A Distance-based Framework for Inconsistency-Tolerant Reasoning and Inconsistency Measurement in DL-Lite

Xiaowang Zhang\textsuperscript{a,b,c}, Kewen Wang\textsuperscript{d}, Zhe Wang\textsuperscript{d}, Yue Ma\textsuperscript{e}, Guilin Qi\textsuperscript{f}, Zhiyong Feng\textsuperscript{a,b}

\textsuperscript{a}School of Computer Science and Technology, Tianjin University, Tianjin, China
\textsuperscript{b}Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
\textsuperscript{c}Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China
\textsuperscript{d}School of Information and Communication Technology, Griffith University, BNE, Australia
\textsuperscript{e}Laboratoire de Recherche en Informatique, University Paris Sud, Paris, France
\textsuperscript{f}School of Computer Science and Engineering, Southeast University, Nanjing, China

Abstract

In this paper, we present a distance-based framework for DL-Lite based on the notion of features. Within this framework, we propose a distance-based paraconsistent semantics for DL-Lite where meaningful conclusions can be rationally drawn even from an inconsistent knowledge base and we develop a distance-based inconsistency measurement for DL-Lite to provide more informative metrics which can tell the differences between axioms causing inconsistency and among inconsistent knowledge. Furthermore, we investigate several important logical properties (e.g., consistency preservation, closure consistency, splitting property etc.) of the entailment relation based on the new semantics and show its advantages in non-monotonic reasoning for DL-Lite. Finally, we show that our two distance-based inconsistency measures are basic inconsistency measures where some good properties hold such as Free Axiom Independence and Dominance of inconsistency etc.

Keywords: Semantic Web, description logics, DL-Lite, inconsistency handling, inconsistency-tolerant reasoning, inconsistency measurement, distance-based semantics
1. Introduction

Inconsistency handling is an important issue in ontology (knowledge base, KB) management communities since inconsistency is not rare in ontology applications and may be due to several reasons, such as errors in modeling, migration from other formalisms, ontology merging, and ontology evolution. However, as a logical foundation of Web Ontology Language (OWL), description logic (DL) reasoning mechanism based on two-valued semantics faces problem when inconsistency occurs since DL is a fragment of predicate logic, which is referred to as the triviality problem. That is, any conclusions, that are possibly irrelevant or even contradicting, will be entailed from an inconsistent DL ontology under the classical semantics.

In many practical ontology applications, there is a strong need for inferring (only) useful information from inconsistent ontologies. For instance, consider a simple DL KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{T} = \{\text{Penguin} \sqsubseteq \text{Bird}, \text{Swallow} \sqsubseteq \text{Bird}, \text{Bird} \sqsubseteq \text{Fly}, \text{Fly} \sqsubseteq \exists \text{hasWing}\}$ and $\mathcal{A} = \{\text{Penguin(tweety)}, \neg\text{Fly(tweety)}, \text{Swallow(fred)}\}$. The KB says that penguins are birds; swallows are birds; birds can fly; flying animals have wings; tweety is a penguin; tweety cannot fly; and fred is a swallow. Under the classical semantics for DLs, anything can be inferred from $\mathcal{K}$ since $\mathcal{K}$ is inconsistent (i.e., it has no model.). Intuitively, one might wish to still infer $\text{Bird(fred)}$ and $\text{Fly(fred)}$, while it is useless to derive both $\text{Fly(tweety)}$ and $\neg\text{Fly(tweety)}$ from $\mathcal{K}$.

There exist several proposals for inconsistency-tolerant reasoning DL KBs in the literature. These approaches usually fall into one of two different streams. The first one is based on the assumption that inconsistencies are caused by erroneous data and thus, they should be removed in order to obtain a consistent KB. In most approaches in this stream, the task of repairing inconsistent ontologies is actually reduced to finding a maximal consistent subset of the original KB. A shortcoming of these approaches is similar to the so-called multi-extension problem in Reiter’s default logic. That is, in many cases, an inconsistent KB may have several different maximal subsets that are consistent. The other stream, based on the idea of living with inconsistency, is to introduce a form of paraconsistent reasoning or inconsistency-tolerant reasoning by employing non-standard reasoning methods (e.g., non-standard inference and non-classical semantics). There are some strategies to select consistent subsets from an inconsistent KB as substitutes of the original KB in reasoning. The Belnap’s four-valued semantics has been successfully extended into DL where two additional logical values besides “true” and “false” are introduced to indicate contradictory conclusions. Inference power of the four-valued semantic...
tics is further enhanced by a new quasi-classical semantics for DLs proposed by Zhang et al. \cite{58}, which is a generalization of Hunter’s quasi-classical semantics for propositional logic. However, the reasoning capability of such paraconsistent methods is not strong enough for many practical applications. For instance, a conclusion, that can be derived from a consistent KB under the classical semantics, may become not derivable under their paraconsistent semantics. One limitation of existing approaches in the two approaches is mostly coarse-grained in the sense that they fail to fully utilize semantic information in the given inconsistent KB. For instance, when two interpretations make a concept unsatisfiable, one interpretation may be more reasonable than the other. But those existing approaches to paraconsistent semantics in DLs do not take this into account usually. Recently, there are some works in considering the priorities among different interpretations by introducing some preferred repair semantics \cite{15,8,7,16}. However, those approaches are based on the assumption that TBoxes are consistent.

As an important approach to handling inconsistency, inconsistency measurement is providing some measures for the inconsistency of a KB so that we can compare different KBs and evaluate their quality of information and then choose one that is least inconsistent \cite{18}. There exist many works in measuring inconsistency of KBs by applying minimal inconsistent sets \cite{26}, shapley inconsistency values \cite{23}, partial Max-SAT solvers \cite{53}, four-valued semantics \cite{35}, MUS decomposition \cite{24}, closed set packing \cite{25} and measuring inconsistency for prioritized KBs \cite{41}, stratified KBs \cite{40}, arbitrary KBs \cite{36}, probabilistic query answering \cite{54}, DL-Lite ontologies \cite{59} by three-valued semantics. Most of those approaches to measuring inconsistency of KBs in DLs are either syntax-based (where the results are sensitive to the syntactical structure of KBs) \cite{26,23,53,24,25} or multi-valued semantics (where the results are not so intuitive in characterizing facts in a classical logic) \cite{35,59}.

A distance-based semantics presented by Arieli \cite{1} has been proposed to deal with inconsistent KBs in propositional logic, which is inspired from distance-based merging procedures in propositional logic \cite{29,30}, where those interpretations with minimal distances defined between interpretations are chosen as models so that those models could closely characterize the semantics. A distance-based inconsistency-measurement has been developed for DLs, in particular, for DL-Lite \cite{34}. However, because the finiteness problem from the approach of \cite{34} is dealt by considering only interpretations with finite domains, it is interesting to observe models with infinite domains since a propositional KB has a finite number of finite models) while, in DLs, a KB might have infinite number
of models and a model might also be infinite.

To overcome these difficulties in the finiteness problem of DL, in this paper we first use the notion of features [49] and introduce a distance-based semantics for paraconsistent reasoning with DL-Lite. Features in DL-Lite are Herbrand interpretations extended with limited structure, which provide a novel semantic characterization for DLs. In addition, features also generalize the notion of types for TBoxes [31] to general KBs. Each KB in DL-Lite has a finite number of features and each feature is finite. This makes it possible to cast Arieli’s distance-based semantics to DL.

In this paper, we present a distance-based framework for both inconsistency-tolerant reasoning and inconsistency measurement in DL-Lite ontologies. The DL-Lite [11] is a family of lightweight description logics (DLs), which form the logical foundation of OWL 2 QL, one of the three profiles of OWL 2 for Web ontology language recommended by W3C [12]. Following [50], we choose DL-Lite$_{bool}^N$ [3], one of the most expressive members of the DL-Lite family, and define distance-based semantics for DL-Lite$_{bool}^N$ in a way analogous to the model-based approaches in propositional logic. DL-Lite$_{bool}^N$ [4], expressive enough to allow all boolean operators, generalizes the main DL-Lite dialects such as DL-Lite$_{core}$ and DL-Lite$_F$. Though our paper mainly discuss DL-Lite$_{bool}^N$, our proposed approach can be conveniently adapted to other DL-Lite dialects like DL-Lite$_R$ since types, a foundation of our distance-based semantics, can be adapted to them [50].

The main innovations and contributions of this paper can be summarized as follows.

- We introduce distance functions on types of DL-Lite$_{bool}^N$ KBs, which avoids the problem of domain infiniteness and model infiniteness in defining the distance function in terms of models of KBs. We choose DL-Lite$_{bool}^N$ [3], one of the most expressive members of the DL-Lite family, and define distance-based semantics for DL-Lite$_{bool}^N$ in a way analogous to the model-based approaches in propositional logic.

- We develop a way of measuring types that are closest to a TBox, based on the new distance function on types, and the notion of minimal model types is introduced. This notion is also extended to minimal model features for KBs. We propose a distance-based semantics for DL-Lite$_{bool}^N$ so that useful information can still be inferred when a KB is inconsistent. Our results show that the distance-based entailment is paraconsistent, non-monotonic, cautious as the paraconsistent based on multi-valued semantics. We also
show that the distance-based entailment is not over-skeptical in the sense that for a classically consistent KB, the distance-based entailment coincides with the classical entailment, which is missing in most existing para-consistent semantics for DLs.

- We present two distance-based measures for DL-Lite TBoxes and KBs respectively by introducing minimal inconsistency value. We show that our distance-based measurement can provide more informative metrics which can tell the differences between axioms causing inconsistency and between inconsistent KBs.

This paper is a significant extension of our previous conference paper [57] with a new section on distance-based inconsistency measurement, complete and detailed proofs for the results, a new section on experiments, and a new section on related works. In [55], we develop a forgetting-based approach to characterizing some distance-based reasoning and distance-based repair in DL-Lite.

The rest of this paper is organized as follows: The following section reviews briefly the syntax and classical semantics of DL-LiteN bool, some notions of types and features. Section 3 and Section 4 introduce the distance-based semantics for TBoxes and KBs respectively. Section 5 further discusses many important logical properties of the distance-based semantics. Section 6 presents the distance-based inconsistency measurement for DL-Lite. In the last two sections discuss related works and summarizes our work.

2. DL-LiteN bool and features

In this section, we briefly introduce the syntax and semantics of DL-LiteN bool, an important expressive member in the DL-Lite family and notions of types and features. For more comprehensive background knowledge of DL-Lite, we refer the reader to some basic references such as the DL-Lite family [11, 4] and features [31].

2.1. Syntax and semantics of DL-LiteN bool

A signature is a finite set Σ = Σ_A ∪ Σ_R ∪ Σ_I ∪ Σ_N where Σ_A is the set of concept names, Σ_R the set of role names, Σ_I the set of individual names (or, objects) and Σ_N the set of natural numbers in Σ. We use A (with subscripts A_1, A_2) to denote a concept name, P (with subscripts P_1, P_2) to denote a role name, lowercase letters a, b, c to denote individual names and assume 1 is always in Σ_N. Note that ⊤ and ⊥ will not be considered as concept names or role names since they can be expressed by other concept names via operators.
Formally, given a signature $\Sigma$, the DL-Lite$_{\text{bool}}$ language is inductively constructed by syntax rules:

- **r1**: $R \leftarrow P \mid P^-$;
- **r2**: $B \leftarrow \top \mid A \mid \geq n R$;
- **r3**: $C \leftarrow B \mid \neg C \mid C_1 \sqcap C_2$.

We say $B$ a basic concept and $C$ a general concept. Other standard concept constructs such as $\bot$, $\exists R$, $\leq n R$ and $C_1 \sqcup C_2$ can be introduced as abbreviations: $\bot$ for $\neg \top$, $\exists R$ for $\geq 1 R$, $\leq n R$ for $\neg (\geq (n + 1 R))$ and $C_1 \sqcup C_2$ for $\neg (\neg C_1 \cap \neg C_2)$. For any $P \in \Sigma_R$, $P^- = P$.

A TBox $T$ is a finite set of (concept) inclusions of the form $C_1 \sqsubseteq C_2$ where $C_1$ and $C_2$ are general concepts. An ABox $A$ is a finite set of concept assertions $C(a)$ and role assertions $R(a, b)$\footnote{It is slight different from the definition of ABoxes [11] where concept names, the negation of concept names, and role names are allowed. To present our technique clearly, we generalize general concepts, role names and its converse similar to the treatment in [49].} Concept inclusions, concept assertions and role assertions are axioms. A KB is composed of a TBox and an ABox, written by $K = (T, A)$. Let $\text{Sig}(K)$ denote the signature of $K$, that is, a collection of all concept names, role names, individual names, and natural numbers occurring in $K$. We say $K$ a KB over $\Sigma$ if $\text{Sig}(K) \subseteq \Sigma$ and $\varphi$ an axiom over $\Sigma$ if $\text{Sig}(\varphi) \subseteq \Sigma$ respectively.

An interpretation $\mathcal{I}$ is a pair $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set called the domain and $\cdot^\mathcal{I}$ is an interpretation function such that $a^\mathcal{I} \in \Delta^\mathcal{I}$, $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. General concepts are interpreted as follows:

- $(P^-)^\mathcal{I} = \{(a^\mathcal{I}, b^\mathcal{I}) \mid (b^\mathcal{I}, a^\mathcal{I}) \in P^\mathcal{I}\}$;
- $(\geq n R)^\mathcal{I} = \{a^\mathcal{I} \mid \sharp\{b^\mathcal{I} : (a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}\} \geq n\}$;
- $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} - C^\mathcal{I}$;
- $(C_1 \sqcap C_2)^\mathcal{I} = C_1^\mathcal{I} \cap C_2^\mathcal{I}$.

The definition of interpretation is based on the unique name assumption (UNA), i.e., $a^\mathcal{I} \neq b^\mathcal{I}$ for two different individual names $a$ and $b$.

An interpretation $\mathcal{I}$ is a model of a concept inclusion $C_1 \sqsubseteq C_2$ (a concept assertion $C(a)$, or a role assertion $R(a, b)$) if $C_1^\mathcal{I} \subseteq C_2^\mathcal{I}$ ($a^\mathcal{I} \in C^\mathcal{I}$, or $(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$); and $\mathcal{I}$ is called a model of a TBox $T$ (an ABox $A$) if $\mathcal{I}$ is a model of each
inclusion of $\mathcal{T}$ (each assertion of $\mathcal{A}$). $\mathcal{I}$ is called a model of a KB $(\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model of both $\mathcal{T}$ and $\mathcal{A}$. We use $\text{Mod}(\mathcal{K})$ to denote the set of models of $\mathcal{K}$. A KB $\mathcal{K}$ entails an axiom $\phi$, if $\text{Mod}(\mathcal{K}) \subseteq \text{Mod}(\{\phi\})$. Two KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are equivalent if $\text{Mod}(\mathcal{K}_1) = \text{Mod}(\mathcal{K}_2)$, denoted by $\mathcal{K}_1 \equiv \mathcal{K}_2$. A KB $\mathcal{K}$ is consistent if it has at least one model; and, inconsistent otherwise. An ABox $\mathcal{A}$ is consistent with respect to (w.r.t.) a TBox $\mathcal{T}$ if there exists some model $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{I}$ is also a model of $\mathcal{A}$. Let $\mathcal{T}$ be a TBox. A concept or a role $E$ is satisfiable w.r.t. $\mathcal{T}$ if there exists some model $I$ such that $E^I \neq \emptyset$; and unsatisfiable w.r.t. $\mathcal{T}$ otherwise. A TBox $\mathcal{T}$ is coherent if there exists no atomic concept or atomic role which is unsatisfiable w.r.t. $\mathcal{T}$; and incoherent otherwise.

2.2. Types and features

As we well known, models of a KB are often infinite and the number of models of a KB is also possibly infinite. To characterize infinite models in a finite way, two important notions, namely, type and feature, are respectively defined by Kontchakov et al. [31] and Wang et al. [49, 50] for TBoxes and general KBs in DL-Lite respectively.

**Definition 1 (Types).** Let $\Sigma$ be a signature. A $\Sigma$-type (or simply a type) is a set of basic concepts over $\Sigma$ if the followings hold:

- $\top \in \tau$; and
- for any $m, n \in \Sigma_N$ with $m < n$, $R \in \Sigma_R \cup \{P^- | P \in \Sigma_R\}$, $\geq n R \in \tau$ implies $\geq m R \in \tau$.

As $\top \in \tau$ for any type $\tau$, we omit it in examples for simplicity. The empty type, simply denoted by $\{\}$, is a type without any element. Let $T_\Sigma$ denote the set of all $\Sigma$-types. Note that if $\exists P$ (or $\exists P^-$) occurs in a general concept $C$ then $\exists P^-$ (or $\exists P$) should be also considered as a new concept independent of $\exists P$ (or $\exists P^-$) in computing types of $C$ respectively. We define a type set as a set of types $\{\tau_1, \ldots, \tau_m\}$, denoted as $\Xi$ and a type group as a set of type sets $\{\Xi_1, \ldots, \Xi_n\}$, denoted as $\Pi$. Then we denote $\bigcup \Xi = \tau_1 \cup \cdots \cup \tau_m$ and $\bigcap \Pi = \Xi_1 \cap \cdots \cap \Xi_n$.

Note that types coincide with models of propositional formulas where $T_\Sigma(C)$ coincides with the set of models satisfying concept $C$.

A type $\tau$ satisfies a basic concept $B$ if $B \in \tau$, $\tau$ satisfies $\neg C$ if $\tau$ does not satisfy $C$, and $\tau$ satisfies $C_1 \cap C_2$ if $\tau$ satisfies both $C_1$ and $C_2$. Let $T_\Sigma(C)$ denote a collection of all $\Sigma$-types of $C$. In this way, each general concept $C$ over $\Sigma$ corresponds to a set $T_\Sigma(C)$ of all $\Sigma$-types satisfying $C$. A type $\tau$ satisfies a concept inclusion $C \subseteq D$ if $\tau \in T_\Sigma(\neg C \cup D)$.
Example 1. Given a signature $\Sigma = \{A, B, P, D\}$, a concept $C = A \cap (B \cup \exists P)$ can be equivalently transformed into the unique disjunction normal form (DNF) [50] $(A \cap B \cap \exists P) \cup (A \cap \neg B \cap \exists P) \cup (A \cap B \cap \neg \exists P)$. Thus $T_\Sigma(C) = \{\tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}, \tau_{21}, \tau_{22}, \tau_{23}, \tau_{24}, \tau_{31}, \tau_{32}, \tau_{33}, \tau_{34}\}$, where

$$
\begin{align*}
\tau_{11} &= \{A, B, \exists P, \exists P^-, D\}, & \tau_{12} &= \{A, B, \exists P, D\}, \\
\tau_{13} &= \{A, B, \exists P, \exists P^\sim\}, & \tau_{14} &= \{A, B, \exists P\}, \\
\tau_{21} &= \{A, \exists P, \exists P^-, D\}, & \tau_{22} &= \{A, \exists P, D\}, \\
\tau_{23} &= \{A, \exists P, \exists P^\sim\}, & \tau_{24} &= \{A, \exists P\}, \\
\tau_{31} &= \{A, B, \exists P^-, D\}, & \tau_{32} &= \{A, B, D\}, \\
\tau_{33} &= \{A, B, \exists P^\sim\}, & \tau_{34} &= \{A, B\}.
\end{align*}
$$

To define the satisfaction of a type to a TBox $\mathcal{T}$, certain non-propositional inference needs to be taken into account [60]. Formally, a type $\tau$ is a model type of a TBox $\mathcal{T}$ if and only if (1) it satisfies each inclusion in $\mathcal{T}$, and (2) $\exists R \in \tau$ only if $\exists R$ is satisfiable w.r.t. $\mathcal{T}$ (under classical DL semantics) for each role or its inverse $R$. Model type sets and model type groups are analogously defined. It can be shown that if $\Xi$ is a model type set of a TBox $\mathcal{T}$ then $\exists P \in \bigcup \Xi$ if and only if $\exists P^- \in \bigcup \Xi$ [60]. This property is called role coherence which can be used to check whether a type set is the model type set of some TBox. Let $\Pi_\Sigma(\mathcal{T})$ denote the model type group $\{T_\Sigma(\neg C_1 \sqcup D_1), \ldots, T_\Sigma(\neg C_n \sqcup D_n)\}$ of $\mathcal{T}$ where $\mathcal{T} = \{C_1 \subseteq D_1, \ldots, C_n \subseteq D_n\}$ is a TBox over $\Sigma$. It is not hard to see that when $\mathcal{T}$ is a coherent TBox, $\bigcap (\Pi_\Sigma(\mathcal{T}))$ is the collection of model $\Sigma$-types of $\mathcal{T}$.

Intuitively, a basic concept $\exists P$ is satisfiable w.r.t. $\mathcal{T}$ if and only if $\exists P^-$ is satisfiable w.r.t. $\mathcal{T}$ for any $P \in \Sigma_R$ since the satisfiability of $\exists P$ infers $P(a, b)$ for some individuals $a$ and $b$, which also ensures the satisfiability of $\exists P$

Definition 2 (Herbrand set). Let $\Sigma$ be a signature. A $\Sigma$-Herbrand set (or simply Herbrand set) $\mathcal{H}$ is a finite set of member assertions satisfying:

- for each $a \in \Sigma_I$, if $B_1(a), \ldots, B_k(a)$, where $\{B_1, \ldots, B_k\} \subseteq \Sigma_A$ are all the concept assertions about $a$ in $\mathcal{H}$, then the set $\{B_1, \ldots, B_k\}$ is a $\Sigma$-type;

- for each $P \in \Sigma_R$, if $P(a, b_1)(1 \leq i \leq n)$ are all the role assertions about $a$ in $\mathcal{H}$, then for any $m \in \Sigma_N$ with $m \leq n$, $(\geq m P)(a)$ is in $\mathcal{H}$; and if $P(b_1, a)(1 \leq i \leq n)$ are all the role assertions in $\mathcal{H}$, then for any $m \in \Sigma_N$ with $m \leq n$, $(\geq m P^-)(a)$ is in $\mathcal{H}$.

Note that Herbrand sets are different from ABoxes by adopting a closed world assumption (CWA). For instance, assertion $\neg B(a)$
holds in $\mathcal{H}$ and $P(a, b) \notin \mathcal{H}$ means that there does no relationship w.r.t. $P$ from individual $a$ to individual $b$.

We simply write $\tau(a) = \{B_1(a), \ldots, B_k(a)\}$ where $\tau = \{B_1, \ldots, B_k\}$. Moreover, given a set of types $\Xi = \{\tau_1, \ldots, \tau_m\}$, $\Xi(a)$ denotes $\{\tau_1(a), \ldots, \tau_m(a)\}$ without confusion. In this case, we say $\tau(a)$ is in $\mathcal{H}$ if $\{B_1(a), \ldots, B_k(a)\} \subseteq \mathcal{H}$.

A Herbrand set $\mathcal{H}$ satisfies a concept assertion $C(a)$ (a role assertion $P(a, b)$ or $P^-(b, a)$) if $\tau(a)$ is in $\mathcal{H}$ and $\tau \in T_\Sigma(C)$ ($P(a, b) \in \mathcal{H}$ or $P^-(b, a) \in \mathcal{H}$). A Herbrand set $\mathcal{H}$ satisfies an ABox $\mathcal{A}$ if $\mathcal{H}$ satisfies all assertions in $\mathcal{A}$.

**Definition 3 (Feature).** Let $\Sigma$ be a signature. A $\Sigma$-feature (or simply a feature) $\mathcal{F}$ is a pair $\langle \Xi, \mathcal{H} \rangle$, where $\Xi$ is a non-empty set of $\Sigma$-types and $\mathcal{H}$ a $\Sigma$-Herbrand set, if $\mathcal{F}$ satisfies:

- for each $P \in \Sigma_R$, $\exists P \in \bigcup \Xi$ if and only if $\exists P^- \in \bigcup \Xi$; and
- for each $a \in \Sigma_I$ and $\tau(a)$ in $\mathcal{H}$, $\tau \in \Xi$.

Intuitively speaking, in a feature $\langle \Xi, \mathcal{H} \rangle$, $\Xi$ consists of the types which are induced by the elements in the domain of interest, and $\mathcal{H}$ specifies the membership of all named individuals [50]. A feature $\mathcal{F}$ satisfies an inclusion $C_1 \sqsubseteq C_2$ over $\Sigma$, if $\Xi \subseteq T_\Sigma(\neg C_1 \sqcup C_2)$; $\mathcal{F}$ satisfies a concept assertion $C(a)$ over $\Sigma$, if $\tau(a) \in \mathcal{H}$ and $\tau \in T_\Sigma(C)$; and $\mathcal{F}$ satisfies a role assertion $P(a, b)$ (resp., $P^-(b, a)$) over $\Sigma$, if $P(a, b) \in \mathcal{H}$. A feature $\mathcal{F}$ is a model feature of KB $\mathcal{K}$ if $\mathcal{F}$ satisfies each inclusion and each assertion in $\mathcal{K}$. $\text{Mod}^F(\mathcal{K})$ denotes the set of all model features of $\mathcal{K}$.

It is shown that $\mathcal{K}$ is consistent if and only if $\text{Mod}^F(\mathcal{K}) \neq \emptyset$ [49]. Let $\Sigma$ be a signature. Given two KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ over $\Sigma$, $\mathcal{K}_1 F$-entails $\mathcal{K}_2$ if $\text{Mod}^F(\mathcal{K}_1) \subseteq \text{Mod}^F(\mathcal{K}_2)$, written by $\mathcal{K}_1 \models^F \mathcal{K}_2$; and $\mathcal{K}_1$ is $F$-equivalent $\mathcal{K}_2$ if $\text{Mod}^F(\mathcal{K}_1) = \text{Mod}^F(\mathcal{K}_2)$, written by $\mathcal{K}_1 \equiv^F \mathcal{K}_2$.

**Lemma 1.** [57] Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be two KBs. The followings hold.

- $\mathcal{K}_1 \models \mathcal{K}_2$ if and only if $\mathcal{K}_1 \models^F \mathcal{K}_2$;
- $\mathcal{K}_1 \equiv \mathcal{K}_2$ if and only if $\mathcal{K}_1 \equiv^F \mathcal{K}_2$.

As a result, F-entailment characterizes major DL reasoning tasks such as consistency, axiom entailment and KB equivalence. In the remainder of this paper, we just use $\models$ for $\models^F$ when there is no confusion.

Moreover, a finite signature can characterize model types and model features of a KB since each KB contains a finite number of concept names, role names, natural numbers, and, individual names. In this paper, we set that all signatures are finite.
Example 2. Consider a KB $K = (T, A)$, where $T = \{ A \sqsubseteq \exists P, B \sqsubseteq \exists P, \exists P^- \sqsubseteq B, (A \cap B) \sqsubseteq \bot, \geq 2 P^- \sqsubseteq \bot \}$ and $A = \{ A(a), P(a, b) \}$. It is shown that $K$ has no finite model \textbf{[77]}. An infinite model $I$ of $K$ is defined as follows: $\Delta^I = \{d_1, d_2, d_3, d_4, \ldots \}$, $a^I = d_a$ and $b^I = d_b$; the concept $A$ is interpreted as a singleton $\{ k \}$ and $B$ as $\{ d_b, d_1, d_2, d_3 \}$; and role $P$ is interpreted as $\{(d_a, d_b), (d_b, d_1), (d_1, d_2), \ldots, (d_2, d_{i+1}) \}$. Let $\Sigma_K = \{ A, B, P, 1, 2, a, b \}$. The (finite) model feature of $K$ that corresponds to $I$ is $F = \langle \Xi, H \rangle$ where the Herbrand set $H = \{ \tau_1(a), \tau_2(b), P(a, b) \}$ and $\Xi = \{ \tau_1, \tau_2 \}$ with $\tau_1 = \{ A, \neg B, \exists P, \neg \exists P^- \}$ and $\tau_2 = \{ \neg A, B, \exists P, \exists P^- \}$. Moreover, $K \models^F B(b)$ and $K \models^F \exists P^- (b)$ since $\tau_2(b) \in H$. We can conclude that $T \models^F A \sqsubseteq \exists P$ since $\{ \tau_1, \tau_2 \} \subseteq T_\Sigma (\neg A \cup \exists P)$ where $T_\Sigma (\neg A \cup \exists P)$ is a set of types as follows: $\{ B, \exists P, \exists P^- \}, \{ A, B, \exists P, \exists P^- \}, \{ B, \exists P \}, \{ B, \exists P^- \}, \{ A, B, \exists P \}, \{ A, \exists P, \exists P^- \}, \{ B, \exists P^- \}, \{ \exists P^- \}, \{ \exists P, \exists P^- \}, \{ \exists P \}$, and $\{ \}$.

Finally, we introduce two special axioms: tautology and contradiction. An axiom is a tautology if all features are its model features. A tautology is of the forms: $\bot \subseteq C, C \subseteq \top, (D \cap \neg D) \subseteq C, C \subseteq (D \cup \neg D) \cup \bot (a), (C \cup \neg C)(a)$ etc. and a contradiction is of the forms: $\top \subseteq \bot, (C \cup \neg C) \subseteq (D \cap \neg D), \bot (a), (C \cap \neg C)(a)$ etc. Accordingly, an axiom is a contradiction if and only if there exists no model feature of it.

3. Distance-based semantics for TBoxes

In this section, we introduce distance functions between types of TBoxes and then define the distance-based semantics for TBoxes.

To measure the closeness of two types, we first define a distance function between two types in terms of the symmetric difference for sets.

**Definition 4.** Let $\Sigma$ be a signature. A total function $d : T_\Sigma \times T_\Sigma \rightarrow \mathbb{R}^+ \cup \{0\}$ is a pseudo-distance function (for short, distance function) on $T_\Sigma$ if it satisfies:

- $\forall \tau_1, \tau_2 \in T_\Sigma, d(\tau_1, \tau_2) = 0$ if and only if $\tau_1 = \tau_2$; and

- $\forall \tau_1, \tau_2 \in T_\Sigma, d(\tau_1, \tau_2) = d(\tau_2, \tau_1)$.

Given a type $\tau \in T_\Sigma$ and a type set $\Xi \subseteq T_\Sigma$, the distance function between $\tau$ and $\Xi$ is defined as $d(\tau, \Xi) = \min\{d(\tau, \tau') \mid \tau' \in \Xi\}$.

If $\Xi = \emptyset$, then we set $d(\tau, \Xi) = d$ where $d$ is a default value of distance function greater than any value be to considered. This setting is used to exclude all contradictions (e.g., $\top \subseteq \bot$) under our candidate semantics since a contradiction can bring less useful information.
Note that our distance functions define among types over the same signature. In other words, by default, we set that two types are incomparable if they are defined over different signatures.

There are two representative distance functions on types, namely, Hamming distance function where \( d^H(\tau_1, \tau_2) = |(\tau_1 - \tau_2) \cup (\tau_2 - \tau_1)| \) and drastic distance function where \( d^D(\tau_1, \tau_2) = 0 \) if \( \tau_1 = \tau_2 \) and \( d^D(\tau_1, \tau_2) = 1 \) otherwise.

Firstly, we introduce aggregation functions to aggregate all distance functions between a type and each set of types in a type group. An aggregation function \( f \) is a total function that accepts a multi-set of real numbers and returns a real number, satisfying:

- \( f \) is non-decreasing in the values of its argument;
- \( f(\{x_1, \ldots, x_n\}) = 0 \) if and only if \( x_1 = \ldots = x_n = 0 \); and
- \( \forall x \in \mathbb{R}^+ \cup \{0\}, f(\{x\}) = x \).

There exist some popular aggregation functions [34]:

- The summation function: \( f^s(x_1, \ldots, x_n) = \sum_{1 \leq i \leq n} x_i \);
- The maximum function: \( f^m(x_1, \ldots, x_n) = \max_{1 \leq i \leq n} x_i \);
- The \( \kappa \)-voting function \((0 < \kappa < 1)\):
  \[
  f^\kappa(x_1, \ldots, x_n) = \begin{cases} 
  0, & \text{if } \text{Zero}(\{x_1, \ldots, x_n\}) = n; \\
  \frac{1}{\kappa}, & \text{if } \lceil \kappa \cdot n \rceil \leq \text{Zero}(\{x_1, \ldots, x_n\}) < n; \\
  1, & \text{otherwise.}
  \end{cases}
  \]
  where \( \text{Zero}(\{x_1, \ldots, x_n\}) \) is the number of zeros in \( \{x_1, \ldots, x_n\} \). \( \kappa \) is called the voting index of \( f^\kappa \).

Using aggregation functions, the distance function between two types can be extended to a type and a type group.

**Definition 5.** Let \( \Sigma \) be a signature, \( \tau \) a type and \( \Pi = \{\Xi_1, \ldots, \Xi_n\} \) a type group. Given a distance function \( d \) and an aggregation function \( f \), \( \lambda_{d,f} \) between \( \tau \) and \( \Pi \) is defined as \( \lambda_{d,f}(\tau, \Pi) = f(\{d(\tau, \Xi_1), \ldots, d(\tau, \Xi_n)\}) \). Furthermore, a type \( \tau \) is called \((d,f)\)-minimal (for short, minimal) w.r.t. \( \Pi \) if for any type \( \tau' \in T_\Sigma \), \( \lambda_{d,f}(\tau, \Pi) \leq \lambda_{d,f}(\tau', \Pi) \).
Let $\Sigma$ be a signature, $\Xi$ a type set over $\Sigma$, $\Pi$ a type group over $\Sigma$, $d$ a distance function, and $f$ an aggregation function. We use $\Lambda_{d,f}(\Pi, \Xi)$ to denote a set of all $(d,f)$-minimal types w.r.t. $\Pi$ in $\Xi$.

First, minimal types have the following simple properties.

**Proposition 1.** Let $\Sigma$ be a finite signature and $\Pi = \{\Xi_1, \ldots, \Xi_n\}$ a type group over $\Sigma$. For any distance function $d$ and any aggregation function $f$, we have

1. $\Lambda_{d,f}(\Pi, T_\Sigma) \neq \emptyset$;
2. If $\bigcap \Pi \neq \emptyset$ then $\Lambda_{d,f}(\Pi, T_\Sigma) = \bigcap \Pi$.

The first statement guarantees that a minimal type of it always exists if a type group contains a non-empty type set and the second shows that each type belong to all members of a type group is exactly a minimal type.

Let $\Sigma$ be a signature and $T = \{\psi_1, \ldots, \psi_n\}$ a TBox over $\Sigma$. Each axiom $\psi_i$ is of the form $C_i \sqsubseteq D_i$ ($1 \leq i \leq n$) where $C_i$ and $D_i$ ($1 \leq i \leq n$) are concepts. We simply write $\Pi_{\Sigma}(T)$ as $\Pi(T)$ if $\Sigma = \text{Sig}(T)$.

Proposition 1 can be directly reformulated for TBoxes as follows.

**Corollary 1.** Let $\Sigma$ be a finite signature and $T$ a TBox over $\Sigma$. For any distance function $d$ and any aggregation function $f$, we have

1. $\Lambda_{d,f}(\Pi_{\Sigma}(T), T_\Sigma) \neq \emptyset$;
2. if $T$ is consistent then $\Lambda_{d,f}(\Pi_{\Sigma}(T), T_\Sigma) = \bigcap \Pi_{\Sigma}(T)$.

The above second item is no longer true if a TBox $T$ is not consistent.

**Example 3.** Let $T = \{\top \sqsubseteq A, A \sqsubseteq \exists P, \exists P \sqsubseteq \bot\}$ and $\Sigma = \text{Sig}(T)$. Assume that $d$ is the Hamming distance function and $f$ is the summation function. So $\Sigma = \{A, P\}$ and $T$ is inconsistent. Thus we can conclude that $T$ has eight possible types: $\tau_{11} = \{\}, \tau_{12} = \{\exists P^\bot\}, \tau_{21} = \{\exists P\}, \tau_{22} = \{\exists P, \exists P^\bot\}, \tau_{31} = \{A\}, \tau_{32} = \{A, \exists P^\bot\}, \tau_{41} = \{A, \exists P\},$ and $\tau_{42} = \{A, \exists P, \exists P^\bot\}$. Thus, $\bigcap \Pi(T)$ = $\emptyset$ while $\Lambda_{d,H,f_s}(\Pi(T), T_\Sigma) = \{\tau_{11}, \tau_{12}, \tau_{31}, \tau_{32}, \tau_{41}, \tau_{42}\}$.

Unfortunately, note that $\Lambda_{d,f}(\Pi(T), T_\Sigma)$ does not always satisfy the role coherence as the following example shows.

**Example 4.** Let $T = \{\top \sqsubseteq (A \sqcap \exists P), \exists P^\bot \sqsubseteq \bot\}$ and $\Sigma = \text{Sig}(T)$. Consider the Hamming distance function and the summation function, we can compute $\Lambda_{d_H,f_s}(\Pi(T), T_\Sigma) = \{\{A, \exists P\}\}$. Thus $\exists P^\bot \not\in \bigcup \Lambda_{d_H,f_s}(\Pi(T), T_\Sigma)$.

12
The reason that the role coherence might be absent in $\Lambda_{d,f}(\Pi(T), T_\Sigma)$ is that $\exists P$ and $\exists P^-$ are taken as two independent concepts so that the relation of satisfiability between $\exists P$ and $\exists P^-$ cannot be captured when minimal types are computed. In other words, given an arbitrary type set, there is not always a TBox such that it is a model type set of the TBox \[60].

Given an arbitrary type set $\Xi$, if it is not a model type set of any TBox, there are two possible options to recovery the role coherence: removing and adding. For instance, if $\tau \in \Xi$ such that $\exists R \in \tau$ and $\exists R^- \not\in \bigcup \Xi$ for some role $R$, then we can either remove $\tau$ from $\Xi$ or add a new type $\tau$ such that $\exists R^- \in \tau$ in to $\Xi$. In Example 4 if we remove the type $\{A, \exists P\}$, then $\Lambda_{d,f}(\Pi(T), T_\Sigma)$ will be empty, which is not desirable. In other words, the removing approach could cause the empty type set where the reasoning becomes trivial. So we will extend the type violating the role coherence.

For instance, consider Example 4 again, there are three possible types $\tau_1 = \{\exists P^-\}$, $\tau_2 = \{A, \exists P^-\}$, and $\tau_3 = \{A, \exists P, \exists P^-\}$ such that $\exists P^- \in \tau_i \ (i = 1, 2, 3)$ where $\lambda_{d,f}(\tau_1, \Pi(T)) = 3$, $\lambda_{d,f}(\tau_2, \Pi(T)) = 2$, and $\lambda_{d,f}(\tau_3, \Pi(T)) = 1$. So we can pick $\tau_3$ as the desired minimal type. Furthermore, this extension is an iterative process since newly added types possibly contains new role names and role incoherence is not yet satisfied at every step. To construct a model type set from a random type set $\Xi$, we introduce an iterative operator $\mu_{d,f}(\Xi)$ and its fixpoint.

Formally, let $\Sigma$ be a finite signature and $\Pi$ a type group over $\Sigma$. Given a type set $\Xi$ over $\Sigma$, let $\mu_{d,f}(\Xi) = \Xi \cup \Xi'$, where $\Xi' \subseteq T_\Sigma$ and $\Xi' = \{\tau\ | \ \exists R \in \bigcup \Xi$ and $\exists R^+ \not\in \bigcup \Xi, \exists R^- \in \tau$ and for any type $\tau' \in T_\Sigma, \exists R^- \in \tau'$ implies $\lambda_{d,f}(\tau, \Pi) \leq \lambda_{d,f}(\tau', \Pi)\}$. We use $\Xi^+$ to denote the fixpoint of $\mu_{d,f}$, i.e., $\Xi^+ = FP(\mu_{d,f})(\Xi)$. For any distance function $d$, any aggregation function $f$, and any type set $\Xi$, we can conclude that $\Xi^+$ always exists since $\mu_{d,f}$ is inflationary (i.e., $\Xi \subseteq \mu_{d,f}(\Xi)$) and $\Sigma$ is finite.

Given a signature $\Sigma$ and a TBox $T$ over $\Sigma$, $\Lambda_{d,f}^+(\Pi(T), T_\Sigma)$ is the minimal model type set of $T$. Intuitively, a minimal model type set is a set of minimal types with maintaining role coherence. In Example 4, $\Lambda_{d,f}^+(\Pi(T), T_\Sigma) = \Lambda_{d,f}(\Pi(T), T_\Sigma) \cup \{\tau_3\} = \{\{A, \exists P\}, \{A, \exists P, \exists P^+\}\}$. By definitions, given a TBox $T$, a distance function $d$, and an aggregation function $f$, we present a naive algorithm called $\text{MinModelType algorithm}$ to compute $(d,f)$-minimal model types of $T$ shown in Algorithm 1 (implemented in Section 7 later).

We show that minimal model type sets meet our motivation.

**Proposition 2.** Let $\Sigma$ be a signature and $T$ a TBox over $\Sigma$. For any distance function $d$ and aggregation function $f$, the followings hold:
1. $\Lambda_{d,f}^+(\Pi_{\Sigma}(T), T_\Sigma) \neq \emptyset$;

2. $\Lambda_{d,f}^+(\Pi_{\Sigma}(T), T_\Sigma) = \bigcap(\Pi_{\Sigma}(T))$, if $T$ is coherent;

3. $\exists P \in \bigcup(\Lambda_{d,f}^+(\Pi_{\Sigma}(T), T_\Sigma))$ if and only if $\exists P^- \in \bigcup(\Lambda_{d,f}^+(\Pi_{\Sigma}(T), T_\Sigma))$ for any role name $P \in \Sigma_R$.

In Proposition 2, the first item states that there always exist minimal model types for any non-empty TBox; the second shows that when a TBox is consistent, each minimal model type is exactly model type; and the third ensures that minimal model type sets always satisfy the role coherence.

Algorithm 1 The MinModelType algorithm

1: procedure MinModelType($T, d, f$) // $T = \{\phi_1, \ldots, \phi_m\}$
2: $\Sigma \leftarrow \text{Sig}(T)$
3: For any $\phi_i$ in $T$, $\Xi_i \leftarrow \text{Mod}^T(\{\phi_i\})$
4: $\Pi \leftarrow \{\Xi_1, \ldots, \Xi_m\}$
5: $\Xi \leftarrow \Lambda_{d,f}(\Pi, T_\Sigma)$
6: repeat
7:     for $P \in \Sigma_R$ do
8:         if $\exists P$ occurs in $\Xi$ and $\exists P^-$ does not then
9:             Add a type set $\Xi' \subseteq T_\Sigma - \Xi$ satisfying the following:
10:                for any $\tau \in \Xi'$, for any $\tau' \in T_\Sigma - \Xi$, $\lambda_{d,f}(\tau, \Pi) \leq \lambda_{d,f}(\tau', \Pi)$
11:             $\Xi \leftarrow \Xi \cup \Xi'$
12:         end if
13:     end for
14: until The role coherence holds
15: return $\Xi$
16: end procedure

Definition 6. Let $\Sigma$ be a signature, $T$ a TBox, and, $\phi$ an inclusion over $\Sigma$. Given a distance function $d$ and an aggregation function $f$, $T$ distance-based entails (d-entails) $\phi$, denoted by $T \models_{d,f} \psi$, if $\Lambda_{d,f}^+(\Pi_{\Sigma}(T), T_\Sigma) \subseteq \text{Mod}^T(\{\phi\})$.

In Example 3, $T \not\models_{d^H,f^\star} A \subseteq \exists P$ and in Example 4, $T \models_{d^H,f^\star} \top \subseteq A$.

Note that, in Definition 8, for an axiom $\phi$, if $\phi$ is a non-contradiction (i.e., $\{\phi\}$ is consistent) then $\text{Mod}^T(\{\phi\}) = \text{Mod}^T_{d,f}(\phi)$ by Corollary 1. Otherwise, $\text{Mod}^T(\{\phi\}) = \emptyset$ while $\text{Mod}^T_{d,f}(\phi) \neq \emptyset$. In fact, contradictions are always assigned to “false” in classical logic. We define the distance-based entailment by
adopting $\text{Mod}^T(\{\phi\})$ instead of $\text{Mod}^T_{d,f}(\{\phi\})$ to ensure that all contradictions cannot be inferred under our paraconsistent semantics.

4. Distance-based semantics for knowledge bases

In this section we introduce the notion of minimal features for DL KBs. Compared with inconsistency of TBoxes, inconsistency occurring in KBs is much more complex since it contains two extra cases: inconsistency of ABoxes and inconsistency between TBoxes and ABoxes. Specifically, these inconsistencies occur in concept assertions, between concept assertions and role assertions, between assertions and inclusions, even between a single inclusion and a single assertion. For instance,

Example 5. Let $\mathcal{K} = (\{\exists P^- \sqsubseteq \bot\}, \{\exists P(a)\})$ be a KB and $\Sigma = \{P, a, 1\}$. $\mathcal{K}$ is inconsistent and thus has no model feature.

To deal with those various inconsistencies in a unified way, we adopt the technique of using minimal model types to construct minimal model features instead of directly introducing distance function over features. We argue that this adoption cannot only uniformly deal with all inconsistencies but also overcome difficulty of defining distance function over pairs of features [49].

We first introduce concept profiles and then use type distance function to describe how far apart features are. Let $\Sigma$ be a signature and $\mathcal{A}$ an ABox over $\Sigma$. Assume that $N_{\mathcal{A}}$ a set of all named individuals in $\mathcal{A}$. $A_R = \{P(a, b) | P(a, b) \text{ or } P^-(b, a) \in \mathcal{A}\}$. A concept profile of $\mathcal{A}$, denoted by $\Sigma_C(a)$, is defined as follows:

$$
\Sigma_C(a) = \bigcup_{D(a) \in \mathcal{A}} \{D\} \\
\cup \bigcup_{P(a, b_1),...,P(a, b_n) \in A_R} \{\geq m P | m \in \Sigma_N, m \leq n\} \\
\cup \bigcup_{P(b_1, a),...,P(b_n, a) \in A_R} \{\geq m P^- | m \in \Sigma_N, m \leq n\}.
$$

Intuitively, a set of concept profiles is a partition of concepts that are realized in that ABox w.r.t. individuals.

For instance, let $\Sigma = \{C, D, P, a, b_1, b_2, 1, 2\}$ and $\mathcal{A} = \{(C \sqcap D)(a), P(a, b_1), P(a, b_2), D(b_1)\}$. Thus $\Sigma_C(a) = \{C \sqcap D, \exists P, \geq 2 P\}$, $\Sigma_C(b_1) = \{D, \exists P^\neg\}$, and $\Sigma_C(b_2) = \{\exists P^-\}$. 

15
Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB. We extend the signature $\text{Sig}(\mathcal{K})$ of $\mathcal{K}$ as $\text{Sig}^*(\mathcal{K}) = \text{Sig}(\mathcal{T}) \cup \text{Sig}(\Sigma_C(\mathcal{A}))$ where $\Sigma_C(\mathcal{A}) = \bigcup_{a \in N_A} \Sigma_C(a)$. Indeed, $\text{Sig}^*(\mathcal{K})$ is obtained from $\text{Sig}(\mathcal{K})$ by adding all possible natural numbers occurring all concept profiles but not occurring $\mathcal{K}$. In other words, $\text{Sig}^*(\mathcal{K})$ and $\text{Sig}(\mathcal{K})$ are no different except $\Sigma_N$.

For instance, in the above example, $\text{Sig}(\mathcal{A}) = \{C, D, P, a, b_1, b_2\}$ while $\text{Sig}^*(\mathcal{A}) = \{C, D, P, a, b_1, b_2, 1, 2\}$.

Next, we will define the notion of minimal model features.

**Definition 7.** Let $\Sigma$ be a signature and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a KB over $\Sigma$. Denote $\Pi_{\Sigma}(a) = \{T_{\Sigma}(D) \mid D \in \Sigma_C(a)\}$. Given a distance function $d$ and an aggregation function $f$, a $(d,f)$-minimal model feature of $\mathcal{K}$ is a feature $\mathcal{F} = (\Xi, \mathcal{H})$ satisfying the following four conditions:

1. $\Xi \subseteq \Lambda^+_{d,f}(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma})$;
2. for each $P \in \Sigma_R$, $\exists P \in \bigcup \Xi$ if and only if $\exists P^{-} \in \bigcup \Xi$;
3. $\tau \in \Lambda^+_{d,f}(\Pi_{\Sigma}(a), \Lambda^+_f(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma})) \cap \Xi$ for each $a \in \Sigma_I$ and $\tau(a) \in \mathcal{H}$;
4. for any role assertion $P(a, b) \in \mathcal{A} - \mathcal{H}$,
   - either $(\geq n + 1) P(a) \not\in \mathcal{H}$ and $P(a, b_1), \ldots, P(a, b_n) \in \mathcal{H}$;
   - or $(\geq n + 1) P^{-}(b) \not\in \mathcal{H}$ and $P(a_1, b), \ldots, P(a_n, b) \in \mathcal{H}$.

Let $\text{Mod}^{F}_{d,f}(\mathcal{K})$ be the set of $(d,f)$-minimal model features of $\mathcal{K}$.

In Definition 7, a minimal model feature is a feature $\mathcal{F}$ which contains two parts, namely, type set $\Xi$ and Herbrand set $\mathcal{H}$. The first condition requires that all types of $\Xi$ be minimal; the second says that $\Xi$ should be a model type set, i.e., it satisfies the property of role coherence; the third guarantees that each type of $\Xi$ satisfying each concept assertion in $\mathcal{H}$ has the minimal distance function to its corresponding concept profile, that is, if a concept assertion $D(a)$ satisfied by $\mathcal{H}$ then types satisfying $D$ are minimal w.r.t. type group $\Pi_{\Sigma}(a)$ of concept profile $\Sigma_C(a)$; and the last ensures that $\mathcal{F}$ is consistent by those role assertions conflicting with concept assertions.

Based on Algorithms 50, Algorithm 1], given a KB $\mathcal{K}$, a distance $d$, an aggregation function $f$, we propose a naive algorithm called $\text{MinModelFeature Algorithm}$ to compute all $(d,f)$-minimal model features of $\mathcal{K}$ shown in Algorithm 2 where Algorithm 1 is applied to compute $\Lambda^+(\cdot)$ function.
Algorithm 2 The MinModelFeature algorithm

1: procedure MinModelFeature($T$, $A$, $d$, $f$)  
   // $A = \{\psi_1, \ldots, \psi_n\}$
2:   $\Sigma \leftarrow \text{Sig}(T) \cup \text{Sig}(A)$
3:   $F \leftarrow \emptyset$
4:   $\Xi_T \leftarrow \Lambda^+_{d,f}(\Pi_{\Sigma}(T), T_\Sigma)$
5:   for $a \in \Sigma_I$ do
6:     $\Pi_{\Sigma}(a) \leftarrow \{T_\Sigma(D) \mid D \in \Sigma_C(a)\}$
7:     $\Xi_a \leftarrow \Lambda^+_{d,f}(\Pi_{\Sigma}(a), \Lambda^+_{d,f}(\Pi_{\Sigma}(T), T_\Sigma))$
8:   end for
9:   repeat
10:      Compute a Herbrand set $H$ such that
11:         $RA^+(A) \subseteq H$ and $H \cap RA^-(A) = \emptyset$, and
12:         For each $a \in \Sigma_I$, the following satisfies: $\tau^H_a \in \Xi_a$
13:      $\Xi \leftarrow \{\tau^H_a \mid a \in \Sigma_I\}$
14:      repeat
15:         if $\exists P$ occurs in $\Xi$ and $\exists P^-$ does not then
16:            Add a type in $\Xi_T$ containing $\exists P^-$ to $\Xi$
17:         end if
18:      until $\Xi$ does not change
19:      $F \leftarrow F \cup \{\langle \Xi, H\rangle\}$
20:   until All such pairs $\langle \Xi, H\rangle$ are in $F$
21: return $F$
22: end procedure

Next, we give a running example to illustrate how to compute (d,f)-minimal model features by applying Algorithm 1 and Algorithm 2

Example 6. Let $K = (T, A)$ where $T = \{\text{Penguin} \sqsubseteq \text{Bird}, \text{Swallow} \sqsubseteq \text{Bird}, \text{Bird} \sqsubseteq \text{Fly}, \text{Fly} \sqsubseteq \exists \text{hasWing}\}$ and $A = \{\text{Penguin(tweety)}, \neg\text{Fly(tweety)}, \text{Swallow(fred)}\}$. Let $\Sigma = \{\text{Penguin, Swallow, Bird, Fly, tweety, Fred, hasWing}\}$. Thus we have $\Sigma_C(\text{tweety}) = \{\text{Penguin, \neg Fly}\}$ and $\Sigma_C(\text{fred}) = \{\text{Swallow}\}$.

We mainly consider the Hamming distance function and the summation function. In total, $T_\Sigma = \{\tau_1, \ldots, \tau_{64}\}$ contains 64 types. Indeed, the 64 types collect all possible $\Sigma$-types.

17
Types | $d_1$ | $d_2$ | $\lambda_{d,H,f}(\tau, \Pi_{\Sigma}(\text{tweety}))$ | $d_4$ | $\lambda_{d,H,f^*}(\tau, \Pi_{\Sigma}(\text{fred}))$
---|---|---|---|---|---
$\tau_1$ | 5 | 0 | 5 | 5 | 5
$\tau_2$ | 5 | 0 | 5 | 5 | 5
$\tau_3$ | 5 | 0 | 5 | 5 | 5
$\tau_4$ | 5 | 0 | 5 | 5 | 5
$\tau_7$ | 5 | 5 | 10 | 5 | 5
$\tau_8$ | 5 | 5 | 10 | 5 | 5
$\tau_{23}$ | 5 | 5 | 10 | 5 | 5
$\tau_{24}$ | 5 | 5 | 10 | 5 | 5
$\tau_{31}$ | 5 | 5 | 10 | 0 | 0
$\tau_{32}$ | 5 | 5 | 10 | 0 | 0
$\tau_{55}$ | 0 | 5 | 5 | 5 | 5
$\tau_{56}$ | 0 | 5 | 5 | 5 | 5
$\tau_{63}$ | 0 | 5 | 5 | 0 | 0
$\tau_{64}$ | 0 | 5 | 5 | 0 | 0

Table 1: Distance on minimal model types

- Firstly, we apply Algorithm 7 to compute $\Xi_T = \Lambda_{d,H,f^*}(\Pi_{\Sigma}(T), T_{\Sigma}) = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_7, \tau_8, \tau_{23}, \tau_{24}, \tau_{31}, \tau_{32}, \tau_{55}, \tau_{56}, \tau_{63}, \tau_{64}\}$ shown in Table 7 where
  - $\tau_1 = \{\}$;
  - $\tau_2 = \{\exists \text{ hasWing}^{-}\}$;
  - $\tau_3 = \{\exists \text{ hasWing}\}$;
  - $\tau_4 = \{\exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$;
  - $\tau_7 = \{\text{Fly}, \exists \text{ hasWing}\}$;
  - $\tau_8 = \{\text{Fly}, \exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$;
  - $\tau_{23} = \{\text{Bird, Fly, } \exists \text{ hasWing}\}$;
  - $\tau_{24} = \{\text{Bird, Fly, } \exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$;
  - $\tau_{31} = \{\text{Bird, Swallow, Fly, } \exists \text{ hasWing}\}$;
  - $\tau_{32} = \{\text{Bird, Swallow, Fly, } \exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$;
  - $\tau_{55} = \{\text{Penguin, Bird, Fly, } \exists \text{ hasWing}\}$;
  - $\tau_{56} = \{\text{Penguin, Bird, Fly, } \exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$;
  - $\tau_{63} = \{\text{Penguin, Swallow, Bird, Fly, } \exists \text{ hasWing}\}$;
  - $\tau_{64} = \{\text{Penguin, Swallow, Bird, Fly, } \exists \text{ hasWing}, \exists \text{ hasWing}^{-}\}$.

Clearly, $\Xi_T$ satisfies the role coherence since $\exists \text{ hasWing}$ occurs in $\Xi_T$ and $\exists \text{ hasWing}^{-}$ does.
• Secondly, we apply Algorithm 2 to compute the followings:
  
  \[ \Sigma_C(\text{tweety}) = \{ \text{Penguin}, \neg \text{Fly} \}; \]
  
  \[ T_\Sigma(\text{Penguin}) = \{ \tau_1, \ldots, \tau_{32} \}; \]
  
  \[ T_\Sigma(\neg \text{Fly}) = \bigcup \{ \tau_i \} \text { where } i = 5, 6, 7, 8, 13, 14, 15, 16, 21, 22, 23, 24, 29, 30, 31, 32, 37, 38, 39, 40, 45, 46, 47, 48, 53, 54, 55, 56, 61, 62, 63, 64; \]
  
  \[ \Pi_\Sigma(\text{tweety}) = \{ T_\Sigma(\text{Penguin}), T_\Sigma(\neg \text{Fly}) \}; \]
  
  \[ \Xi_{\text{tweety}} = \Lambda^+_{dH+f}(\Pi_\Sigma(\text{tweety}), \Lambda^+_{dH+f}(\Pi_\Sigma(\mathcal{T}), T_\Sigma)) = \{ \tau_1, \tau_2, \tau_3, \tau_4, \tau_{55}, \tau_{56}, \tau_{63}, \tau_{64} \}; \]
  
  \[ \Sigma_C(\text{fred}) = \{ \text{Swallow} \}; \]
  
  \[ T_\Sigma(\text{Swallow}) = \{ \tau_1, \ldots, \tau_8, \tau_{17}, \ldots, \tau_{24}, \tau_{33}, \ldots, \tau_{40}, \tau_{49}, \ldots, \tau_{56} \}; \]
  
  \[ \Pi_\Sigma(\text{fred}) = \{ T_\Sigma(\text{Swallow}) \}; \]
  
  \[ \Xi_{\text{fred}} = \Lambda^+_{dH+f}(\Pi_\Sigma(\text{fred}), \Lambda^+_{dH+f}(\Pi_\Sigma(\mathcal{T}), T_\Sigma)) = \{ \tau_{31}, \tau_{32}, \tau_{63}, \tau_{64} \}. \]

• Thirdly, we construct a minimal model feature (Line 10-19, Algorithm 2) in the following steps:
  
  \[ \text{Select a type subset } \Xi_t \text{ of } \Xi_{\text{tweety}} \text{ and a type subset } \Xi_f \text{ of } \Xi_{\text{fred}}; \]
  
  \[ \text{Generate a Herbrand set } \mathcal{H} = \{ \tau(\text{tweety}) \mid \tau \in \Xi_t \} \cup \{ \tau(\text{fred}) \mid \tau \in \Xi_f \}. \]
  
  \[ \text{Let } \Xi = \Xi_t \cup \Xi_f. \text{ And repeat to add some types from } \Xi_T \text{ in } \Xi \text{ till the role coherence holds.} \]

For instance,

\[ \Xi_t = \{ \tau_3 \} \text{ and } \Xi_f = \{ \tau_{31} \}; \]

\[ \text{Let } \mathcal{H} = \{ \exists \text{hasWing(\text{tweety})}, \text{Bird(\text{fred})}, \text{SwallowBird(\text{fred})}, \text{FlyBird(\text{fred})} \}. \]

\[ \text{Let } \Xi = \Xi_t \cup \Xi_f = \{ \tau_3, \tau_{31} \}. \text{ Since } \Xi \text{ does not satisfy the role coherence } (\exists \text{hasWing} \in \Xi \text{ but } \exists \text{hasWing} \neg \notin \Xi), \text{ we add } \tau_8 \text{ in } \Xi \text{ to update } \Xi = \{ \tau_3, \tau_8, \tau_{31} \} \text{ where the role coherence holds.} \]

Thus, we construct a minimal model feature \( \langle \Xi, \mathcal{H} \rangle \). Analogously, we can construct all minimal features.

In Table 1, for simplification, we introduction some abbreviations:

\[ d_1 := d^H(\tau, T_\Sigma(\text{Penguin})); \]

\[ d_2 := d^H(\tau, T_\Sigma(\neg \text{Fly})); \]

\[ d_4 := d^H(\tau, T_\Sigma(\text{Swallow})). \]

In Example 6, we further analyze those conclusions under distance-based semantics. The inconsistency of \( \mathcal{K} \) is caused by statement about \text{tweety}. On the one hand, \text{tweety} is a penguin which cannot fly, i.e., \( \neg \text{Fly(\text{tweety})} \). On the
other hand, a penguin is a bird which can fly, i.e., \textit{Fly(tweety)}. Moreover, there exists no more argument for either \textit{Penguin(tweety)} or \textit{Fly(tweety)}. In this sense, neither \textit{Penguin(tweety)} nor \textit{Fly(tweety)} can be entailed under distance-based semantics. However, the statement about \textit{fred} in $K$ contains no conflict. Thus \textit{Fly(fred)} can be entailed under distance-based semantics.

We find that minimal model features can reach our aim.

**Proposition 3.** Let $\Sigma$ be a signature and $K$ a KB over $\Sigma$. For any distance function $d$ and any aggregation function $f$, we have

1. $\text{Mod}_{d,f}^F(K) \neq \emptyset$;
2. $\text{Mod}_{d,f}^F(K) = \text{Mod}^F(K)$, if $K$ is consistent.

The intuition behind Proposition 3 is that the minimal model features of a KB are the closet features to its classical semantics. Remember that, while an inconsistent KB does not have any model feature, each KB has at least one minimal model feature. An expected result is that the second statement of Proposition 3 does not necessarily hold if $K$ is inconsistent. For instance, in Example 5, $\text{Mod}_{d,f}^F(K) = \{F_1, F_2\}$ where $F_1 = \langle \{\exists P\}, \{\exists P(a)\} \rangle$ and $F_2 = \langle \{\exists P, \exists P^-\}, \{\exists P(a), \exists P^-(a)\} \rangle$ while $\text{Mod}^F(K) = \emptyset$.

Now, based on minimal model features, we are ready to define the \textit{distance-based entailment} for KBs, written $\models_{d,f}$, under which meaningful information can be entailed from an inconsistent KB.

**Definition 8.** Let $\Sigma$ be a signature, $K$ a KB, and, $\phi$ an axiom over $\Sigma$. Given a distance function $d$ and an aggregation function $f$, $K$ distance-based entails (d-entails) $\phi$, still denoted by $K \models_{d,f} \phi$, if $\text{Mod}_{d,f}^F(K) \subseteq \text{Mod}^F(\{\phi\})$.

Distance-based entailment brings a new semantics (called distance-based semantics) for inconsistent KBs by weakening classical entailment. It is not hard to see that no contradiction can be entailed in this semantics. For instance, in Penguin KB, $(\neg \text{Fly} \land \text{Fly})(tweety)$ cannot be entailed but $(\neg \text{Fly} \lor \text{Fly})(tweety)$ can under our semantics.

In the rest of this section, we exemplify that distance-based semantics is suitable for reasoning with inconsistent KBs.

Consequences are intuitive and reasonable under the distance-based semantics. In Penguin KB, $K \models_{d_{\text{H},f}} \text{Fly(fred)}$ while $K \not\models_{d_{\text{H},f}} \text{Penguin(tweety)}$ and $K \not\models_{d_{\text{H},f}} \text{Fly(tweety)}$. We further analyze those conclusions under distance-based semantics. The inconsistency of $K$ is caused by statement about tweety.
On the one hand, *tweety* is a penguin which cannot fly, i.e., \( \neg \text{Fly(tweety)} \). On the other hand, a penguin is a bird which can fly, i.e., \( \text{Fly(tweety)} \). Moreover, there exists no more argument for either \( \text{Penguin(tweety)} \) or \( \text{Fly(tweety)} \). In this sense, neither \( \text{Penguin(tweety)} \) nor \( \text{Fly(tweety)} \) can be entailed under distance-based semantics. However, the statement about *fred* in \( \mathcal{K} \) contains no conflict. Thus \( \text{Fly(fred)} \) can be entailed under distance-based semantics. Additionally, let us consider a simple example: let \( A = \{ A(a), \neg A(a), B(a) \} \). We can conclude that \( A \models_{d_H, f_S} B(b) \) while neither \( A \not\models_{d_H, f_S} A(a) \) nor \( A \not\models_{d_H, f_S} \neg A(a) \).

Furthermore, the distance-based semantics also embodies the idea that conclusions must hold more arguments (i.e., minimally consistent subsets of a KB which can entails a given axiom) [56] for them, for resolving contradictions so that the results are more stable.

**Example 7.** Let \( \mathcal{K} \) be a KB \( \{\{A \sqsubseteq B\}, \{A(a), B(a), \neg B(a)\}\} \). Let \( \Sigma = \text{Sig}^*(\mathcal{K}) = \{A, B, a\} \) and \( \Sigma_C(a) = \{A, B, \neg B\} \). Then \( \Lambda_{d_H, f_S}(\Pi_T, T) = \{\{\neg A, \neg B\}, \{A, \neg B\}, \{A, B\}\} \). Thus \( \text{Mod}_{d_H, f_S}(\mathcal{K}) = \{\langle \Xi, \tau(a) \rangle \mid \Xi \subseteq \Lambda_{d_H, f_S}(\Pi_T, T) \text{ and } \tau \in \Xi \} \). Therefore, \( \mathcal{K} \) can \( d \)-entail \( A(a), B(a) \) and \( A \sqsubseteq B \) except for \( \neg B(a) \). Intuitively, different from \( \neg B(a) \), there is an extra argument \( \{A(a), A \sqsubseteq B\} \) for \( B(a) \).

**5. Properties of distance-based semantics**

In general, our distance-based semantics can be taken as a framework in which many logical consequences are defined by selecting various distance functions and aggregation functions. In this section, we present some useful properties of distance-based semantics.

If \( \mathcal{K} \) is inconsistent and there exists an axiom \( \phi \) such that \( \mathcal{K} \not\models_p \phi \) where \( \models_p \) is an entailment relation, then we say \( \models_p \) is paraconsistent. It is well known that classical entailment \( \models \) is not paraconsistent. We reconsider Example 5 and we have \( \mathcal{K} \models_{d_H, f_S} \exists P \sqsubseteq \perp \) while \( \mathcal{K} \not\models_{d_H, f_S} \exists P(a) \).

The following result shows that the distance-based entailment is paraconsistent.

**Proposition 4.** For any distance function \( d \) and any aggregation function \( f \), \( \models_{d, f} \) is paraconsistent.

Most existing semantics for paraconsistent reasoning in DLs are much weaker than the classical semantics in this sense that there exists a consistent KB \( \mathcal{K} \) and an axiom \( \phi \) such that \( \mathcal{K} \models \phi \) (also called consistency preservation) but \( \phi \)
is not entailed by $\mathcal{K}$ under the paraconsistent semantics. The following result shows that the distance-based semantics does not have such shortcoming.

We can conclude a result directly following Proposition 3.

**Proposition 5.** Let $\Sigma$ be a signature, $\mathcal{K}$ a KB, and, $\phi$ an axiom over $\Sigma$. For any distance function $d$ and any aggregation function $f$, if $\mathcal{K}$ is consistent then $\mathcal{K} \models_{d,f} \phi$ if and only if $\mathcal{K} \models \phi$.

In the classical semantics, a property that $\mathcal{K} \models \psi$ if and only if $\mathcal{T} \models \psi$ for any inclusion $\psi$ is called TBox-preservation [1] where the problem of subsumption checking is irrelevant to ABoxes. Our distance-based semantics satisfies such a property.

**Proposition 6.** Let $\Sigma$ be a signature, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a KB, and, $\psi$ an inclusion over $\Sigma$. For any distance function $d$ and any aggregation function $f$, $\mathcal{K} \models_{d,f} \psi$ if and only if $\mathcal{T} \models_{d,f} \psi$.

By Proposition 6, TBox preservation property means that if the TBox by itself is consistent, then it will be entailed (and hence preference is given to preserving TBox statements over ABox statements), such as the same treatment in [32]. This is different from some other approaches to inconsistency-handling in DLs, where the TBox and ABox are treated equally, or the ABox is given preference such as [34, 56, 58].

The closure w.r.t. $\models_{d,f}$ of an arbitrary KB is always consistent.

**Proposition 7.** Let $\Sigma$ be a signature and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a KB over $\Sigma$. For any distance function $d$ and any aggregation function $f$, let

1. $\text{Cn}_{d,f}(\mathcal{T}) = \{ \psi \text{ is an inclusion} \mid \mathcal{T} \models_{d,f} \psi \}$;
2. $\text{Cn}_{d,f}^\mathcal{T}(\mathcal{A}) = \{ \varphi \text{ is an assertion} \mid (\mathcal{T}, \mathcal{A}) \models_{d,f} \varphi \}$.

We conclude that both $\text{Cn}_{d,f}(\mathcal{T})$ and $\text{Cn}_{d,f}^\mathcal{T}(\mathcal{A})$ are consistent.

Proposition 7 provides a theoretical foundation of applying our approach to inconsistency-tolerant conjunctive query answering [8]. Formally, a conjunctive query (CQ) $q(\bar{x})$ is a first-order formula $\exists \bar{y} \varphi(\bar{x}, \bar{y})$, where $\varphi$ is constructed, using $\wedge$, from atoms of the form $A_k(t_1)$ and $P_k(t_1, t_2)$, where each $t_i$ is a term (an individual or a variable from $\bar{x}$ or $\bar{y}$) [7]. Given a KB $\mathcal{K}$ (possibly inconsistent) and a CQ $q(\bar{x})$, $\mathcal{K} \models_{d,f} q(\bar{s})$ if and only if $\bar{s}$ is a solution of CQ $q$ under our distance-based semantics. By Proposition 7 we can conclude that $\text{Cn}_{d,f}(\mathcal{K}) \models$
q(\bar{s}) where \( Cn_{d,f}(K) = (Cn_{d,f}(T), Cn_{d,f}^T(A)) \) if and only if \( \bar{s} \) is a resolution of CQ \( q \) under our distance-based semantics.

Let \( \Sigma \) be a signature. A distance function \( d \) is \( \Sigma \)-unbiased, if for any \( \Sigma \)-concept \( C \) and any two \( \Sigma \)-types \( \tau_1, \tau_2, B \in \tau_1 \) if and only if \( B \in \tau_2 \) for any basic concept \( B \) occurring in \( C \) implies \( d(\tau_1, T_\Sigma(C)) = d(\tau_2, T_\Sigma(C)) \). The Hamming distance function and the drastic distance function are unbiased.

Let us consider a distance function \( d^j \) defined as follows: for any two sets \( S_1, S_2, d^j(S_1, S_2) = 0 \) if \( S_1 = S_2 \); and \( d^j(S_1, S_2) = 1 + |S_1 \cup S_2| \). It clearly concludes that \( d^j(S_1, S_2) = 0 \) if and only if \( S_1 = S_2 \) and \( d^j(S_1, S_2) = d^j(S_2, S_1) \). Thus \( d^j \) is a distance function. Let \( \Sigma = \{A_1, A_2, A_3, A_4\} \) and \( C = A_1 \cap A_2 \). For each type \( \tau \in T_\Sigma(C), \{A_1, A_2\} \subseteq \tau \). Let \( \tau_1 = \{A_1, A_2\} \) and \( \tau_2 = \{A_1, A_2, A_3, A_4\} \). Thus \( d(\tau_1, T_\Sigma(C)) = 5 \) and \( d(\tau_2, T_\Sigma(C)) = 7 \). Then \( d^j \) is not unbiased.

Unbiasedness will bring a good property of relevance in reasoning since the unbiased distance is not sensitive to those irrelevant basic concepts.

**Proposition 8.** Let \( \Sigma \) be a signature, \( K \) a KB, and, \( \phi \) a non-tautology over \( \Sigma \). If \( d \) is an unbiased function and \( \text{Sig}(K) \cap \text{Sig}(\{\phi\}) = \emptyset \) then for any aggregation function \( f, K \models_{d,f} \phi \).

Note that Proposition 8 does no longer hold for tautologies. Let \( K \) be a KB and \( \phi \) a tautology with \( \text{Sig}(K) \cap \text{Sig}(\phi) = \emptyset \), let \( \Sigma = \text{Sig}(K) \cup \text{Sig}(\phi) \), we can conclude that for any distance function \( d \) and any aggregation function \( f, K \models_{d,f} \phi \) since all possible \( \Sigma \)-features can satisfy \( \phi \).

An entailment relation \( \models_m \) is monotonic if \( K' \models_m \phi \) implies \( K \models_m \phi \) for any KB \( K' \subseteq K \); and nonmonotonic otherwise. Another characteristic property of \( \models_{d,f} \) is its non-monotonic nature.

**Proposition 9.** For any distance function \( d \) and any aggregation function \( f, \models_{d,f} \phi \) is non-monotonic.

While the distance-based semantics is non-monotonic in general, it satisfies a kind of cautious monotonicity, which is usually referred to as splitting property \( \Pi \).

For instance, let a KB \( K = (\{\top \subseteq A, A \subseteq \exists P, \exists P \subseteq \bot, \exists P_1 \subseteq \exists P_2\}, \{\exists P_1 \subseteq \exists P_2\}, \{\exists P_1(b)\}) \) and \( K_2 = (\{\top \subseteq A, A \subseteq \exists P, \exists P \subseteq \bot\}, \{\exists P(a)\}) \). Then, let \( \phi = P_3(b) \), the problem of deciding \( K \models_{d,f} \phi \) can be reduced to checking whether \( K_1 \models_{d,f} \phi \). Note that \( K_1 \) is consistent and \( \text{Sig}(K_1) \cap \text{Sig}(K_2) = \emptyset \). For some non-monotonic semantics, these two conditions are sufficient to guarantee
the validity of the splitting property. However, it is not the case when $K_1$ contains more than one axiom. For this reason, we need hereditary aggregation functions to really guarantee the validity of the splitting property [1].

We say $K$ is split into $K'$ and $K''$, denoted $K = K' \oplus K''$, if (1) $K = K' \cup K''$, and (2) $\text{Sig}(K') \cap \text{Sig}(K'') = \emptyset$.

An aggregation function $f$ is hereditary if and only if $f(\{x_1, \ldots, x_n\}) < f(\{y_1, \ldots, y_n\})$ implies $f(\{x_1, \ldots, x_n, z_1, \ldots, z_m\}) < f(\{y_1, \ldots, y_n, z_1, \ldots, z_m\})$ for any $z_1, \ldots, z_m$. For instance, the summation function is hereditary while the maximum function is not hereditary.

**Proposition 10.** Let $\Sigma$ be a signature and $K$ a KB over $\Sigma$. Assume that $K = K' \oplus K''$ where $K'$ is consistent. For each axiom $\phi$ with $\text{Sig}(\phi) \cap \text{Sig}(K'') = \emptyset$, if $K' \models \phi$ then for any distance function $d$ and any hereditary aggregation function $f$, $K \models_d f \phi$.

One advantage of the splitting property is that paraconsistent reasoning in KB $K$ can be localized into the classical reasoning in a consistent module of $K$, which is usually smaller than the original $K$. Such a property can be very useful for a highly distributed ontology system.

A relation $\equiv$ is cautious if it satisfies:

- (cautious reflexivity) If $K = K' \oplus K''$ and $K'$ is consistent, then $K \equiv \varphi$ for all axiom $\varphi \in K'$;
- (cautious monotonicity) If $K \equiv \varphi$ and $K \equiv \psi$, then $K \cup \{\varphi\} \equiv \psi$;
- (cautious cut) If $K \equiv \varphi$ and $K \cup \{\varphi\} \equiv \psi$ then $K \equiv \psi$.

**Proposition 11.** For any distance function $d$ and any monotonic hereditary aggregation function $f$, $\models_d f$ is cautious.

The following example shows that the cautious principle can bring reasonable results.

**Example 8.** Let an ABox $A = \{\text{HasWife}(Mike, Rose), \text{HasWife}(Mike, Mary), \neg(\geq 2 \text{HasWife})(Mike)\}$. Let $\Sigma = \{\text{HasWife}, \text{Mike}, \text{Mary}, \text{Rose}, 1, 2\}$. The first statement claims that Mike has at most one wife. Moreover, we are informed that Mike has two wives Rose and Mary. We conclude that $A$ is inconsistent and $A \models_d f_\phi (\geq 1 \text{HasWife})(Mike)$ while $A \not\models_d f_\phi \neg \text{HasWife}(Mike, Rose)$, and $A \not\models_d f_\phi \text{HasWife}(Mike, Mary)$. Intuitively, Mike has a wife while we don’t know whether his wife is Rose or Mary under our distance-based semantics.
The distance-based semantics w.r.t. the Hamming distance function is stronger
the semantics w.r.t. the drastic distance function.

**Proposition 12.** Let $\Sigma$ be a signature, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a KB, and, $\phi$ an axiom over $\Sigma$. For any aggregation function $f$, if $\mathcal{K} \models_{d^\mathcal{D}, f} \phi$ then $\mathcal{K} \models_{d^\mathcal{H}, f} \phi$.

In Proposition 12, the converse does not hold.

**Example 9.** Let $\mathcal{A} = \{A(a), (\neg A \cap \exists P)(a), \neg \exists P(a)\}$ be an inconsistent ABox and $\Sigma = \{A, P, a\}$. Thus $\mathcal{A} \models_{d^\mathcal{H}, f^\mathcal{M}} (\neg A \cup \exists P)(a)$ while $\mathcal{A} \not\models_{d^\mathcal{D}, f^a} (\neg A \cup \exists P)(a)$.

In general, different result for a KB would be brought by selecting different
distance function and different aggregation. For instance, in Example 5
$\mathcal{A} \models_{d^\mathcal{H}, f^\mathcal{2}} A(a)$ while $\mathcal{A} \not\models_{d^\mathcal{H}, f^a} A(a)$.

The next result shows that for any $\kappa$-voting function, the greater $\kappa$ is, the
stronger the power of distance-based entailment is.

**Proposition 13.** Let $\Sigma$ be a signature, $\mathcal{K}$ a KB, and, $\phi$ an axiom over $\Sigma$. For any
two voting indexes $\kappa_1, \kappa_2 \in (0, 1)$ and any distance function $d$, if $\kappa_1 \leq \kappa_2$ then $\mathcal{K} \models_{d, f, \kappa_1} \phi$ implies $\mathcal{K} \models_{d, f, \kappa_2} \phi$.

6. Distance-based inconsistency measurement

Inconsistency is mostly caused by over-defining. Different from classical entailment where every axiom in a KB is entailed by the KB, distance-based entailment is rather cautious by making some axioms possibly causing inconsistency invalid. So naturally, we want to know the value or possibility of axioms in causing inconsistency of a KB (see [26]). In this section, as an application of the distance-based semantics, we propose a distance-based inconsistency measurement in which we develop two distance-based inconsistency measures to provide more informative metrics which can tell the differences between axioms causing inconsistency and between inconsistent KBs.

6.1. Inconsistency measures in TBoxes

Firstly, we introduce minimal inconsistency value of axioms in TBoxes.

**Definition 9.** Let $\mathcal{T} = \{\psi_1, \ldots, \psi_n\}$ be a TBox, $\psi$ an axiom, $\Sigma = \text{Sig}(\mathcal{T} \cup \{\psi\})$ a signature, $d$ a distance and $f$ an aggregation function. The minimal inconsistency value of $\psi$ in $\mathcal{T}$ w.r.t. $d$ and $f$, denoted by $\text{MIV}_{d, f}(\mathcal{T}, \psi)$, is defined as follows:

$$\text{MIV}_{d, f}(\mathcal{T}, \psi) = \frac{|\Lambda_{d, f}(\Pi(\mathcal{T}), T_\Sigma) - T_\Sigma(\psi)|}{|\Lambda_{d, f}(\Pi(\mathcal{T}), T_\Sigma)|}.$$  \hspace{1cm} (3)
In Example 3, \( \text{MIV}_{d,f}(T, \top \sqsubseteq A) = \text{MIV}_{d,f}(T, A \sqsubseteq \exists P) = \text{MIV}_{d,f}(T, \exists P \sqsubseteq \bot) = \frac{1}{3} \). In other words, three axioms of \( T \) have the same possibility of causing inconsistency.

By Definition 9, we conclude some properties of the minimal inconsistency value in the following proposition.

**Proposition 14.** Let \( T = \{\psi_1, \ldots, \psi_n\} \) be a TBox and \( \psi \) an axiom. For any distance \( d \) and any aggregation function \( f \), we have

1. \( 0 \leq \text{MIV}_{d,f}(T, \psi) \leq 1 \);
2. \( \text{MIV}_{d,f}(T, \psi) = 0 \) if and only if \( T \models_{d,f} \psi \);
3. \( \text{MIV}_{d,f}(T, \psi) = 1 \) if \( \psi \) is a contradiction.
4. \( \text{MIV}_{d,f}(T, \psi') = \text{MIV}_{d,f}(T, \psi'') \) for any two axioms \( \psi', \psi'' \) which are neither tautologies nor contradictions and \( \text{Sig}(T) \cap \text{Sig}(\{\psi'\}) = \text{Sig}(T) \cap \text{Sig}(\{\psi''\}) = \emptyset \).

Secondly, we introduce inconsistency degree of TBoxes.

**Definition 10.** Let \( T = \{\psi_1, \ldots, \psi_n\} \) be a TBox, \( \Sigma = \text{Sig}(T) \) a signature, \( d \) a distance and \( f \) an aggregation function. The minimal inconsistency value of \( T \) w.r.t. \( d \) and \( f \), denoted by \( \text{MIV}_{d,f}(T) \), is defined as follows:

\[
\text{MIV}_{d,f}(T) = \sum_{1 \leq i \leq n} \frac{\text{MIV}_{d,f}(T, \psi_i)}{|T|}.
\]  

In Example 3, \( \text{MIV}_{d,f}(T) = \frac{1}{3} \).

We extend the notion of basic inconsistency measure proposed by [26] in DL-Lite. An inconsistency measure \( M \) is called a basic inconsistency measure if it satisfies the following properties as required postulates, for any two KBs \( \mathcal{K}, \mathcal{K}' \) and any two axioms \( \psi, \psi' \),

- **(Consistency)** \( M(\mathcal{K}) = 0 \) if and only if \( \mathcal{K} \) is consistent;
- **(Monotony)** \( M(\mathcal{K} \cup \mathcal{K}') \geq M(\mathcal{K}) \);
- **(Free Axiom Independence)** If \( \psi \) is a free axiom of \( \mathcal{K} \cup \{\psi\} \), then \( M(\mathcal{K} \cup \{\psi\}) = M(\mathcal{K}) \) where \( \psi \) is free in \( \mathcal{K} \) if \( \psi \) is not belong to any minimal inconsistent subsets of \( \mathcal{K} \);
(Dominance) If \( \{ \psi \} \models \psi' \) and \( \{ \psi \} \not\models \top \sqsubseteq \bot \), then \( M(\mathcal{K} \cup \{ \psi \}) \geq M(\mathcal{K} \cup \{ \psi' \}) \).

The following proposition shows that our minimal inconsistency value is a basic inconsistency measure.

**Proposition 15.** For any distance \( d \) and any aggregation function \( f \), \( \text{MIV}_{d,f} \) is a basic inconsistency measure, i.e., it satisfies the properties of Consistency, Monotonicity, Free Axiom Independence, and Dominance.

### 6.2. Inconsistency measures in KBs

We extend the minimal inconsistency value of axioms in a TBox and TBoxes to general KBs in the following two definitions respectively.

**Definition 11.** Let \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) be a KB, \( \psi \) an axiom, \( \Sigma = \text{Sig}(\mathcal{K} \cup \{ \psi \}) \) a signature, \( d \) a distance and \( f \) an aggregation function. The minimal inconsistency value of \( \psi \) in \( \mathcal{K} \) w.r.t. \( d \) and \( f \), denoted by \( \text{MIV}_{d,f}(\mathcal{K}, \psi) \), is defined as follows:

\[
\text{MIV}_{d,f}(\mathcal{K}, \psi) = \frac{|\text{Mod}_{d,f}(\mathcal{K}) - \text{Mod}_{d,f}(\{ \psi \})|}{|\text{Mod}_{d,f}(\mathcal{K})|}.
\]  

(5)

The following proposition shows that \( \text{MIV}_{d,f}(\mathcal{K}, \psi) \) is a natural extension of \( \text{MIV}_{d,f}(\mathcal{T}, \psi) \).

**Proposition 16.** Let \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) be a KB, \( \psi \) a concept inclusion, \( \Sigma = \text{Sig}(\mathcal{K} \cup \{ \psi \}) \) a signature, \( d \) a distance and \( f \) an aggregation function. If \( \mathcal{A} = \emptyset \) then \( \text{MIV}_{d,f}(\mathcal{K}, \psi) = \text{MIV}_{d,f}(\mathcal{T}, \psi) \).

Furthermore, a good result is shown in the following proposition:

**Proposition 17.** Let \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) be a KB, \( \psi \) a concept inclusion, \( \Sigma = \text{Sig}(\mathcal{K} \cup \{ \psi \}) \) a signature. For any distance \( d \) and any aggregation function \( f \), we have \( \text{MIV}_{d,f}(\mathcal{K}, \psi) = \text{MIV}_{d,f}(\mathcal{T}, \psi) \).

Analogously, we use \( \text{MIV}_{d,f}(\mathcal{K}, \psi) \) to define minimal inconsistency value of KBs.

**Definition 12.** Let \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) be a KB, \( \psi \) an axiom, \( \Sigma = \text{Sig}(\mathcal{K} \cup \{ \psi \}) \) a signature, \( d \) a distance and \( f \) an aggregation function. The minimal inconsistency value of \( \mathcal{K} \) w.r.t. \( d \) and \( f \), denoted by \( \text{MIV}_{d,f}(\mathcal{K}) \), is defined as follows:

\[
\text{MIV}_{d,f}(\mathcal{K}) = \frac{\sum_{1 \leq i \leq n} \text{MIV}_{d,f}(\mathcal{K}, \psi_i)}{|\mathcal{K}|}.
\]  

(6)
For instance, in Example 6, $MIV_{d,f}(\mathcal{K}) = \frac{1}{2}$. Intuitively, one of two individuals brings inconsistency in Penguin KB.

We have the same property stated in Proposition 17.

**Proposition 18.** Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and $\Sigma = \text{Sig}(\mathcal{K} \cup \{\psi\})$ a signature. For any distance $d$ and any aggregation function $f$, if $\mathcal{A} = \emptyset$ then $MIV_{d,f}(\mathcal{K}) = MIV_{d,f}(\mathcal{T})$.

The following result shows that the extended minimal inconsistency measure is also a basic inconsistency measure.

We can define the properties of Consistency, Monotonicity, Free Axiom Independence, and Dominance of inconsistency measure for general KBs analogously.

**Proposition 19.** For any distance $d$ and any aggregation function $f$, the extended minimal inconsistency measure $MIV_{d,f}$ satisfies the properties of Consistency, Monotonicity, Free Axiom Independence, and Dominance.

Finally, we show that minimal inconsistency measure $MIV_{d,f}$ is normalized so that we can analyze inconsistent KBs with different size in a unified way.

**Proposition 20.** For any KB $\mathcal{K}$, for any distance $d$ and any aggregation function $f$, the extended minimal inconsistency measure $MIV_{d,f}(\mathcal{K}) \in [0, 1]$.

By Proposition 19, $MIV_{d,f}(\mathcal{K}) = 0$ if $\mathcal{K}$ is consistent. And we also conclude that $MIV_{d,f}(\mathcal{K}) = 1$ if all axioms of $\mathcal{K}$ are contradictions since $MIV_{d,f}(\mathcal{K}, \psi) = 1$ for any contradiction $\psi$.

7. Experiments

In this section, based on our definitions, we implement and evaluate the MinModelType algorithm to compute the minimal model types of a given TBox, which is a key module in distance-based reasoning (i.e., the MinModelFeature algorithm) and distance-based inconsistency measurement.

The experiments were performed under Windows 7 64-bit on a Intel i5-4430S, 2.70GHz CPU system with 8GB memory. The program were written in Java 1.7 with maximum 4GB heap space allocated for JVM. Eight tested ontologies by revising some popular ontologies mentioned in existing works such as “PhD.owl” and “Staff.owl” in [51], “Car.owl” in [19], “Penguin.owl” and “Pet.owl” in [57], “HarryPotter.owl” in [20], “BadFood.owl” and “BuggyPolicy.owl” in [27] are selected for our experiments. Note that ontologies “PhD.owl”,...
“Car.owl”, and “HarryPotter.owl” are consistent and ontologies (“Penguin.owl”, “Staff.owl”, “Pet.owl”, “BadFood.owl”, and “BuggyPolicy.owl” are inconsistent. In our experiments, we consider six cases with two distance functions (Hamming distance and Drastic distance) and three aggregation functions (1/2-voting, maximum, and summation) in total. All stated runtimes are averaged over five independent runs. Note that all results are in milliseconds. More results can be found in an online demo: http://123.56.79.184/modelType.html.

<table>
<thead>
<tr>
<th>Ontology Name</th>
<th>Axiom Count</th>
<th>Distance Function</th>
<th>Aggregation: time (ms)/types</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1/2-voting</td>
</tr>
<tr>
<td>PhD.owl</td>
<td>4</td>
<td>Hamming</td>
<td>1408/15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>863/15</td>
</tr>
<tr>
<td>Car.owl</td>
<td>5</td>
<td>Hamming</td>
<td>3133/414</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>8168/414</td>
</tr>
<tr>
<td>HarryPotter.owl</td>
<td>6</td>
<td>Hamming</td>
<td>4921/510</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>8168/414</td>
</tr>
<tr>
<td>Penguin.owl</td>
<td>3</td>
<td>Hamming</td>
<td>1145/8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>422/8</td>
</tr>
<tr>
<td>Staff.owl</td>
<td>4</td>
<td>Hamming</td>
<td>1686/32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>533/32</td>
</tr>
<tr>
<td>Pet.owl</td>
<td>5</td>
<td>Hamming</td>
<td>1469/30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>645/30</td>
</tr>
<tr>
<td>BadFood.owl</td>
<td>6</td>
<td>Hamming</td>
<td>1899/128</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>1671/128</td>
</tr>
<tr>
<td>BuggyPolicy.owl</td>
<td>7</td>
<td>Hamming</td>
<td>1873/124</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>1732/124</td>
</tr>
</tbody>
</table>

Table 2: Evaluation on the tested ontologies

By Table 2, we find the following phenomenon:

- All ontologies above have also the same number of (minimal) model types under the same distance function.
- The first three consistent ontologies (“PhD.owl”, “Car.owl”, and “HarryPotter.owl”) have the same number of (minimal) model types under six cases. However, the left five inconsistent ontologies have different number of minimal model types under different aggregation functions.
- The number of minimal model types roughly increases with the growth of

29
the number of axioms shown in Figure 1 where we compare the maximal number of minimal models types among five inconsistent ontologies.

- The performance is sensitive to distance functions and aggregation functions to be selected. For instance, in “BadFood.owl”, fixing Hamming distance, the time is 1899ms, 1158ms, and 785ms if we select aggregation function: 1/2-voting, maximum, and summation respectively. And in “BadFood.owl”, fixing the 1/2-voting aggregation function, the time is 1899ms and 1671ms if we select Hamming distance and Drastic distance respectively.

<table>
<thead>
<tr>
<th>Ontology Name</th>
<th>Axiom Count</th>
<th>Distance Function</th>
<th>Aggregation: time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>random10.owl</td>
<td>10</td>
<td>Hamming</td>
<td>109089 115959 110858</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>952289 945432 923961</td>
</tr>
<tr>
<td>random20.owl</td>
<td>20</td>
<td>Hamming</td>
<td>929977 960105 891183</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>5220304 5182608 5213549</td>
</tr>
<tr>
<td>random30.owl</td>
<td>30</td>
<td>Hamming</td>
<td>1159154 1174994 1051551</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>8733766 8683990 8687981</td>
</tr>
<tr>
<td>random40.owl</td>
<td>38</td>
<td>Hamming</td>
<td>1352410 1342545 1273876</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>9560396 9656362 9564169</td>
</tr>
<tr>
<td>random50.owl</td>
<td>49</td>
<td>Hamming</td>
<td>2137814 1912850 1822756</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drastic</td>
<td>16228185 17324756 17616456</td>
</tr>
</tbody>
</table>

Table 3: Evaluation on the generated ontologies

Figure 1: The number of minimal model types increases via axioms
Figure 2: Evaluation on distance functions

![Graph showing evaluation on distance functions with time (s) on the y-axis and #Hamming distance on the x-axis for 1/2-voting, maximum, and summation.]

Figure 3: Evaluation on aggregation functions

![Graph showing evaluation on aggregation functions with time (s) on the y-axis and #1/2-voting on the x-axis for Hamming and Drastic.]

![Graph showing evaluation on aggregation functions with time (s) on the y-axis and #Maximum on the x-axis for Hamming and Drastic.]

![Graph showing evaluation on aggregation functions with time (s) on the y-axis and #Summation on the x-axis for Hamming and Drastic.]

31
To precisely observe the relationship between the performance and the number of axioms, we generate five ontologies based on a benchmark stated in \cite{52} and their reports about six cases are shown in Table 3, where the time is about computing the first solution.

By Table 3, we can conclude that the time of our naive system in computing minimal model types increases with the growth of axioms in a polynomial way.

In short, the efficiency of the MinModelType algorithm is still acceptable since our implementation is naive without any optimization. Moreover, given a distance function, the efficiency of the MinModelType algorithm is slightly different when we select different aggregation functions shown in Figure 2. However, given an aggregation function, we also find that the efficiency of the MinModelType algorithm with the Hamming distance is much higher than that of the MinModelType algorithm with the Drastic distance shown in Figure 3.

8. Related Works

In this section, we compare our approach with existing approaches to handling inconsistency including paraconsistent reasoning and inconsistency measurements.

8.1. Comparison with existing approaches to handling inconsistency

Existing approaches for inconsistency handling in DLs are usually based on various forms of inconsistency-tolerant semantics, for instance, four-valued description logics \cite{34}, quasi-classical description logics \cite{58}, argumentation-based semantics for description logics \cite{28,56}, and, MKNF-based semantics for description logics \cite{21}. Compared to them, our distance-based semantics works on classical interpretations (i.e., two-valued interpretations) but still can draw more useful and reasonable logical consequences. For example, under four-valued DL-Lite, modus ponens, modus tollens and disjunctive syllogism no longer hold. Taking the Penguin KB for example, we cannot derive the intuitive conclusions that the swallow \textit{fred} is a bird and can fly under the four-valued semantics (with the material inclusion interpretation \cite{33,34}). In contrast, quasi-classical DLs \cite{58} can satisfy basic reasoning rules mentioned above. However, they might bring too much contradictory consequences. Considering again the Penguin KB, all assertions about tweety become contradictory under the quasi-classical semantics \cite{58}. Moreover, these approaches do not provide a mechanism of comparing different models for a KB. Additionally, they are usually monotonic such that they do not hold consistency-preserving. And above all, our scenario provides a framework which can be easily extended in a natural way.
to multiple-valued semantics so that multiple-valued logics can be incorporated. The argumentation-based semantics for description logics presented in \[28, 56\] is based on a dialogue process to evaluate the inconsistent knowledge. Our semantics is based on a totally different mechanism from it by using distance between models. Furthermore, different from \[21\] which introduces a weak negation \textbf{not} to tolerate inconsistency, our approach does not change the syntax of description logics.

Some syntax-based approaches \[48, 22, 27, 37, 17, 45, 46\] and ours is different from them in that they take some consistent subsets as substitutes of KBs in reasoning. Similarly to our approach, those syntax-based paraconsistent semantics can satisfy several properties that do not hold in multi-valued semantics, such as non-monotonicity, consistency-preserving and splitting property. But they differ from ours in the following aspects. Firstly, they do not satisfy the closure consistency, that is, their inference closures are not always consistent. Secondly, those syntax-based approaches focus on local information so that they could difficultly capture the semantics of whole a KB. As a result, their results are sensitive to the syntactical structure. Finally, they might bring the multi-extension problem because of limitations of their selection mechanisms so that users make an extra strategic decision.

There are some model-based approaches presented in \[43, 32\]. These approaches try to recover the consistency arising from adding one TBox to another one \[43\]. Instead of working directly on DL models which are generally infinite structures, our approach works on features which are finite structures. Moreover, our approach can handle the inconsistency between two TBoxes by first combining them into one. In addition, under our approach, we can select different distance functions and aggregation functions to meet different needs in various applications. Some repairing approaches are applied to repair ABoxes such that the repaired KB can ensure the union of conjunctive consistent querying when ABoxes that are inconsistent with the TBoxes \[32, 15, 8, 7, 16\]. Although both of the main goal of this work and our work are recovering consistency by repairing KBs, there exists some difference in strategies. In addition, our approach can handle a KB with an inconsistent TBox. Even though TBoxes to be handled are consistent, our approach is more flexible in that, according to a consistent TBox, we can construct different models of an ABox by selecting different distance function and different aggregation function to meet different practical requirement. In other words, we present a framework of distance-based semantics where distance functions and aggregations are parameters.
8.2. *Comparison with existing approaches to inconsistency measurement*

Comparing with those inconsistency measurements in DL-Lite by using multi-valued semantics such as four-valued semantics [35] and three-valued semantics [59], our distance-based inconsistency measurement employs the distance-based semantics which is two-valued. [34] presents a distance-based approach which is proposed to measure inconsistency of TBoxes by only treating finite models of DL KBs. However, this approach is difficult to implement because of infinite models of DL KBs. Using the Shapley Value in [23], Deng, Haarslev, and Shiriri (2007) introduced some inconsistency measurements in DL ontologies [13]. Comparing their approach, our proposal directly works on the semantic interpretations (i.e., types and features) of that KB instead of the syntactic axioms of a KB. In addition, there are some works about incoherence measurements in DLs such as [42]. However, our proposal is about the inconsistency measurement.

9. **Conclusions**

In this paper, we have presented a distance-based framework to handle inconsistency in DL-Lite by introducing distances over features for KBs. Within this framework, we defined a new semantics called distance-based semantics, which is able to characterize a class of cautious entailment relations. A distinguished property of distance-based entailments is that they retain consistency to a large extent. Furthermore, our framework gives consideration to both semantic minimal change and syntactic minimal change. In this sense, our approach is a natural combination of qualitative and quantitative approaches. As a result, our approach is a more fine-grained compared to previous approaches to paraconsistent reasoning in DLs. Though distance-based semantics presented in this paper is built on two-valued semantics, we can also define distance-based semantics on multi-valued semantics within our framework. Moreover, within this framework, we presented a new distance-based inconsistency measurement for DL-Lite KBs, which is proven to be a basic inconsistency measure. Additionally, our measure is normalized.

In our current implementation of computing minimal model types, the performance is not high for handling ontologies in a big size. It would be interesting to develop efficient algorithms and implement them including the distance-based inconsistency measurement in the future. Though, in the revision of ontologies with two distance functions, only a polynomial-sized part of each feature needs to be considered for complexity analysis (in this sense, the complexity coincides with the propositional belief revision), it is still open for general distance functions and aggregation functions. As an important future work, we will discuss...
the computational complexities of distance-based reasoning and distance-based inconsistency measurement. Moreover, the conjunctive query answering (CQA) is an important problem in OWL 2. We are interested in investigating some algorithm for implementing CQA over inconsistent KBs based on our approach.

Acknowledgments

We thank the anonymous referees for their critical comments on this paper, which encouraged us to significantly improve the paper. We also would like to thank Xingwang Han for his help in experiments. This work is supported by the program of the National Key Research and Development Program of China (2016YFB1000603), the National Natural Science Foundation of China (NSFC) (61502336, 61373035), and the open funding project of Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education. Xiaowang Zhang is supported by Tianjin Thousand Young Talents Program and the project-sponsored by School of Computer Science and Technology in Tianjin University.

References

37


Appendix: Proofs

Proof of Proposition 1
Let us consider two cases:

1. For any type $\tau \in T_\Sigma$, we can compute that $\lambda_{d,f}(\tau, \Pi)$ by Definition 5. That is, all $\lambda_{d,f}(\tau, \Pi)$ are comparable. Suppose, for the sake of contradiction, that $\Lambda_{d,f}(\Pi, T_\Sigma) = \emptyset$. Then, by the definition of $\Lambda_{d,f}(\Pi, T_\Sigma)$, for any $\tau \in T_\Sigma$, there exists some $\tau' \in T_\Sigma$ such that $\lambda_{d,f}(\tau', \Pi) < \lambda_{d,f}(\tau, \Pi)$. Thus $T_\Sigma$ is infinite. However, $T_\Sigma$ is finite since $\Sigma$ is finite, we have arrived at a contradiction. Therefore, $\Lambda_{d,f}(\Pi, T_\Sigma) \neq \emptyset$ by the definition of distance functions and aggregation functions.

2. On the one hand, if $\bigcap \Pi \neq \emptyset$ then for any type $\tau \in \bigcap \Pi$, $\lambda_{d,f}(\tau, \Pi) \leq \lambda_{d,f}(\tau', \Pi)$ for any type $\tau' \in T_\Sigma$ since $\tau \in \bigcap \Pi$, i.e., $\lambda_{d,f}(\tau, \Pi) = 0$ by Definition 4 and Definition 5. On the other hand, if $\tau \in \Lambda_{d,f}(\Pi, T_\Sigma)$, then for any type $\tau'' \in T_\Sigma$, $\lambda_{d,f}(\tau, \Pi) \leq \lambda_{d,f}(\tau'', \Pi)$ by the definition. We choose $\tau'' \in \bigcap \Pi \subseteq T_\Sigma$. So $\lambda_{d,f}(\tau, \Pi) = 0$ since $\lambda_{d,f}(\tau'', \Pi) = 0$ by the proof if the first item.

Proof of Corollary 1
The first item directly follows the first item of Proposition 1. In the second item, if $T$ is consistent then $\bigcap \Pi_\Sigma(T) \neq \emptyset$. By the second item of Proposition 1 we can conclude that $\Lambda_{d,f}(\Pi_\Sigma(T), T_\Sigma) = \bigcap(\Pi_\Sigma(T))$.

Proof of Proposition 2
We can use Corollary 1 and the definitions to prove this proposition.

1. The first item directly follows the first item of Corollary 1 and the definition of minimal model type sets since $\Lambda_{d,f}(\Pi_\Sigma(T), T_\Sigma) \subseteq \Lambda_{d,f}^+(\Pi_\Sigma(T), T_\Sigma)$ and $\Lambda_{d,f}(\Pi_\Sigma(T), T_\Sigma) \neq \emptyset$.

2. In the second item, if $T$ is consistent then $\Lambda_{d,f}(\Pi_\Sigma(T), T_\Sigma) = \bigcap(\Pi_\Sigma(T))$ by the second of Corollary 1. Because $\bigcap(\Pi_\Sigma(T))$ is the model type set...
The first item directly follows the first item of Proposition 2 and Definition 7.

Proof of Proposition 3

The first item directly follows the first item of Proposition 2 and Definition 7.

In the second item, if \( P \in \Sigma_R \), \( \exists P \in \bigcup(\Lambda_{d,f}(\Pi\Sigma(T), T_S)) \) if and only if \( \exists P^– \in \bigcup(\Lambda_{d,f}(\Pi\Sigma(T), T_S)). \)

Therefore, \( \Lambda_{d,f}^+(\Pi\Sigma(T), T_S) = \Lambda_{d,f}(\Pi\Sigma(T), T_S) \).

It directly follows the definition of \( \Lambda_{d,f}^+(\Pi\Sigma(T), T_S). \)

- **Proof of Proposition 3**

  The first item directly follows the first item of Proposition 2 and Definition 7.

  In the second item, if \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) is consistent, then, \( \text{Mod}^F(\mathcal{K}) \neq \emptyset. \)

  1. For any feature \( F = \langle \Xi, \mathcal{H} \rangle \in \text{Mod}^F(\mathcal{K}), \) for any feature \( F' = \langle \Xi', \mathcal{H}' \rangle, \) we have

     - (a) for any type \( \tau \in \Xi \) and for any type \( \tau' \in \Xi', 0 = \lambda_{d,f}(\tau, \Pi\Sigma(T)) \leq \lambda_{d,f}(\tau', \Pi\Sigma(T)), \) that is, \( \Xi \subseteq \Xi' \);

     - (b) \( \exists P \in \bigcup \Xi \) if and only if \( \exists P^– \in \bigcup \Xi; \)

     - (c) for each individual \( a \in \Sigma_I \) and \( \tau(a) \in \mathcal{H} \) for some \( \tau \in \Xi, \) for any type \( \tau' \in \Xi' \) and \( \tau'(a) \in \mathcal{H}', \) \( 0 = \lambda_{d,f}(\tau', T_{\Xi}^\mathcal{K}) \leq \lambda_{d,f}(\tau', T_{\Xi}^{\mathcal{H}}), \) that is, \( \tau \in \Lambda_{d,f}(\tau, T_{\Xi}^{\mathcal{K}}); \)

     - (d) for each assertion \( P(a, b) \in \mathcal{A}_R - \mathcal{H}, \) either \( (\leq n) P(a) \in \mathcal{H} \) and \( P(a, b_1), \ldots, P(a, b_{n+1}) \in \mathcal{A}_R, \) or \( (\leq n) P^–(b) \in \mathcal{H} \) and \( P(a_1, b), \ldots, P(a_{n+1}, b) \in \mathcal{A}_R. \)

     Based on (a), (b), (c) and (d), \( F \in \text{Mod}^F_{d,f}(\mathcal{K}). \)

  2. For any feature \( F = \langle \Xi, \mathcal{H} \rangle \in \text{Mod}^F_{d,f}(\mathcal{K}) \) and \( F' = \langle \Xi', \mathcal{H}' \rangle \in \text{Mod}^F(\mathcal{K}) \) since \( \text{Mod}^F(\mathcal{K}) \neq \emptyset, \) we have

     - (a) for any type \( \tau \in \Xi \) and for any type \( \tau' \in \Xi', \lambda_{d,f}(\tau, \Pi\Sigma(T)) \leq \lambda_{d,f}(\tau', \Pi\Sigma(T)) = 0, \) that is, \( \lambda_{d,f}(\tau, \Pi\Sigma(T)) = 0. \) Therefore, \( \tau \in \bigcap(\Pi\Sigma(T)), \) i.e., \( \Xi \subseteq \bigcap \Pi\Sigma(T). \) Therefore, \( F \) satisfies all inclusions in the TBox of \( \mathcal{K}. \)

     - (b) \( \exists P \in \bigcup \Xi \) if and only if \( \exists P^– \in \bigcup \Xi; \)

     - (c) for each assertion \( C(a) \in \mathcal{A}, \) there exists \( \tau' \in \Xi ' \) such that \( \tau' \in T_{\Xi}^{\mathcal{K}} \) and \( \tau'(a) \in \mathcal{H}'. \) For any type \( \tau \in \Xi, \lambda_{d,f}(\tau, T_{\Xi}^\mathcal{K}) \leq \lambda_{d,f}(\tau', T_{\Xi}^{\mathcal{H}}) = 0, \) that is, \( \lambda_{d,f}(\tau, T_{\Xi}^{\mathcal{H}}) = 0. \) Therefore, \( \tau(a) \in \mathcal{H} \) and \( \tau \in T_{\Xi}^\mathcal{K}(C). \) That is, \( F \) satisfies \( C(a). \)
Based on (a), (b), (c), and (d), $\mathcal{F} \in \text{Mod}^F(\mathcal{K})$. Therefore, $\text{Mod}^F(\mathcal{K}) = \text{Mod}_{d,f}^F(\mathcal{K})$.

Proof of Proposition 4
Let $\Sigma$ be a signature. Let $\mathcal{K}$ be a KB over $\Sigma$. For any contradiction $\phi$, for any distance function $d$ and for any aggregation function $f$, we can conclude that $\text{Mod}_{d,f}^F(\mathcal{K}) \neq \emptyset$ by the first item of Proposition 3. Because $\phi$ is a contradiction, $\text{Mod}^F(\{\phi\}) = \emptyset$. Therefore, $\text{Mod}_{d,f}^F(\mathcal{K}) \not\subseteq \text{Mod}^F(\{\phi\})$, that is, $\mathcal{K} \not\models_{d,f} \phi$.

Proof of Proposition 5
This proposition directly follows the second item of Proposition 3.

Proof of Proposition 6
Let $\psi$ be of the form $C \subseteq D$ where $C, D$ are concepts. Let $\Sigma' = \text{Sig}(\mathcal{T} \cup \{C \subseteq D\})$. We can conclude that $\mathcal{K} \models_{d,f} \psi$ if and only if $\text{Mod}_{d,f}^F(\mathcal{K}) \subseteq \text{Mod}^F(\{\psi\})$ by Definition 8. That is, for each $\Sigma$-feature $\mathcal{F} = \langle \Xi, \mathcal{H} \rangle \in \text{Mod}_{d,f}^F(\mathcal{K})$, $\mathcal{F} \in \text{Mod}^F(\{C \subseteq D\})$ by Definition 8. Then $\Xi \subseteq T_{\Sigma'}(-C \cup D)$ and $\mathcal{H}$ is arbitrary since $\psi$ is a concept inclusion. Therefore, $\Lambda_{d,f}^+(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma}) \subseteq T_{\Sigma'}(-C \cup D)$ since for all $\Xi \in \Lambda_{d,f}^+(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma'})$, $\Xi \subseteq T_{\Sigma'}(-C \cup D)$ by Definition 7. We still conclude that $\Lambda_{d,f}^+(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma'}) \subseteq T_{\Sigma'}(-C \cup D)$ since $T_{\Sigma'}(-C \cup D) = T_{\Sigma}(-C \cup D)$ obtained by removing all literals of $\Sigma - \Sigma'$ in $T_{\Sigma}(-C \cup D)$.

Next, we claim that

$$\mathcal{T} \models_{d,f} C \subseteq D \text{ if and only if } \Lambda_{d,f}^+(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma'}) \subseteq T_{\Sigma'}(-C \cup D).$$

Now, we prove this claim. If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{A} = \emptyset$, then $\text{Mod}_{d,f}^F(\mathcal{K}) = \{\mathcal{F}_1, \ldots, \mathcal{F}_m\}$ where $\mathcal{F}_i = \langle \Xi_i, \emptyset \rangle$ $i \in \{1, \ldots, m\}$. Let $\Xi = \bigcup_{i=1}^m \Xi_i$ where $\{\Xi_1, \ldots, \Xi_m\}$ is an enumeration of all possible subsets of $\Xi_{\mathcal{K}}$. For each $\Xi_i (1 \leq i \leq m)$, $\exists P \in \bigcup \Xi_i$ if and only if $\exists P^+ \cup \Xi_i$ for any role name $P \in \Sigma_R$. Thus $\Xi = \Xi_{\mathcal{K}}$. On the other hand, analogously, $\text{Mod}^F(\{C \subseteq D\}) = \{\mathcal{F}_1', \ldots, \mathcal{F}_n'\}$ where $\mathcal{F}_i' = \langle \Xi_i', \emptyset \rangle$ $1 \leq i \leq n$ where $\{\Xi_1', \ldots, \Xi_n'\}$ is an enumeration of all possible subsets of $\Xi_{\mathcal{K}}$ respectively. Therefore, $\mathcal{T} \models_{d,f} C \subseteq D \text{ if and only if } \Xi_{\mathcal{K}} \subseteq T_{\Sigma'}(-C \cup D)$.

Thus $\text{Mod}_{d,f}^F(\mathcal{K}) \subseteq \text{Mod}^F(\{C \subseteq D\})$ if and only if $\{\mathcal{F}_1, \ldots, \mathcal{F}_m\} \subseteq \{\mathcal{F}_1', \ldots, \mathcal{F}_n'\}$. That is, $\{\mathcal{F}_1, \ldots, \mathcal{F}_m\} \subseteq \{\mathcal{F}_1', \ldots, \mathcal{F}_n'\}$ if and only if $\Xi_{\mathcal{K}} \subseteq T_{\Sigma'}(-C \cup D)$ since $\{\Xi_1, \ldots, \Xi_m\}$ and $\{\Xi_1', \ldots, \Xi_n'\}$ are enumerations of all possible subsets of $\Xi_{\mathcal{K}}$ and $T_{\Sigma'}(-C \cup D)$ respectively. Therefore, $\text{Mod}_{d,f}^F(\mathcal{K}) \subseteq \text{Mod}^F(\{C \subseteq D\})$ if and only if $\Xi_{\mathcal{K}} \subseteq T_{\Sigma'}(-C \cup D)$. Therefore, $\mathcal{K} \models_{d,f} \psi$ if and only if $\mathcal{T} \models_{d,f} \psi$.  

40
Proof of **Proposition 7**

We only need to show that \( Cn_{d,f}(T) \not\models T \sqsubseteq \bot \). Assume that \( Cn_{d,f}(T) \models T \sqsubseteq \bot \). Mod\(^T\)\((Cn_{d,f}(T)) = \text{Mod}^T\)\((T)\) since \( Cn_{d,f}(T) \) is the deductive closure of \( \models_{d,f} \) over \( T \). Thus \( \text{Mod}^T_{d,f}(T) \subseteq \text{Mod}^T\{(T \sqsubseteq \bot)\} \) while \( \text{Mod}^T\{(T \sqsubseteq \bot)\} = \emptyset \) and \( \text{Mod}^T_{d,f}(T) \neq \emptyset \) by this claim in the proof of Proposition 6.

Suppose, for the sake of contradiction, that \( Cn^T_{d,f}(A) \) is inconsistent. That is, there is an assertion \( C(a) \) where \( C \) is a concept and \( a \) is an individual name such that \( Cn^T_{d,f}(A) \models C(a) \) and \( Cn^T_{d,f}(A) \models \neg C(a) \). Then \( \text{Mod}^F\((Cn^T_{d,f}(A)) = \text{Mod}^F_{d,f}((\mathcal{T}, A)) \) since \( Cn^T_{d,f}(A) \) is the deductive closure of \( \models_{d,f} \) over \( A \) w.r.t. \( \mathcal{T} \). Thus, \( (\mathcal{T}, A) \models_{d,f} C(a) \) and \( (\mathcal{T}, A) \models_{d,f} \neg C(a) \) at the same time. Then \( \text{Mod}^F_{d,f}((\mathcal{T}, A)) \subseteq \text{Mod}^F\{(C(a))\} \) and \( \text{Mod}^F_{d,f}((\mathcal{T}, A)) \subseteq \text{Mod}^F\{\neg C(a)\} \).

Thus we can conclude that \( \text{Mod}^F_{d,f}((\mathcal{T}, A)) \subseteq \text{Mod}^F\{(C(a))\} \cap \text{Mod}^F\{\neg C(a)\} \) \( = \text{Mod}^F\{(C(a), \neg C(a))\} = \emptyset \), that is, \( \text{Mod}^F_{d,f}((\mathcal{T}, A)) \neq \emptyset \) by Proposition 6, we have arrived at a contradiction.

Proof of **Proposition 8**

Let \( K = (\mathcal{T}, A) \). If \( \phi \) is a contradiction then this claim already holds. Otherwise, let us consider three forms of \( \phi \):

1. If \( \phi \) is of the form \( C \sqsubseteq D \) where \( C, D \) are concepts. By Proposition 6, we only prove that \( \mathcal{T} \not\models_{d,f} \phi \). Let \( \Sigma_1 = \text{Sig}(\mathcal{T}) \) and \( \Sigma_2 = \text{Sig}(\phi) \). Thus \( \Sigma_1 \cap \Sigma = \emptyset \) and \( \Sigma_1 \cup \Sigma_2 \subseteq \Sigma \). Let \( \tau_1 \) be a \( \Sigma_1 \)-type and \( \tau_1 \in \Lambda^+_{d,f}(\Pi_{\Sigma_1}(\mathcal{T}), T_{\Sigma_1}) \) and \( \tau_2 \not\in T_{\Sigma_2}(\neg C \sqcup D) \). By Corollary 1, \( \tau_1 \) exists. Because \( \phi \) is neither a contradiction nor a tautology, \( \tau_2 \) exists. Let \( \tau = \tau_1 \cup \tau_2 \) (i.e., the union of \( \tau_1 \) and \( \tau_2 \)) since \( \tau_1 \cap \tau_2 = \emptyset \). For each \( i \) \( (1 \leq i \leq m) \), for any basic concept \( B \) occurring in \( \neg C_i \sqcup D_i \), \( B \in \tau_1 \) if and only if \( B \in \tau \). Because \( d \) is unbiased, we can conclude that \( d(\tau_1, T_{\Sigma}(\neg C_i \sqcup D_i)) = d(\tau, T_{\Sigma}(\neg C_i \sqcup D_i)) \) for any \( i \in \{1, \ldots, m\} \). Because \( \tau_1 \in \Lambda^+_{d,f}(\Pi_{\Sigma_1}(\mathcal{T}), T_{\Sigma_1}) \), we can conclude that \( \tau \in \Lambda^+_{d,f}(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma}) \). However, \( \tau \not\in T_{\Sigma}(\neg C \sqcup D) \) since \( \tau_2 \not\in T_{\Sigma_2}(\neg C \sqcup D) \) and \( \tau_2 \not\in \tau \). Therefore, \( \Lambda^+_{d,f}(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma}) \not\subseteq T_{\Sigma}(\neg C \sqcup D) \), that is, \( \mathcal{T} \not\models_{d,f} \phi \). By Proposition 6, we can conclude that \( K \not\models_{d,f} \phi \).

2. If \( \phi \) is of the form \( C(a) \) where \( C \) is a concept and \( a \) is an individual name. Let \( \Sigma_1 = \text{Sig}(\mathcal{T}) \) and \( \Sigma_2 = \text{Sig}(\phi) \). Thus \( \Sigma_1 \cap \Sigma = \emptyset \) and \( \Sigma_1 \cup \Sigma_2 \subseteq \Sigma \). Let \( \tau_1 \) be a \( \Sigma_1 \)-type and \( \tau_1 \in \Lambda^+_{d,f}(\Pi_{\Sigma_1}(\mathcal{T}), T_{\Sigma_1}) \) and \( \tau_2 \not\in T_{\Sigma_2}(C) \). Because \( C(a) \) is neither a contradiction nor a tautology, \( \tau_2 \) exists. Let \( \mu = \mu_1 \cup \mu_2 \). By the proof of (1), if \( d \) is unbiased then for any aggregation \( f \), we can conclude that \( \mu \in \Lambda^+_{d,f}(\Pi_{\Sigma}(\mathcal{T}), T_{\Sigma}) \). However, \( \tau \) does not satisfy \( C \). Then
those features of the form \((\Xi, \mathcal{H})\) with \(\tau \in \Xi\) do not satisfy \(C\). Such a feature always exists since \(\tau_1\) is a arbitrary type. Therefore, \(K \not\models_{d,f} \phi\).

3. If \(\phi\) is of the form \(P(a, b)\) where \(P\) is a role name and \(a, b\) are individual names. Let \(\mathcal{F} = (\Xi, \mathcal{H})\) be a feature in \(\text{Mod}^F_{d,f}(K)\). Let \(\mathcal{F}'\) be a new feature obtained from \(\mathcal{F}\) by removing \(P(a, b)\) or \(P^-(b, a)\) in \(\mathcal{H}\). Since \(\text{Sig}(K) \cap \text{Sig}(P(a, b)) = \emptyset\), we can still conclude that \(\mathcal{F}' \in \text{Mod}^F_{d,f}(K)\) while \(\mathcal{F}'\) does not satisfy \(P(a, b)\). Therefore, \(K \not\models_{d,f} \phi\). We can analogously prove the form \(P^-(a, b)\).

Based on (1), (2) and (3), we conclude that \(K \not\models_{d,f} \phi\).

Proof of Proposition 9

Let \(\Sigma = \{A, a\}\). We can conclude that \(\{A(a)\} \models_{d,f} A(a)\) and \(\{\neg A(a)\} \models_{d,f} \neg A(a)\) by Proposition 2. However, we have both \(\{A(a), \neg A(a)\} \not\models_{d,f} A(a)\) and \(\{A(a), \neg A(a)\} \not\models_{d,f} \neg A(a)\) by Definition 8.

Proof of Proposition 10

If \(\phi\) is a tautology then we can directly conclude that \(K \models_{d,f} \phi\). Otherwise, \(\phi\) is a non-tautology. For each feature \(\mathcal{F} = (\Xi, \mathcal{H}) \in \text{Mod}^F_{d,f}(K)\), \(\Xi = \Xi' \cup \Xi''\) and \(\mathcal{H} = \mathcal{H}' \cup \mathcal{H}''\) for some feature \(\mathcal{F}' = (\Xi', \mathcal{H}') \in \text{Mod}^F_{d,f}(K')\) and some feature \(\mathcal{F}'' = (\Xi'', \mathcal{H}'') \in \text{Mod}^F_{d,f}(K'')\) by Definition 7 since \(K = K' \oplus K''\). Because \(K'\) is consistent, we have \(K' \models \phi\) if and only if \(K' \models_{d,f} \phi\) by Proposition 5. That is, \(\mathcal{F}' \in \text{Mod}^F_{d,f}(K') = \text{Mod}^F(K') \subseteq \text{Mod}^F(\{\phi\})\). Because \(\text{Sig}(\phi) \cap \text{Sig}(\Xi'') = \emptyset\), \(\mathcal{F} \in \text{Mod}^F(\{\phi\})\). Therefore, \(K \models_{d,f} \phi\).

Proof of Proposition 11

Let us prove that \(\models_{d,f}\) satisfies three properties: cautious reflexivity, cautious monotonicity, and, cautious cut.

- If \(K = K' \oplus K''\) and \(K'\) is consistent, then, for any distance function \(d\) and any monotonic hereditary aggregation function \(f\), \(K \models_{d,f} \varphi\) since \(K \models_{d,f} \varphi\) for all axiom \(\varphi \in K'\) by Proposition 10.

- If \(K \models_{d,f} \varphi\) then \(\text{Mod}^F_{d,f}(K) \subseteq \text{Mod}^F(\{\varphi\}) \subseteq \text{Mod}^F_{d,f}(\{\varphi\})\) by Definition 5 and Proposition 3. If \(K \models_{d,f} \psi\) then \(\text{Mod}^F_{d,f}(K) \subseteq \mathcal{F} \in \text{Mod}^F(\{\psi\})\) by Definition 8. \(\text{Mod}^F_{d,f}(K \cup \{\varphi\}) \subseteq \text{Mod}^F_{d,f}(K)\) since \(\text{Mod}^F_{d,f}(K) \cap \text{Mod}^F_{d,f}(\{\varphi\}) \neq \emptyset\) by Definition 7. Then \(\text{Mod}^F_{d,f}(K \cup \{\varphi\}) \subseteq \text{Mod}^F(\{\psi\})\), that is, \(K \cup \{\varphi\} \models_{d,f} \psi\).

- If \(K \models_{d,f} \varphi\) then \(\text{Mod}^F_{d,f}(K) \subseteq \text{Mod}^F(\{\varphi\}) \subseteq \text{Mod}^F_{d,f}(\{\varphi\})\) by Definition 8 and Proposition 3. If \(K \cup \{\varphi\} \models_{d,f} \psi\) then \(\text{Mod}^F_{d,f}(K \cup \{\varphi\}) \subseteq \text{Mod}^F(\{\psi\})\).
Mod^F(\{\psi\}) by Definition 8. Mod^F_{d,f}(K) = Mod^F_{d,f}(K) \cap Mod^F_{d,f}(\varphi) \subseteq Mod^F_{d,f}(K \cup \{\varphi\}) since Mod^F_{d,f}(K) \cap Mod^F_{d,f}(\varphi) \neq \emptyset by Definition 7. Then, Mod^F_{d,f}(K) \subseteq Mod^F(\{\psi\}) since Mod^F_{d,f}(K) \cap Mod^F_{d,f}(\varphi) \neq \emptyset by Definition 7. Therefore, if K \models_{d,f} \varphi then K \models_{d,f} \psi by Definition 8.

Based on (1), (2) and (3), we conclude that \models_{d,f} is cautious by the definition of cautious relation.

Proof of Proposition 12
For any KB K = (T, A), we have \Lambda^+_d(T, T) \subseteq \Lambda^+_d(T, T) by Definition 4. Thus we have Mod^F_{d,f}(K) \subseteq Mod^F_{d,f}(K) by Definition 7. Therefore, if K \models_{d,f} \phi then K \models_{d,f} \phi by Definition 8.

Proof of Proposition 13
For any KB K = (T, A), we have \Lambda^+_d(T, T) \subseteq \Lambda^+_d(T, T) for any \kappa_1, \kappa_2 (0 < \kappa_1 \leq \kappa_2 < 1) by Definition 5. Thus we have Mod^F_{d,f}(K) \subseteq Mod^F_{d,f}(K) by Definition 7. Therefore, if K \models_{d,f} \phi then K \models_{d,f} \phi by Definition 8.

Proof of Proposition 14
The first item directly follows Definition 9 and the second item holds since T \models_{d,f} \psi implies \Lambda_d(T, T) \subseteq T\Sigma(\psi). The third item holds since \psi is a contradiction implies T\Sigma(\psi) = \emptyset. The last item directly follows the fact \Lambda_d(T, T) - T\Sigma(\psi') = \Lambda_d(T, T) - T\Sigma(\psi'') if \psi', \psi'' are neither tautologies nor contradictions.

Proof of Proposition 15
Proof. It is clear that three properties Consistency, Monotony, and Dominance hold by following two facts:

- T is consistent if and only if MIV_{d,f}(T, \psi) = 0 for any \psi \in T, that is, MIV_{d,f}(T) = 0.
- MIV_{d,f}(T \cup T') \geq MIV_{d,f}(T).

To prove that MIV_{d,f} satisfies the property of Free Axiom Independence, we need the following claim.

Claim 1. Let T be a TBox and \psi an axiom. Let \Sigma = \text{Sig}(T \cup \{\psi\}) a signature. For d a distance and f an aggregation function, if \psi is a free axiom of T then T \models_{d,f} \psi.

Let us consider two cases.

- If \psi is a free axiom of T \cup \{\psi\} then T \models_{d,f} \psi, i.e., MIV_{d,f}(T, \psi) = 0 by definitions. Then MIV_{d,f}(T \cup \{\psi\}) = MIV_{d,f}(T);
• If \( \{ \psi \} \models \psi' \) and \( \{ \psi \} \not\models \top \sqsubseteq \bot \) then \( \text{MIV}_{d,f}(\mathcal{T} \cup \{ \psi \}) \geq \text{MIV}_{d,f}(\mathcal{T} \cup \{ \psi' \}) \).

Proof of Proposition 16

This proposition directly follows the fact that \( \text{Mod}^{F}_{d,f}(\mathcal{K}) = \{ \langle \Xi, \emptyset \rangle \mid \Xi \in \Lambda_{d,f}(\Pi(\mathcal{T}), \mathcal{T}_\Sigma) \} \) since \( \mathcal{A} = \emptyset \).

Proof of Proposition 17

This proposition directly follows the fact that \( \text{Mod}^{F}_{d,f}(\{ \psi \}) = \{ \langle \Xi, \mathcal{F} \rangle \mid \Xi \in \Lambda_{d,f}(\Pi(\mathcal{T}), \mathcal{T}_\Sigma) \} \) for any concept inclusion \( \psi \), that is, \( \text{Mod}^{F}_{d,f}(\mathcal{K}) - \text{Mod}^{F}_{d,f}(\{ \psi \}) = \{ \langle \Xi, \mathcal{F} \rangle \mid \Xi \not\in \Lambda_{d,f}(\Pi(\mathcal{T}), \mathcal{T}_\Sigma) \} \).

Proof of Proposition 18

This proposition directly follows Definition 12 and Proposition 17.

Proof of Proposition 19

It is clear that three properties Consistency, Monotony, and Dominance hold by following two facts:

• \( \mathcal{K} \) is consistent if and only if \( \text{MIV}_{d,f}(\mathcal{K}, \psi) = 0 \) for any \( \psi \in \mathcal{K} \), that is, \( \text{MIV}_{d,f}(\mathcal{K}) = 0 \).

• \( \text{MIV}_{d,f}(\mathcal{K} \cup \mathcal{K}') \geq \text{MIV}_{d,f}(\mathcal{K}) \).

To prove that \( \text{MIV}_{d,f} \) satisfies the property of Free Axiom Independence, we need the following claim.

Claim 2. Let \( \mathcal{K} \) be a KB and \( \psi \) an axiom. Let \( \Sigma = \text{Sig}(\mathcal{K} \cup \{ \psi \}) \) a signature. For \( d \) a distance and \( f \) an aggregation function, if \( \psi \) is a free axiom of \( \mathcal{K} \) then \( \mathcal{K} \models_{d,f} \psi \).

Let us consider two cases.

• If \( \psi \) is a free axiom of \( \mathcal{K} \cup \{ \psi \} \) then \( \mathcal{K} \models_{d,f} \psi \), i.e., \( \text{MIV}_{d,f}(\mathcal{K}, \psi) = 0 \) by definitions. Then \( \text{MIV}_{d,f}(\mathcal{K} \cup \{ \psi \}) = \text{MIV}_{d,f}(\mathcal{K}) \);

• If \( \{ \psi \} \models \psi' \) and \( \{ \psi \} \not\models \top \sqsubseteq \bot \) then \( \text{MIV}_{d,f}(\mathcal{K} \cup \{ \psi \}) \geq \text{MIV}_{d,f}(\mathcal{K} \cup \{ \psi' \}) \).

Proof of Proposition 20

It is clear that \( \text{MIV}_{d,f}(\mathcal{K}, \psi) \in [0, 1] \) for any \( \psi \in \mathcal{K} \) by Definition 11 and then we can conclude that \( \text{MIV}_{d,f}(\mathcal{K}) \in [0, 1] \) by Definition 12.