paraconsistent version of description logics. A precedence relation between two paradoxical models of knowledge bases is firstly introduced to obtain minimally paradoxical models by filtering those models which contain more inconsistencies than others. A new entailment relationship between a KB and an axiom characterized by minimal paradoxical models is applied to characterize the semantics of a paraconsistent and nonmonotonic description logic. An important advantage of our adaptation is simultaneously overtaking proverbial shortcomings of existing two kinds extensions of description logics: the weak inference power of paraconsistent description logics and the incapacity of nonmonotonic description logics in handling inconsistencies. Moreover, our paraconsistent and nonmonotonic extension not only preserves the syntax of description logic but also maintains the decidability of basic reasoning problems in description logics. Finally, we develop a sound and complete tableau algorithm for instance checking with the minimally paradoxical semantics.

1 Introduction

Description logics (DLs) [1] are a family of formal knowledge representation languages which build on classical logic and are the logic formalism for Frame-based systems and Semantic Networks. E.g. DLs are the logical foundation of the Web Ontology Language (OWL) in the Semantic Web [2] which is conceived as a future generation of the World Wide Web (WWW). As is well known, ontologies or knowledge bases (KBs) in an open, constantly changing and collaborative environment might be not prefect for a variety of reasons, such as modeling errors, migration from other formalisms, merging ontologies, ontology evolution and epistemic limitation etc [3,4,5,6,7]. That is, it is unrealistic to expect that real ontologies are always logically consistent and complete. However, DLs, like classical logics, are not good enough to represent some non-classical features of real ontologies [4] such as paraconsistent reasoning and nonmonotonic reasoning.

In order to capture these non-classical features of ontologies or KBs, several extensions of DLs have been proposed. They can be roughly classified into two two categories. The first (called *paraconsistent approach*) is extending paraconsistent semantics into DLs to tolerate inconsistencies occurring in ontologies, e.g., based on

Belnap's four-valued logic [8,9], Besnard and Hunter's quasi-classical logic [10], Elvang-Gøransson and Hunter's argumentative logic [11] and Priest's paradoxical logic [12]. While these paraconsistent semantics can handle inconsistencies in DLs in a way, they share the same shortcoming: their reasoning ability is too weak to infer useful information in some cases. For instance, the resolution rules do not work in four-valued DLs [8,9] and paradoxical DLs [12]; the application of proof rules are limited in a specific order in quasi-classical DLs [10]; and the proof systems are localized in sub-ontologies in argumentative DLs [11]. Moreover, the reasoning in most of existing paraconsistent DLs is monotonic and thus they are not sufficient to express evolving ontologies coming from a realistic world.

Considering a well known example about *tweety*: let $\mathcal{K}$ be a KB whose TBox is $\{Bird \sqsubseteq Fly, Bird \sqsubseteq Wing\}$ and ABox is $\{Brid(tweety), \neg Fly(tweety)\}$. In short, $\mathcal{K}$ tells us that all birds can fly, all birds have wings and tweety is a bird and cannot fly. It is easy to see that $\mathcal{K}$ is inconsistent. In our view, it might be reasonable that *tweety* has wings since the fact that *tweety* cannot fly doesn't mean that *tweety* hasn't wings, e.g., penguin has wings but it cannot fly. However, $Wing(tweety)$ could not be drawn from $\mathcal{K}$ in four-valued DLs [9] or in paradoxical DLs [12]. $Wing(tweety)$ is unknown in argumentative DLs [11]. Though $Wing(tweety)$ might be inferred in quasi-classical DLs [10], both $Bird(tweety)$ and $Fly(tweety)$ are taken as "contradiction" (both *true* and *flase*). In this sense, the quasi-classical inference might bring over contradictions. In addition, assume that we get a new information $\neg Bird(tweety)$ about *tweety*. That is, we have known that tweety is not bird. Intuitively, we would not conclude that either *tweety* can fly or *tweety* has wings. A new KB could be is obtained by adding $\neg Bird(tweety)$ in $\mathcal{K}$. However, conclusions from the new KB are the same as $\mathcal{K}$ in quasi-classical DLs. In other words, reasoning based on quasi-classical semantics cannot capture nonmonotonic feature of a true world.

The second (called *nonmonotonic approach*) is extending DLs with nonmonotonic features, e.g., based on Reiter's default logic [13], based on epistemic operators [14,15] and based on McCarthy's circumscription [16]. They provide some versions of nonmonotonic DLs. However, they are still unable to handle some inconsistent KBs because they are based on classical models. In other words, the capability of inconsistency handling is limited. For instance, in the *tweety* example, the original KB tells us that all birds can fly. When we find that penguin are birds which are unable to fly, we will usually amend the concept of bird by treating penguin as an exception (of birds) in those nonmonotonic DLs. It is impossible that we could enumerate all exceptions from incomplete KBs. Therefore, nonmonotonic mechanisms, in our view, might not be competent for deal with inconsistencies.

As argued above, paraconsistent DLs and nonmonotonic DLs have their advantages and disadvantages. It would be interesting to investigate paraconsistent nonmonotonic DLs by combining both paraconsistent and nonmonotonic approaches. While such ideas are not new in knowledge representation, it is rarely investigated how to define a paraconsistent nonmonotonic semantics for DLs. The challenge of combining paraconsistent DLs with nonmonotonic DLs is to preserve the features of classical DLs while some non-classical features such as nonmonotonicity and paraconsistency in DLs are incorported. Ideally, such a semantics should satisfy the following properties: (1) It is based on the original syntax of DLs (e.g. no new modal operators are introduced); (2) It is paraconsistent, i.e., every KB has at least one model (no matter it is consistent or inconsistent); (3) It is nonmonotonic; (4) It is still decidable. We note, however, the current approaches to incorporating nonmonotonic mechanism in DL reasoning hardly preserve the standard syntax of DLs, e.g., adding model operators in [14,15], open default rules in [13] and circumscription patten in [16]. In this sense, some proposals of introducing nonmonotonic reasoning in paraconsistent semantics have been considered in propositional logic, e.g., extending default reasoning with four-valued semantics in [17] and with quasi-classical semantics in [18]. However, it is not straightforward to adapt these proposals to DLs. Therefore, defining a paraconsistent and non-monotonic semantics for DLs that is of sufficient expressivity and computationally tractable is a non-trivial task.

In this paper, we propose a nonmonotonic paraconsistent semantics, called *minimally paradoxical semantics*, for DLs. The major idea is to introduce a concept of minimal inconsistency presented in the logic of paradox [19] in DLs so that some undesirable models of DL KBs that are major source of inconsistency are removed. Specifically, given a DL knowledge base, we first define a precedence relationship between paradoxical models of the ontology by introducing a partial order w.r.t. a principle of minimal inconsistency, i.e., the ontology satisfying inconsistent information as little as possible. Based on this partial order, we then select all minimally models from all paradoxical models as candidate models to serve for characterizing our new inference. We can show that our inference is paraconsistent and nonmonotonic. Thus nonmonotonic feature is extended in paraconsistent reasoning of DLs without modifying the syntax of DLs. Furthermore, we develop a decidable, sound and complete tableau algorithm for ABoxes to implement answering queries. In this paper, though we mainly consider DL $\mathcal{ALC}$ since it is a most basic member of the DL family, we also argue that our technique for $\mathcal{ALC}$ can be generalized in the other expressive DL members. The paraconsistent logic adopted in the paper is the paradoxical DL $\mathcal{ALC}$ [12] because it is closer to classical $\mathcal{ALC}$ compared to other multi-valued semantics, e.g., Belnap's four-valued semantics.

2 Description Logic $\mathcal{ALC}$ and Paradoxical Semantics

Description logics (DLs) is a well-known family of knowledge representation formalisms. In this section, we briefly recall the basic notion of logics $\mathcal{ALC}$ and the paradoxical semantics of $\mathcal{ALC}$. For more comprehensive background reasoning, we refer the reader to Chapter 2 of the DL Handbook [1] and paradoxical DLs [12].

Description Logic $\mathcal{ALC}$. Let $\mathcal{L}$ be the language of $\mathcal{ALC}$. 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 $\bot$ denote the *universal concept* and the *bottom concept* respectively. Complex $\mathcal{ALC}$ concepts C, D are constructed as follows:

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

An interpretation $\mathcal{I}_c = (\Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c})$ consisting of a non-empty domain $\Delta^{\mathcal{I}_c}$ and a mapping $\cdot^{\mathcal{I}_c}$ which maps every concept to a subset of $\Delta^{\mathcal{I}_c}$ and every role to a subset of $\Delta^{\mathcal{I}_c} \times \Delta^{\mathcal{I}_c}$, for all concepts C, D and a role$R$, satisfies conditions as follows: (1) top concept: $\top^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c}$; (2) bottom concept: $\bot^{\mathcal{I}_c} = \emptyset^{\mathcal{I}_c}$; (3) negation: $(\neg C)^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c} \setminus C^{\mathcal{I}_c}$; (4) conjunction: $(C_1 \sqcap C_2)^{\mathcal{I}_c} = C_1^{\mathcal{I}_c} \cap C_2^{\mathcal{I}_c}$; (5) disjunction: $(C_1 \sqcup C_2)^{\mathcal{I}_c} = C_1^{\mathcal{I}_c} \cup C_2^{\mathcal{I}_c}$; (6) existential restriction: $(\exists R.C)^{\mathcal{I}_c} = \{x \mid \exists y, (x, y) \in R^{\mathcal{I}_c} \text{ and } y \in C^{\mathcal{I}_c}\}$; (7) value restriction: $(\forall R.C)^{\mathcal{I}_c} = \{x \mid \forall y, (x, y) \in R^{\mathcal{I}_c} \text{ implies } y \in C^{\mathcal{I}_c}\}$

An $\mathcal{ALC}$ knowledge base (KB, for short) is a finite set of axioms formed by concepts, roles and individuals. A concept assertion is an axiom of the form $C(a)$ that assigns membership of an individual a to a concept C. A role assertion is an axiom of the form $R(a, b)$ that assigns a directed relation between two individuals a, b by the role R. An *ABox* contains a finite set of concept assertions and role assertions. A concept inclusion is an axiom of the form $C_1 \sqsubseteq C_2$ that states the subsumption of the concept C_1 by the concept C_2. A *TBox* contains a finite set of concept inclusions. An KB contains an ABox and a TBox. An interpretation $\mathcal{I}_c$ satisfies a concept assertion $C(a)$ if $a^{\mathcal{I}_c} \in C^{\mathcal{I}_c}$, a role assertion $R(a, b)$ if $(a^{\mathcal{I}_c}, b^{\mathcal{I}_c}) \in R^{\mathcal{I}_c}$, a concept inclusion $C_1 \sqsubseteq C_2$if $C_1^{\mathcal{I}_c} \subseteq C_2^{\mathcal{I}_c}$. An interpretation that satisfies all axioms of a KB is called a *model* of the KB. Given a KB $\mathcal{K}$, we use $Mod(\mathcal{K})$ to denote a set of all models of $\mathcal{K}$. A KB $\mathcal{K}$ is *inconsistent* iff $Mod(\mathcal{K}) = \emptyset$. We say a KB $\mathcal{K}$ entails an axiom ϕ iff $Mod(\mathcal{K}) \subseteq Mod(\{\phi\})$, denoted $\mathcal{K} \models \phi$.

Paradoxical Semantics for $\mathcal{ALC}$. Compared with classical two-valued ($\{t, f\}$) semantics, the paradoxical semantics is three-valued ($\{t, f, \ddot{\top}\}$, $\ddot{\top}$ expressing both *true* and *false*) semantics where each concept C is interpreted as a pair $\langle +C, -C\rangle$ of (not necessarily disjoint) subsets of a domain $\Delta^{\mathcal{I}}$ and the union of them covering whole domain, i.e., $+C \cup -C = \Delta^{\mathcal{I}}$. We denote $\text{proj}^+(C^{\mathcal{I}}) = +C$ and $\text{proj}^-(C^{\mathcal{I}}) = -C$.

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 complemental 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.

Formally, a *paradoxical interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with $\Delta^{\mathcal{I}}$ as domain, where $\cdot^{\mathcal{I}}$ is a function assigning elements of $\Delta^{\mathcal{I}}$ to individuals, subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ to concepts and subsets of $(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})^2$ to roles, so that $\cdot^{\mathcal{I}}$ satisfies conditions as follows: (1) $(\top)^{\mathcal{I}} = \langle \Delta^{\mathcal{I}}, \emptyset\rangle$; (2) $(\bot)^{\mathcal{I}} = \langle \emptyset, \Delta^{\mathcal{I}}\rangle$; (3) $(\neg C)^{\mathcal{I}} = \langle \text{proj}^-(C^{\mathcal{I}}), \text{proj}^+(C^{\mathcal{I}})\rangle$; (4) $(C_1 \sqcap C_2)^{\mathcal{I}} = \langle \text{proj}^+(C_1^{\mathcal{I}}) \cap \text{proj}^+(C_2^{\mathcal{I}}), \text{proj}^-(C_1^{\mathcal{I}}) \cup \text{proj}^-(C_2^{\mathcal{I}})\rangle$; (5) $(C_1 \sqcup C_2)^{\mathcal{I}} = \langle \text{proj}^+(C_1^{\mathcal{I}}) \cup \text{proj}^+(C_2^{\mathcal{I}}), \text{proj}^-(C_1^{\mathcal{I}}) \cap \text{proj}^-(C_2^{\mathcal{I}})\rangle$; (6) $(\exists R.C)^{\mathcal{I}} = \langle\{x \mid \exists y, (x, y) \in \text{proj}^+(R^{\mathcal{I}}) \text{ and } y \in \text{proj}^+(C^{\mathcal{I}})\}, \{x \mid \forall y, (x, y) \in \text{proj}^+(R^{\mathcal{I}}) \text{ implies } y \in \text{proj}^-(C^{\mathcal{I}})\}\rangle$; (7) $(\forall R.C)^{\mathcal{I}} = \langle\{x \mid \forall y, (x, y) \in \text{proj}^+(R^{\mathcal{I}}) \text{ implies } y \in \text{proj}^+(C^{\mathcal{I}})\}, \{x \mid \exists y, (x, y) \in \text{proj}^+(R^{\mathcal{I}}) \text{ and } y \in \text{proj}^-(C^{\mathcal{I}})\}\rangle$

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

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

For instance, let $\Delta = \{a, b\}$ be a domain and A, B two concept names. Assume that $\mathcal{I}$ is a paradoxical interpretation on Δ such that $A^{\mathcal{I}} = \langle\{a, b\}, \{b\}\rangle$ and $B^{\mathcal{I}} = \langle\{a\}, \{a, b\}\rangle$. Then $A^{\mathcal{I}}(a) = t$, $A^{\mathcal{I}}(b) = B^{\mathcal{I}}(a) = \ddot{\top}$ and $B^{\mathcal{I}}(b) = f$.

A paradoxical interpretation $\mathcal{I}$ satisfies a concept assertion $C(a)$ if $a^{\mathcal{I}} \in \text{proj}^{+}(C^{\mathcal{I}})$, a role assertion $R(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in \text{proj}^{+}(R^{\mathcal{I}})$, a concept inclusion $C_1 \sqsubseteq C_2$ iff $\Delta^{\mathcal{I}} \backslash \text{proj}^{-}(C_1^{\mathcal{I}}) \subseteq \text{proj}^{+}(C_2^{\mathcal{I}})$. A paradoxical interpretation that satisfies all axioms of a KB is called a *paradoxical model* of the KB. Given a KB $\mathcal{K}$, we use $Mod^P(\mathcal{K})$ to denote a set of all paradoxical models of $\mathcal{K}$. For instance, in the above example, $\mathcal{I}$ is a paradoxical model of an ABox $\{A(a), A(b), B(a)\}$ while it is not a paradoxical model of $\{A(a), A(b), B(a), B(b)\}$. We also find that $\mathcal{I}$ is also a paradoxical model of a TBox $\{A \sqsubseteq B\}$. A KB $\mathcal{K}$ *paradoxically entails* an axiom ϕ iff $Mod^P(\mathcal{K}) \subseteq Mod^P(\{\phi\})$, denoted $\mathcal{K} \models_{LP} \phi$.

We say a paradoxical interpretation $\mathcal{I}$ is *trivial* if $A^{\mathcal{I}} = \langle\Delta^{\mathcal{I}}, \Delta^{\mathcal{I}}\rangle$ for any concept name A. Note that contradictions which have form of $A \sqcap \neg A(a)$ for some concept name A and some individual a in $\mathcal{ALC}$ are only satisfied by trivial paradoxical interpretations. In general, trivial paradoxical interpretations/models cannot provide any information about querying since all queries are answered as "$\ddot{\top}$". For simplifying discussion, we mainly consider KBs without contradictions.

3 Minimally Paradoxical Semantics for $\mathcal{ALC}$

Paradoxical description logic is proposed in [12] as a paraconsistent version of description logics. In this section, we present a *minimally paradoxical semantics* for $\mathcal{ALC}$ by introducing a nonmonotonic reasoning mechanism *minimally inconsistency* (see [20]) to extend $\mathcal{ALC}$ with both paraconsistent and nonmonotonic features. The basic idea of our proposal is to introduce a preference relation on paradoxical models and to filter those models that cause more inconsistencies. Indeed, our proposal is able to maintain more consistent information in reasoning than previous approaches.

In a paradoxical DL, every paradoxical model can be represented as a set of concept assertions and role assertions similar to classical models. Since the constructor of the negation of a role $\neg R$ is absent in $\mathcal{ALC}$ (see [9]), for simplicity to convey our idea, we mainly consider concept assertions in defining the preference relation in this paper.

Next, we introduce a partial order $\prec$ between two paradoxical interpretations to characterize their difference on concept assertions.

Definition 1. *Let $\mathcal{I}$ and $\mathcal{I}'$ be two paradoxical interpretations in $\mathcal{ALC}$. We say $\mathcal{I}$ is* more consistent *than $\mathcal{I}'$, denoted $\mathcal{I} \prec \mathcal{I}'$ iff*

- *$\mathcal{I}$ and $\mathcal{I}'$ have the same domain Δ;*
- *if $A^{\mathcal{I}}(a) = \ddot{\top}$ then $A^{\mathcal{I}'}(a) = \ddot{\top}$ for any concept name $A \in N_C$ and any individual $a \in N_I$;*
- *there exists a concept name $A \in N_c$ and an individual $a \in \Delta$ such that $A^{\mathcal{I}'}(a) = \ddot{\top}$ but $A^{\mathcal{I}}(a) \neq \ddot{\top}$.*

Intuitively, the first condition states that if $\mathcal{I}_1$ and $\mathcal{I}_2$ do not share a common domain, then they are not comparable; the second condition ensures that if $\mathcal{I}_1 \prec \mathcal{I}_2$ then $\mathcal{I}_2$

contains no less inconsistencies than $\mathcal{I}_1$ does; and the third condition shows that if $\mathcal{I}_1 \prec \mathcal{I}_2$ then $\mathcal{I}_1$ contains less inconsistencies than $\mathcal{I}_2$.

For instance, let $\Delta = \{a, b\}$ be a domain and A a concept name. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two paradoxical interpretations such that $A^{\mathcal{I}_1} = \langle\{a\}, \{b\}\rangle$ and $A^{\mathcal{I}_2} = \langle\{a, b\}, \{b\}\rangle$. Then we can easily see $\mathcal{I}_1 \prec \mathcal{I}_2$. If $\mathcal{I}_3$ is a paradoxical interpretation such that $A^{\mathcal{I}_3} = \langle\{b\}, \{a\}\rangle$. Then $\mathcal{I}_1 \not\prec \mathcal{I}_3$ and $\mathcal{I}_3 \not\prec \mathcal{I}_1$. That is, $\mathcal{I}_1$ and $\mathcal{I}_3$ are incomparable. So, in general $\prec$ is a partial order.

Note that $\prec$ is anti-reflexive, anti-symmetric and transitive. In other words, (1) $\mathcal{I} \not\prec \mathcal{I}$ for any paradoxical interpretation $\mathcal{I}$; (2) if $\mathcal{I} \prec \mathcal{I}'$ then $\mathcal{I}' \not\prec \mathcal{I}$; and (3) if $\mathcal{I} \prec \mathcal{I}'$ and $\mathcal{I}' \prec \mathcal{I}''$ then $\mathcal{I} \prec \mathcal{I}''$.

We denote $\mathcal{I} \preceq \mathcal{I}'$ as either $\mathcal{I} \prec \mathcal{I}'$ or $\mathcal{I} = \mathcal{I}'$. So $\preceq$ is also a partial order.

Definition 2. *Let $\mathcal{K}$ be a KB in $\mathcal{ALC}$. A paradoxical model $\mathcal{I}$ of $\mathcal{K}$ is* minimal *if there exists no other paradoxical model $\mathcal{I}'$ of $\mathcal{K}$ such that $\mathcal{I}' \prec \mathcal{I}$. We use $Mod^P_{min}(\mathcal{K})$ to denote the set of all minimal paradoxical models of $\mathcal{K}$.*

Intuitively, minimal paradoxical models are paradoxical models which contain minimal inconsistency. Since a non-empty KB always has a paradoxical model, it also has a minimal paradoxical model. That is, no matter whether a non-empty KB $\mathcal{K}$ is consistent or not, we have $Mod^P_{min}(\mathcal{K}) \neq \emptyset$ while $Mod(\mathcal{K}) = \emptyset$ for any inconsistent KB $\mathcal{K}$.

Example 1. *Let $\mathcal{K} = (\{A \sqsubseteq B\}, \{A(b), \forall R.\neg B(a), R(a, b)\})$ be a KB and $\Delta = \{a, b\}$ be a domain. We assume $\mathcal{I}$ is a paradoxical interpretation such that $A^{\mathcal{I}} = \langle\{b\}, \emptyset\rangle$, $B^{\mathcal{I}} = \langle\{b\}, \{b\}\rangle$, and $R^{\mathcal{I}} = \langle\{(a, b)\}, \emptyset\rangle$. It can be verified that $\mathcal{I}$ is the only minimally paradoxical model of $\mathcal{K}$. Thus $Mod^P_{min}(\mathcal{K}) = \{\mathcal{I}\}$.*

For a consistent KB, we can show that the set of its classical models corresponds to the set of its minimally paradoxical models. To this end, we first show how to transform each classical interpretation $\mathcal{I}_c$ into a paradoxical interpretation $\mathcal{I}$ as follows: for a concept name $A \in N_C$, a role name $R \in N_R$ and an individual name $a \in N_I$, define

$$\begin{cases} a^{\mathcal{I}} = a^{\mathcal{I}_c}, & \text{for any individual } a \in N_I; \\ a^{\mathcal{I}} \in +A, & \text{if } a^{\mathcal{I}_c} \in A^{\mathcal{I}_c}; \\ a^{\mathcal{I}} \in -A, & \text{if } a^{\mathcal{I}_c} \notin A^{\mathcal{I}_c}; \\ (a^{\mathcal{I}}, b^{\mathcal{I}}) \in +R \text{ and } -R = \emptyset, & \text{if } (a^{\mathcal{I}_c}, b^{\mathcal{I}_c}) \in R^{\mathcal{I}_c}. \end{cases} \tag{1}$$

For instance, let $\Delta = \{a, b\}$ and $\mathcal{I}_c$ be a classical interpretation such that $A^{\mathcal{I}_c} = \{a\}$ and $B^{\mathcal{I}_c} = \{a, b\}$. We transform $\mathcal{I}_c$ into a paradoxical interpretation $\mathcal{I}$ where $A^{\mathcal{I}} = \langle\{a\}, \{b\}\rangle$ and $B^{\mathcal{I}} = \langle\{a, b\}, \emptyset\rangle$.

If $\mathcal{K}$ is consistent, we use $Mod^T(\mathcal{K})$ to denote the collection of all paradoxical models of $\mathcal{K}$ transformed as above. It is interesting that each classical model corresponds to exactly one minimally paradoxical model for a consistent KB.

Proposition 1. *For any consistent $\mathcal{ALC}$ KB $\mathcal{K}$, $Mod^P_{min}(\mathcal{K}) = Mod^T(\mathcal{K})$.*

To see the validity of this result, we note that every paradoxical model in $Mod^T(\mathcal{K})$ is minimal because paradoxical models of a consistent KB are incomparable.

Based on the notion of minimally paradoxical models, we can define the following entailment relation.

Definition 3. *Let $\mathcal{K}$ be a KB and ϕ an axiom in $\mathcal{ALC}$. We say $\mathcal{K}$* minimally paradoxically entails *ϕ iff $Mod^P_{min}(\mathcal{K}) \subseteq Mod^P_{min}(\{\phi\})$, denoted by $\mathcal{K} \models^m_{LP} \phi$.*

Intuitively, the minimally paradoxical entailment ($\models^m_{LP}$) characterizes an inference relation from a KB to an axiom by their minimally paradoxical models. Because the inference focuses on those paradoxical models in which inconsistency is minimized, $\models^m_{LP}$ can give consideration to both the classical entailment $\models$ and the paradoxical entailment $\models_{LP}$. When a KB is consistent, $\models^m_{LP}$ is equivalent to $\models$ since no model does contain an inconsistency. When a KB is inconsistent, $\models^m_{LP}$ inherits $\models_{LP}$ since every minimally paradoxical model contains at least an inconsistency. In this sense, $\models^m_{LP}$ is more reasonable than $\models$ and $\models_{LP}$. For instance, in the example about *tweety* (presented in Section 1). We have $\mathcal{K} \models^m_{LP} Wing(tweety)$ and $\mathcal{K} \cup \{\neg Bird(tweety)\} \not\models^m_{LP} Wing(tweety)$.

Note that the minimally paradoxical entailment is determined by restricting paradoxical entailment to the subclass of minimal paradoxical models. Thus the entailment relation is nonmonotonic.

For instance, given an ABox $\mathcal{A} = \{A(a), \neg A \sqcup B(a)\}$ where A, B are concept names and $\Delta = \{a\}$ a domain. Let $\mathcal{I}$ be a paradoxical interpretation s.t. $A^{\mathcal{I}} = \langle \{a\}, \emptyset \rangle$ and $B^{\mathcal{I}} = \langle \{a\}, \emptyset \rangle$. It easily check $\mathcal{I}$ is only one minimally paradoxical model of $\mathcal{A}$. Then $\{A(a), \neg A \sqcup B(a)\} \models^m_{LP} B(a)$. However, if we assume that $\mathcal{A}' = \mathcal{A} \cup \{\neg A(a)\}$ then there exist three minimally paradoxical models of $\mathcal{A}'$ $\mathcal{I}_i (i = 1, 2)$ where $A^{\mathcal{I}_i} = \langle \{a\}, \{a\} \rangle$ and $B^{\mathcal{I}_1} = \langle \{a\}, \emptyset \rangle$ and $B^{\mathcal{I}_2} = \langle \emptyset, \{a\} \rangle$. Thus $\mathcal{A}' \not\models^m_{LP} B(a)$.

In addition, $\models^m_{LP}$ is paraconsistent because each consistent KB has at least one minimally paradoxical model.

Theorem 1. *$\models^m_{LP}$ is paraconsistent and nonmonotonic.*

The next result shows that, if a KB is consistent, then the entailment $\models^m_{LP}$ coincides with the classical entailment.

Proposition 2. *Let $\mathcal{K}$ be a consistent KB and ϕ an axiom in $\mathcal{ALC}$. Then*

$$\mathcal{K} \models^m_{LP} \phi \text{ iff } \mathcal{K} \models \phi.$$

This proposition directly follows from Proposition 1. Note that for an inconsistent KB, $\models^m_{LP}$ differs from $\models$. Thus, our new entailment relation naturally extends the classical entailment to all KBs in $\mathcal{ALC}$ while the classical reasoning in consistent KBs is still preserved.

Under classical entailment, anything can be inferred from an inconsistent KB. Thus, it is straightforward to see the following corollary.

Corollary 1. *Let $\mathcal{K}$ be a KB and ϕ an axiom in $\mathcal{ALC}$. If $\mathcal{K} \models^m_{LP} \phi$ then $\mathcal{K} \models \phi$.*

However, the converse of Corollary 1 is not true in general when $\mathcal{K}$ is inconsistent. For instance, we have $\{A(a), \neg A(a), \neg A \sqcup B(a), \neg B(a)\} \models B(a)$
while $\{A(a), \neg A(a), \neg A \sqcup B(a), \neg B(a)\} \not\models^m_{LP} B(a)$.

The resolution rule is important for automated reasoning in DLs. It is well-know that the inference rules *modus ponens*, *modus tollens* and *disjunctive syllogism* special cases of the *resolution rule*.

The resolution rule is not valid for $\models^m_{LP}$ in the following sense, while it is invalid in paradoxical DLs. In other words, the inference power of paradoxical DLs is strengthened by the concept of minimally paradoxical models.

Proposition 3. *Let C, D, E be concepts and a an individual in $\mathcal{ALC}$.*

$$\{C \sqcup D(a), \neg C \sqcup E(a)\} \models^m_{LP} D \sqcup E(a).$$

However, the resolution rule is not valid in general under minimally paradoxical semantics. For instance, $\{A(a), \neg A(a), A \sqcup B(a)\} \not\models^m_{LP} B(a)$.

We remark that minimally paradoxical semantics does not only preserve the reasoning ability of classical semantics for consistent knowledge but also tolerate inconsistencies (possibly) occurring KBs.

The results shown in this section demonstrate that the entailment relation $\models^m_{LP}$ is much better than the paradoxical entailment defined in [12].

4 Minimal Signed Tableau Algorithm

The minimally paradoxical semantics introduced in last section is based on minimal paradoxical models. A naive algorithm for reasoning under the new paraconsistent semantics could be developed by finding all minimal models from (possibly infinite) paradoxical models of a KB. However, such an algorithm would be very inefficient if it is not impossible. Instead, in this section, we develop a tableau algorithm for the new semantics. Tableau algorithms are widely used for checking satisfiability in DLs. Especially, *signed tableau algorithm*) has been developed in [12] for paradoxical $\mathcal{ALC}$. Our new tableau algorithm for minimally paradoxical semantics is obtained by embedding the minimality condition into the signed tableau algorithm. The challenge of doing so is how to find redundant clashes (i.e., a clash is caused by an inconsistency) and remove them from the signed tableau.

We first briefly recall the signed tableau algorithm for instance checking in ABoxes (the details can be found in [12]). The signed tableau algorithm is based on the notion of signed concepts. Note that roles are not signed because they represent edges connecting two nodes in tableaux. A sign concept is either $\mathbf{T}C$ or $\mathbf{F}C$ where the concept C is in NNF (i.e., negation ($\neg$) only occurs in front of concept names). Each signed concept can be transformed into its NNF by applying De Morgan's law, distributive law, the law of double negation and the following rewriting rules:

$$\mathbf{T}(C \sqcap D) = \mathbf{T}C \sqcap \mathbf{T}D,\ \mathbf{T}(C \sqcup D) = \mathbf{T}C \sqcup \mathbf{T}D,\ \mathbf{T}\forall R.C = \forall R.\mathbf{T}C,\ \mathbf{T}\exists R.C = \exists R.\mathbf{T}C$$
$$\mathbf{F}(C \sqcap D) = \mathbf{F}C \sqcup \mathbf{F}D,\ \mathbf{F}(C \sqcup D) = \mathbf{F}C \sqcap \mathbf{F}D,\ \mathbf{F}\forall R.C = \exists R.\mathbf{F}C,\ \mathbf{F}\exists R.C = \forall R.\mathbf{F}C$$

We use $\mathbf{T}\mathcal{A}$ to denote a signed ABox whose concept names are marked with $\mathbf{T}$, i.e., $\mathbf{T}\mathcal{A} = \{\mathbf{T}C \mid C \in \mathcal{A}\}$. A *signed tableau* is a forest whose trees are actually composed of nodes $\mathcal{L}(x)$ containing signed concepts and edges $\mathcal{L}(x, y)$ containing role names. Given an ABox $\mathcal{A}$ and an axiom $C(a)$, the signed tableau algorithm starts with $\mathcal{F}_{\mathcal{A}}$ as the initial forest of $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}C(a)\}$. The algorithm then applies the signed expansion rules, which are reformulated in Table 1. The algorithm terminates if it encounters a clash: $\{\mathbf{T}A, \mathbf{F}A\} \subseteq \mathcal{L}(x)$ or $\{\mathbf{F}A, \mathbf{F}\neg A\} \subseteq \mathcal{L}(x)$ where A is a concept name. Finally,

Table 1. Expansion Rules in Signed Tableau

$\sqcap_{\mathbf{T}}$-rule	If: $\mathbf{T}(C_1 \sqcap C_2) \in \mathcal{L}$, but not both $\mathbf{T}C_1 \in \mathcal{L}(x)$ and $\mathbf{T}C_2 \in \mathcal{L}(x)$. Then: $\mathcal{L}(x) := \mathcal{L}(x) \cup \{\mathbf{T}C_1, \mathbf{T}C_2\}$.
$\sqcup_{\mathbf{T}}$-rule	If: $\mathbf{T}(C_1 \sqcup C_2) \in \mathcal{L}(x)$, but neither $\mathbf{T}C_i \in \mathcal{L}(x)$ for $i = 1, 2$. Then: $\mathcal{L}(x) := \mathcal{L}(x) \cup \{\mathbf{T}C_i\}$ ($i = 1$ or 2).
$\exists_{\mathbf{T}}$-rule	If: $\mathbf{T}\exists R.C \in \mathcal{L}(x)$, but there is no node $\mathcal{L}(z)$ s.t. $\mathbf{T}C \in \mathcal{L}(z)$ and $R \in \mathcal{L}(x, z)$. Then: create a new node $\mathcal{L}(y) := \{\mathbf{T}C\}$ and $\mathcal{L}(x, y) := \{R\}$.
$\forall_{\mathbf{T}}$-rule	If: $\mathbf{T}\forall R.C \in \mathcal{L}(x)$ and $R \in \mathcal{L}(x, y)$, but $\mathbf{T}C \notin \mathcal{L}(y)$. Then: $\mathcal{L}(y) := \mathcal{L}(y) \cup \{\mathbf{T}C\}$.
$\sqcap_{\mathbf{F}}$-rule	If: $\mathbf{F}(C_1 \sqcap C_2) \in \mathcal{L}(x)$, but neither $\mathbf{F}C_i \in \mathcal{L}(x)$ for $i = 1, 2$. Then: $\mathcal{L}(x) := \mathcal{L}(x) \cup \{\mathbf{F}C_i\}$ ($i = 1$ or 2).
$\sqcup_{\mathbf{F}}$-rule	If: $\mathbf{F}(C_1 \sqcup C_2) \in \mathcal{L}$, but not both $\mathbf{F}C_1 \in \mathcal{L}(x)$ and $\mathbf{F}C_2 \in \mathcal{L}(x)$. Then: $\mathcal{L}(x) := \mathcal{L}(x) \cup \{\mathbf{F}C_1, \mathbf{F}C_2\}$.
$\exists_{\mathbf{F}}$-rule	If: $\mathbf{F}\exists R.C \in \mathcal{L}(x)$ and $R \in \mathcal{L}(x, y)$, but $\mathbf{F}C \notin \mathcal{L}(y)$. Then: $\mathcal{L}(y) := \mathcal{L}(y) \cup \{\mathbf{F}C\}$.
$\forall_{\mathbf{F}}$-rule	If: $\mathbf{F}\forall R.C \in \mathcal{L}(x)$, but there is no node $\mathcal{L}(z)$ s.t. $\mathbf{F}C \in \mathcal{L}(z)$ and $R \in \mathcal{L}(x, z)$. Then: create a new node $\mathcal{L}(y) := \{\mathbf{F}C\}$ and $\mathcal{L}(x, y) := \{R\}$.

we obtain a completion forest The problem whether $\mathcal{A}$ paradoxically entails $C(a)$ is decided by checking whether the completion forest is closed, i.e., checking whether every tree of the completion forest contains at least one clash. The algorithm preserves a so-called *forest model* property, i.e., the paradoxical model has the form of a set of (potentially infinite) trees, the root nodes of which can be arbitrarily interconnected.

In the following, we develop a preference relation on trees of the completion forest to eliminating the trees with redundant inconsistencies.

Let $\mathcal{F}$ be a completion forest and $\mathbf{t}$ a tree of $\mathcal{F}$. We denote $IC(\mathbf{t}) = \{\mathbf{T}A \mid \{\mathbf{T}A, \mathbf{T}\neg A\} \subseteq \mathcal{L}(x)$ for some node $\mathcal{L}(x) \in \mathbf{t}\}$. Intuitively, $IC(\mathbf{t})$ is the collection of contradictions in $\mathbf{t}$.

Definition 4. *Let* $\mathbf{t}_1$ *and* $\mathbf{t}_2$ *be two trees of a identical completion forest. We denote* $\mathbf{t}_1 \prec_{IC} \mathbf{t}_2$ *if* $IC(\mathbf{t}_1) \subset IC(\mathbf{t}_2)$ *and* $\mathbf{t}_1 \preceq_{IC} \mathbf{t}_2$ *if* $IC(\mathbf{t}_1) \subseteq IC(\mathbf{t}_2)$. *If* $\mathbf{t}_1 \prec_{IC} \mathbf{t}_2$ *then we say* $\mathbf{t}_2$ *is* redundant *w.r.t.* $\mathbf{t}_1$.

Intuitively, if $\mathbf{t}_2$ is redundant w.r.t. $\mathbf{t}_1$ then $\mathbf{t}_2$ contains more inconsistencies than $\mathbf{t}_1$ does.

A tree $\mathbf{t}$ of $\mathcal{F}$ is a *minimally redundant tree* of $\mathcal{F}$ if $\mathbf{t} \preceq_{IC} \mathbf{t}'$ for each tree $\mathbf{t}'$ in $\mathcal{F}$ and there is not any other tree $\mathbf{t}''$ in $\mathcal{F}$ s.t. $\mathbf{t}'' \prec_{IC} \mathbf{t}$. A minimally redundant tree is a tree we want to keep. A *minimal completion forest* of $\mathcal{F}$, denoted by $\mathcal{F}^m$, is composed of all minimal trees of $\mathcal{F}$. It can be easily verfied that $\mathcal{F}^m$ always exists. Given an ABox $\mathcal{A}$ and an axiom $C(a)$, the process of computing the minimal completion forest of $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}C(a)\}$ is called the *minimal signed tableau algorithm.*

A paradoxical interpretation $\mathcal{I}$ *satisfies* a tree $\mathbf{t}$ iff for any node $\mathcal{L}(x)$ and any edge $\mathcal{L}(x, y)$ in $\mathbf{t}$, we have
(1) $x^{\mathcal{I}} \in \text{proj}^+(A^{\mathcal{I}})$ if $\mathbf{T}A \in \mathcal{L}(x)$ and $x^{\mathcal{I}} \in \text{proj}^-(A^{\mathcal{I}})$ if $\mathbf{T}\neg A \in \mathcal{L}(x)$;
(2) $x^{\mathcal{I}} \in \Delta^{\mathcal{I}} - \text{proj}^+(A^{\mathcal{I}})$ if $\mathbf{F}A \in \mathcal{L}(x)$ and $x^{\mathcal{I}} \in \Delta^{\mathcal{I}} - \text{proj}^-(A^{\mathcal{I}})$ if $\mathbf{F}\neg A \in \mathcal{L}(x)$;
(3) $(x^{\mathcal{I}}, y^{\mathcal{I}}) \in \text{proj}^+(R^{\mathcal{I}})$ if $R \in \mathcal{L}(x, y)$.

Based on the above definition, it follows that if $\mathcal{I}$ is a paradoxical interpretation satisfying $\mathbf{t}$ then $A^{\mathcal{I}}(a) = \ddot{\top}$ iff $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{t}$ for any concept name $A \in N_C$ and any individual $a \in N_I$. As a result, there exists a close relation between $\prec$ (defined over paradoxical models) and $\prec_{IC}$ (defined over trees).

Theorem 2. *Let* $\mathbf{t}$ *and* $\mathbf{t}'$ *be two trees of a completion forest* $\mathcal{F}$. *If* $\mathcal{I}$ *and* $\mathcal{I}'$ *be two paradoxical interpretations satisfying* $\mathbf{t}$ *and* $\mathbf{t}'$ *respectively, then* $\mathcal{I} \prec \mathcal{I}'$ *iff* $\mathbf{t} \prec_{IC} \mathbf{t}'$.

Proof.(Sketch) Note that the ABox contains only concept assertions. We show only that the theorem is true for atomic concept A. It is straightforward to prove that the conclusion is also true for a complex concept by induction.

($\Leftarrow$) If $\mathbf{t} \prec_{IC} \mathbf{t}'$, then $IC(\mathbf{t}) \subset IC(\mathbf{t}')$. We want to show that $\mathcal{I} \prec \mathcal{I}'$. Assume that $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{t}$ then $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{t}'$. Thus $A^{\mathcal{I}}(a) = \ddot{\top}$ implies $A^{\mathcal{I}'}(a) = \ddot{\top}$. There is a concept assertion $B(b)$, where B is a concept name and b an individual name, such that $\{\mathbf{T}B(b), \mathbf{T}B(b)\} \subseteq \mathbf{t}'$ but $\{\mathbf{T}B(b), \mathbf{T}B(b)\} \not\subseteq \mathbf{t}$. Thus $B^{\mathcal{I}'}(b) = \ddot{\top}$ implies $B^{\mathcal{I}}(a) \neq \ddot{\top}$. Therefore, $\mathcal{I} \prec \mathcal{I}'$ by Definition 1.

($\Rightarrow$) If $\mathcal{I} \prec \mathcal{I}'$, we need to show that $\mathbf{t} \prec_{IC} \mathbf{t}'$, i.e., $IC(\mathbf{t}) \subset IC(\mathbf{t}')$. For any concept assertion $A(a)$, if $A^{\mathcal{I}}(a) = \ddot{\top}$ then $A^{\mathcal{I}'}(a) = \ddot{\top}$. Thus $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{t}$ then $\{\mathbf{T}A(a), \mathbf{T}\neg A(a)\} \subseteq \mathbf{t}'$. There is a concept assertion $B(b)$, where B is a concept name and b an individual name, such that $B^{\mathcal{I}'}(b) = \ddot{\top}$ but $B^{\mathcal{I}}(a) \neq \ddot{\top}$. Thus $\{\mathbf{T}B(b), \mathbf{T}B(b)\} \subseteq \mathbf{t}'$ but $\{\mathbf{T}B(b), \mathbf{T}B(b)\} \not\subseteq \mathbf{t}$. Therefore, $\mathbf{t} \prec_{IC} \mathbf{t}'$ since $IC(\mathbf{t}) \subset IC(\mathbf{t}')$.

Now we are ready to define the concept of minimally closed completion forests. A completion forest $\mathcal{F}$ is *minimally closed* iff every tree of $\mathcal{F}^m$ is closed.

If the completion forest of $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}C(a)\}$ by applying the minimal signed tableau algorithm is minimally closed, then we write $\mathcal{A} \vdash^m_{LP} C(a)$.

We show that our minimal signed tableau algorithm is sound and complete.

Theorem 3. *Let* $\mathcal{A}$ *be an ABox and* $C(a)$ *an axiom in* $\mathcal{ALC}$. *We have*

$$\mathcal{A} \vdash^m_{LP} C(a) \text{ iff } \mathcal{A} \models^m_{LP} C(a).$$

Proof.(Sketch) We consider only atomic concept A here. Let $\mathcal{F}$ be a completion forest for $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}A(a)\}$ by applying the minimal signed tableau algorithm. We need to prove that $\mathcal{F}$ is minimally closed, i.e., every tree of the minimal forest $\mathcal{F}^m$ is closed, iff $\mathcal{A} \models_{LPm} A(a)$.

($\Rightarrow$) Assume that $\mathcal{F}$ is minimally closed. On the contrary, supposed that $\mathcal{A} \not\models^m_{LP} A(a)$, that is, there exists a minimally paradoxical model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{I} \not\models^m_{LP} A(a)$, then $\mathcal{I} \not\models_{LP} A(a)$ since every minimal paradoxical model is always a paradoxical model. There exists a tree $\mathbf{t}$ which is satisfied by $\mathcal{I}$ in $\mathcal{F}$ and $\mathbf{t}$ is not closed by the proof of Theorem 7 which states that the signed tableau algorithm is sound and complete w.r.t. paradoxical semantics for $\mathcal{ALC}$ (see [12]). We assert that $\mathbf{t}$ is not redundant. Otherwise, there might be another tree $\mathbf{t}' \prec_{IC} \mathbf{t}$. We define a paradoxical interpretation $\mathcal{I}'$ satisfing $\mathbf{t}'$. It is easy to see that $\mathcal{I}'$ and $\mathcal{I}$ have the same domain by induction on the structure of complete $\mathcal{F}$. By Theorem 2, we have $\mathcal{I}' \prec \mathcal{I}$, which contradicts the minimality of $\mathcal{I}$. Thus the completion forest $\mathcal{F}$ for $\mathbf{T}\mathcal{A} \cup \{\mathbf{F}A(a)\}$ contains at least a tree $\mathbf{t}$ that is neither closed nor redundant, which contadicts with the assumption that $\mathcal{F}$ is minimally closed. Thus $\mathcal{A} \models^m_{LP} A(a)$.

($\Leftarrow$) Let $\mathcal{A} \models^m_{LP} A(a)$. On the contrary, supposed that $\mathcal{F}$ is not minimally closed. Then there exists a tree $\mathbf{t}$ of complete $\mathcal{F}$ such that $\mathbf{t}$ is neither closed nor redundant. Since $\mathbf{t}$ is not closed, by the proof of Theorem 7 (see [12]), we can construct a paradoxical interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a paradoxical model of $\mathcal{A}$. However, $\mathcal{I} \not\models_{LP} A(a)$, a contradiction. Supposed that $\mathcal{I}$ is not minimal, there exists a paradoxical model $\mathcal{I}'$ of $\mathcal{A}$ such that $\mathcal{I}' \prec \mathcal{I}$ in the same domain. Thus there exists a tree $\mathbf{t}'$ that is satisfied by $\mathcal{I}'$ with $\mathbf{t}' \prec_{IC} \mathbf{t}$ by Theorem 2, which contradicts to the assumption that $\mathbf{t}$ is not redundant. Then $\mathcal{I} \not\models_{LP} A(a)$, i.e., $\mathcal{I} \notin Mod^P(\{A(a)\})$. Because $\{A(a)\}$ is consistent and A is a concept name, $Mod^P(\{A(a)\}) = Mod^P_{min}(\{A(a)\})$. Thus $\mathcal{I}$ is a minimally paradoxical model of $\mathcal{A}$ but $\mathcal{I} \not\models^m_{LP} A(a)$. Therefore, $\mathcal{F}$ is minimally closed.

Example 2. *Let $\mathcal{A} = \{C \sqcup D(a), \neg C \sqcup E(a)\}$ be an ABox and $D \sqcup E(a)$ an axiom. There are four trees of the completion forest $\mathcal{F}$ of $\{\mathbf{T}\mathcal{A} \cup \{\mathbf{F}(D \sqcup E)(a)\}$ where*
$\mathbf{t}_1 = \{\mathbf{T}C(a), \mathbf{T}\neg C(a), \mathbf{F}D(a), \mathbf{F}E(a)\}$; $\mathbf{t}_2 = \{\mathbf{T}C(a), \mathbf{T}E(a), \mathbf{F}D(a), \mathbf{F}E(a)\}$;
$\mathbf{t}_3 = \{\mathbf{T}D(a), \mathbf{T}\neg C(a), \mathbf{F}D(a), \mathbf{F}E(a)\}$; $\mathbf{t}_4 = \{\mathbf{T}D(a), \mathbf{T}E(a), \mathbf{F}D(a), \mathbf{F}E(a)\}$.
Since the minimal completion forest $\mathcal{F} = \{\mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4\}$ are closed, $\mathcal{F}$ is minimally closed. Thus $\{C \sqcup D(a), \neg C \sqcup E(a)\} \vdash^m_{LP} D \sqcup E(a)$. Therefore, $\{C \sqcup D(a), \neg C \sqcup E(a)\} \vdash^m_{LP} D \sqcup E(a)$ by Theorem 3.

5 Conclusion and Future Work

In this paper, we have presented a nonmonotonic and paraconsistent semantics, called minimally paradoxical semantics, for $\mathcal{ALC}$, which can be seen a naturally extension of the classical semantics. The suitability of our semantics is justified by several important properties. In particular, the new semantics overcomes some shortcomings of existing paraconsistent DLs and nonmonotonic DLs. Based on the signed tableau, we have developed a sound and complete algorithm, named minimal signed tableau, to implement paraconsistent and nonmonotonic reasoning with DL ABoxes. This is achieved by introducing a preference relation on trees of completion forests in signed tableau. This new approach can be used in developing new tableau algorithms for other nonmonotonic DLs. There are several issues for future work: First, we plan to implement the minimal signed tableau for DLs. Before this is done, we will first develop some heuristics for efficient implementation; Second, we will explore applications of the new paraconsistent semantics in ontology repair, revision and merging.

Acknowledgments

The authors appreciate the referees for their helpful and constructive comments. This work was supported by NSFC under 60973003, 60496322 and ARC under DP1093652.

References

1. 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, Cambridge (2003)

2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
3. Bertossi, L.E., Hunter, A., Schaub, T. (eds.): Inconsistency Tolerance. LNCS, vol. 3300. Springer, Heidelberg (2005)
4. Schaffert, S., Bry, F., Besnard, P., Decker, H., Decker, S., Enguix, C.F., Herzig, A.: Paraconsistent reasoning for the semantic web. In: Proc. of ISWC-URSW 2005, Ireland, pp. 104–105 (2005)
5. Huang, Z., van Harmelen, F., ter Teije, A.: Reasoning with inconsistent ontologies. In: Proc. of IJCAI 2005, UK, pp. 454–459. Professional Book Center (2005)
6. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 353–367. Springer, Heidelberg (2005)
7. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Proc. of IJCAI 2009, USA, pp. 891–897 (2009)
8. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proc. of IJCAI 2003, Mexico, pp. 355–362. Morgan Kaufmann, San Francisco (2003)
9. Ma, Y., Hitzler, P., Lin, Z.: Algorithms for paraconsistent reasoning with OWL. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 399–413. Springer, Heidelberg (2007)
10. Zhang, X., Xiao, G., Lin, Z.: A tableau algorithm for handling inconsistency in OWL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 399–413. Springer, Heidelberg (2009)
11. Zhang, X., Zhang, Z., Xu, D., Lin, Z.: Argumentation-based reasoning with inconsistent knowledge bases. In: Farzindar, A., Kešelj, V. (eds.) Canadian AI 2010. LNCS, vol. 6085, pp. 87–99. Springer, Heidelberg (2010)
12. Zhang, X., Lin, Z., Wang, K.: Towards a paradoxical description logic for the Semantic Web. In: Link, S., Prade, H. (eds.) FoIKS 2010. LNCS, vol. 5956, pp. 306–325. Springer, Heidelberg (2010)
13. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. J. Autom. Reasoning 14(1), 149–180 (1995)
14. Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W., Schaerf, A.: An epistemic operator for description logics. Artif. Intell. 100(1-2), 225–274 (1998)
15. Donini, F.M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Trans. Comput. Log. 3(2), 177–225 (2002)
16. Bonatti, P.A., Lutz, C., Wolter, F.: Description logics with circumscription. In: Proc. of KR 2006, UK, pp. 400–410. AAAI Press, Menlo Park (2006)
17. Yue, A., Ma, Y., Lin, Z.: Four-valued semantics for default logic. In: Lamontagne, L., Marchand, M. (eds.) Canadian AI 2006. LNCS (LNAI), vol. 4013, pp. 195–205. Springer, Heidelberg (2006)
18. Lin, Z., Ma, Y., Lin, Z.: A fault-tolerant default logic. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 253–265. Springer, Heidelberg (2006)
19. Priest, G.: Minimally inconsistent LP. Studia Logica: An Inter. J. for Symbolic Log. 50(2), 321–331 (1991)
20. Ginsberg, M.L. (ed.): Readings in nonmonotonic reasoning. Morgan Kaufmann Publishers Inc., San Francisco (1987)