

Reasoning with Inconsistencies in Hybrid MKNF Knowledge Bases

Shasha Huang^a, Qingguo Li^{a,*}, Pascal Hitzler^b

^a*College of Mathematics and Econometrics, Hunan University, P. R. China*

^b*Kno.e.sis Center, Wright State University, Dayton, OH, USA*

Abstract

This paper is concerned with the handling of inconsistencies occurring in the combination of description logics and rules, especially in hybrid MKNF knowledge bases. More precisely, we present a paraconsistent semantics for hybrid MKNF knowledge bases (called para-MKNF knowledge bases) based on four-valued logic as proposed by Belnap. We also reduce this paraconsistent semantics to the stable model semantics via a linear transformation operator, which shows the relationship between the two semantics and indicates that the data complexity in our paradigm is not higher than that of classical reasoning. Moreover, we provide fixpoint operators to compute paraconsistent MKNF models, each suitable to different kinds of rules. At last we present the data complexity of instance checking in different para-MKNF knowledge bases.

Keywords: Knowledge representation, Description logics and rules, Non-monotonic reasoning, Paraconsistent reasoning, Data complexity

1. Introduction and Motivation

The Semantic Web [3, 15] extends the current World Wide Web by standards and techniques that help machines to understand the meaning of data on the web to enable more powerful intelligent system applications. The essence of the Semantic Web is to describe data on the web by metadata that

*corresponding author

Email addresses: `hss.ekaterina@gmail.com` (Shasha Huang),
`liqingguoli@yahoo.com.cn` (Qingguo Li), `pascal.hitzler@wright.edu` (Pascal Hitzler)

conveys the meaning—the semantics—of the data, and that is expressed by means of so-called ontologies, which are knowledge bases as studied in the field of Knowledge Representation and Reasoning.

The Web Ontology Language OWL [14] has been recommended by the World Wide Web Consortium¹ for representing ontologies. However, OWL is not as expressive as needed for modeling some real world problems. For example, it cannot model integrity constraints or closed-world reasoning that may be more suitable in some application scenarios. Consequently, how to improve OWL has become a very important branch of research in the Semantic Web field.

Knowledge representation approaches using rules in the sense of logic programming (LP), which is complementary to modeling in description logics (DLs, which underly OWL, see [15]) with respect to expressivity, have become a mature reasoning mechanism in the past thirty years. Thus combining rules and DLs is of continuous interest for the Semantic Web. However, significant differences between DLs and rules make the development of merged paradigms a hard problem. One of these differences is that the Open World Assumption (OWA) is employed in DLs, while the Closed World Assumption (CWA) is adhered to rules. Naive combinations of DLs and rules also lead to undecidable languages which are often deemed undesirable and should be avoided when developing DL-based paradigms.

A significant number of different approaches have been proposed for integrating DLs with rules. They can roughly be divided into two kinds: On the one hand, there are homogeneous approaches that unify DLs and LP in a special, unified, knowledge representation language. DLP [12], SWRL [16], ELP [25], nominal schemas [4, 5, 23, 24] and Hybrid MKNF knowledge bases [21, 33] are methods that belong to this kind of approach. On the other hand, there are hybrid approaches that view DLs and rules as independent parts, retaining their own reasoning mechanisms. AL-log [6], CARIN [26], HEX-programs [8], DL-programs [7] and DL+log [37] are all examples of this integration approach.

Among these approaches, hybrid MKNF knowledge bases, originally based on the stable model semantics [11], is one of the most mature integration methods. It has favorable properties of decidability, flexibility, faithfulness and tightness. A well-founded semantics [40] for such knowledge bases has

¹W3C, <http://www.w3.org/>

been proposed subsequently for better efficiency of reasoning [20, 21]. However, the integration of different knowledge bases can also easily lead to inconsistencies, even if both of the integrated knowledge bases are consistent if taken alone. Accordingly, reasoning systems based on the previous two semantics will break down. Therefore it is necessary to present a new semantics for hybrid MKNF knowledge bases to handle inconsistencies.

Traditionally there are two kinds of approaches to handle inconsistencies, one of which is recovering consistencies [39, 13] by repairing the knowledge base. But this approach may cause new problems, such as different results caused by different methods of recovering consistencies, inability of reusing information that has been eliminated, and so on. The other method admits inconsistencies and deals with them directly in a paraconsistent logic, and usually a four-valued logic [2, 28, 29, 30, 35, 38] is chosen for such purpose.

In this paper, we adopt the four-valued logic from [28, 29, 30], and present a paraconsistent semantics for hybrid MKNF knowledge bases. We will call the obtained paradigm para-MKNF knowledge bases. Our contribution can be summarized as follows:

- The paraconsistent MKNF model is faithful w.r.t. the four-valued model of the description logic \mathcal{ALC} from [28, 29, 30] and w.r.t. the paraconsistent stable model of extended disjunctive logic programs from [38].
- We present a transformation from para-MKNF knowledge bases to hybrid MKNF knowledge bases, which shows that our paraconsistent semantics is also faithful w.r.t. two-valued semantics for hybrid MKNF knowledge bases, and indicates that the data complexity of paraconsistent reasoning is not higher than that of standard MKNF reasoning.
- We define a fixpoint operator to compute paraconsistent MKNF models for para-MKNF knowledge bases in the positive case and stratified case, and provide a transformation from general MKNF rules to positive MKNF rules, such that the fixpoint operator can evaluate the paraconsistent MKNF models of para-MKNF knowledge bases in the general case.
- We discuss the data complexity of instance checking in different kinds of para-MKNF knowledge bases.

The remainder of the paper is organized as follows. In Section 2, we recall preliminaries on the four-valued Description Logic $\mathcal{ALC4}$ and on hybrid MKNF knowledge bases. In Section 3, we propose paraconsistent semantics for hybrid MKNF knowledge bases, and study its fundamental properties. In Section 4, we present a transformation from paraconsistent semantics to the stable model semantics of hybrid MKNF knowledge bases. In Section 5, we characterize the paraconsistent MKNF models via fixpoint operators, design procedures for computing the paraconsistent MKNF models and analyze the data complexity of different types of rules. In Section 6, we discuss related work. We conclude and discuss future work in Section 7.

This paper is a significantly extended and revised version of [17].

Throughout the paper, we make use of the following penguin example.

Example 1. Most bird species can fly, with some exceptions, such as penguins.

A bird ontology can structure and maintain the database. But it is not sufficient to correctly explain this statement by just building a bird ontology. In fact, an ontology specifies concepts, such as bird and penguin, and the relationships between them, such as every penguin is a bird. However, exceptions can not be represented correctly in the bird ontology. Thus one needs to employ other proper knowledge representation tools, such as non-monotonic rules.

2. Preliminaries

In this section, we introduce notions and notations used in the sequel. In detail, we present preliminaries for the four-valued description logic $\mathcal{ALC4}$, the logic of minimal knowledge and negation as failure (abbreviation MKNF) and hybrid MKNF knowledge bases.

2.1. The Four-valued Description Logic $\mathcal{ALC4}$

The basic idea of the four-valued description logic is to substitute four truth values for the two truth values used in classical logic: the four truth values are \mathbf{t} , \mathbf{f} , \top and \perp , representing *true*, *false*, *contradictory* (both *true* and *false*) and *unknown* (neither *true* nor *false*) respectively. With two partial orders \leq_k and \leq_t , which stand for a measure of the amount of information and a measure of truth, respectively, the set $IV = \{\mathbf{t}, \mathbf{f}, \top, \perp\}$

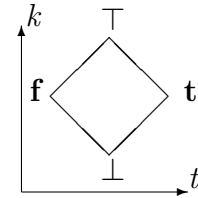


Figure 1: *FOUR*

	Syntax	Semantics
Concepts	A	$A^{\mathcal{I}} = \langle P, N \rangle$, where $P, N \subseteq \Delta^{\mathcal{I}}$
	\circ	$\circ^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
	$\ddot{\top}$	$\langle \Delta^{\mathcal{I}}, \emptyset \rangle$
	$\ddot{\perp}$	$\langle \emptyset, \Delta^{\mathcal{I}} \rangle$
	$\neg C$	$\langle N_1, P_1 \rangle$, if $C^{\mathcal{I}} = \langle P_1, N_1 \rangle$
	$C_1 \sqcap C_2$	$\langle P_1 \cap P_2, N_1 \cup N_2 \rangle$, if $C_i^{\mathcal{I}} = \langle P_i, N_i \rangle$, $i = 1, 2$
	$C_1 \sqcup C_2$	$\langle P_1 \cup P_2, N_1 \cap N_2 \rangle$, if $C_i^{\mathcal{I}} = \langle P_i, N_i \rangle$, $i = 1, 2$
	$\exists R.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \text{ and } y \in \text{proj}^+(C^{\mathcal{I}})\}$, $\{x \mid \forall y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \text{ implies } y \in \text{proj}^-(C^{\mathcal{I}})\}$
$\forall R.C$	$\{x \mid \forall y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \text{ implies } y \in \text{proj}^+(C^{\mathcal{I}})\}$, $\{x \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \text{ and } y \in \text{proj}^-(C^{\mathcal{I}})\}$	
Axioms	$C \mapsto D$	$\Delta^{\mathcal{I}} \setminus \text{proj}^-(C^{\mathcal{I}}) \subseteq \text{proj}^+(D^{\mathcal{I}})$
	$C \sqsubset D$	$\text{proj}^+(C^{\mathcal{I}}) \subseteq \text{proj}^+(D^{\mathcal{I}})$
	$C \twoheadrightarrow D$	$\text{proj}^+(C^{\mathcal{I}}) \subseteq \text{proj}^+(D^{\mathcal{I}})$, and $\text{proj}^-(D^{\mathcal{I}}) \subseteq \text{proj}^-(C^{\mathcal{I}})$
	$C(a)$	$a^{\mathcal{I}} \in \text{proj}^+(C^{\mathcal{I}})$
	$R(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Table 1: Four-valued Semantics for $\mathcal{ALC4}$

forms a bilattice \mathcal{FOUR} [2] as shown in Figure 1. We now recall the syntax and semantics of the four-valued description logic $\mathcal{ALC4}$ [28] that is based on \mathcal{FOUR} .

Syntactically, the elementary ingredients of $\mathcal{ALC4}$ are similar to the crisp \mathcal{ALC} , except for three kinds of class inclusions, which are called internal inclusion \sqsubset , material inclusion \mapsto and strong inclusion \twoheadrightarrow , corresponding to the three implication connectives in the four-valued logic case [2]. An $\mathcal{ALC4}$ knowledge base \mathcal{O} consists of axioms of the forms presented in Table 1, where $C(a)$ is called a concept assertion and $R(a, b)$ is called a role assertion.

Semantically, a four-valued interpretation $\mathcal{I} = (\cdot^{\mathcal{I}}, \Delta^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set (i.e., the domain of the interpretation) and $\cdot^{\mathcal{I}}$ is a function that assigns a distinct element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each individual a , a pair $\langle P, N \rangle$ of (not necessarily disjoint) subsets of $\Delta^{\mathcal{I}}$ to each concept A and a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role R . Intuitively, P is the set of elements that are known to belong to the extension of a concept C , while N is the

set of elements known to be not contained in the extension of concept C . Complex concepts are interpreted inductively as shown in Table 1, where $proj^+(\langle P, N \rangle) = P$ and $proj^-(\langle P, N \rangle) = N$. The semantics of the axioms in an ontology is formally defined in Table 1.

A four-valued interpretation \mathcal{I} is a *paraconsistent model* of an ontology \mathcal{O} iff it satisfies each axiom as shown in the bottom part of Table 1. An ontology \mathcal{O} is *paraconsistently satisfiable* iff there exists such a model. Furthermore, \mathcal{O} *paraconsistently entails* an axiom α , written $\mathcal{O} \models_4 \alpha$, iff \mathcal{I} satisfies α , for every paraconsistent model \mathcal{I} of \mathcal{O} .

Example 2. Consider the penguin scenario of Example 1. The following axioms can be built as a part of the bird ontology.

$$\text{Penguin} \sqsubset \text{Bird} \tag{1}$$

$$\text{Bird} \sqsubset \text{Fly} \tag{2}$$

$$\text{Penguin}(\text{Tweety}) \tag{3}$$

$$\neg \text{Fly}(\text{Tweety}) \tag{4}$$

Axiom (1) states that every penguin is a bird, axiom (2) expresses that every bird can fly. Axioms (3) and (4) refer to individual Tweety. Axiom (3) states that Tweety is a penguin, and axiom (4) expresses that Tweety cannot fly.

Let C be a concept in $\mathcal{ALC}4$ and $a \in \Delta^{\mathcal{I}}$ an instance. For the four-valued semantics defined above, the following indicates the correspondence between truth values from \mathcal{FOUR} and concept assertions:

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

In [28], a transformation from $\mathcal{ALC}4$ to \mathcal{ALC} was defined as follows (where A is an atomic concept, C , D and E are concepts and R is a role):

- If $C = A$, then $\lambda(C) = A_+$, where A_+ is a new concept;
- If $C = \neg A$, then $\lambda(C) = A_-$, where A_- is a new concept;
- If $C = \ddot{\top}$, then $\lambda(C) = \ddot{\top}$, where $\ddot{\top}$ is the top concept;
- If $C = \ddot{\perp}$, then $\lambda(C) = \ddot{\perp}$, where $\ddot{\perp}$ is the bottom concept;
- If $C = E \sqcap D$, then $\lambda(C) = \lambda(E) \sqcap \lambda(D)$;
- If $C = E \sqcup D$, then $\lambda(C) = \lambda(E) \sqcup \lambda(D)$;
- If $C = \forall R.D$, then $\lambda(C) = \forall R.\lambda(D)$;
- If $C = \exists R.D$, then $\lambda(C) = \exists R.\lambda(D)$;
- If $C = \neg\neg D$, then $\lambda(C) = \lambda(D)$;
- If $C = \neg(E \sqcap D)$, then $\lambda(C) = \lambda(\neg E) \sqcup \lambda(\neg D)$;
- If $C = \neg(E \sqcup D)$, then $\lambda(C) = \lambda(\neg E) \sqcap \lambda(\neg D)$;
- If $C = \neg(\forall R.D)$, then $\lambda(C) = \exists R.\lambda(\neg D)$;
- If $C = \neg(\exists R.D)$, then $\lambda(C) = \forall R.\lambda(\neg D)$.

An $\mathcal{ALC}4$ knowledge base \mathcal{O} is transformed to an \mathcal{ALC} knowledge base $\overline{\mathcal{O}}$ based on the above transformation λ :

$$\lambda(C \mapsto D) = \neg\lambda(\neg C) \sqsubseteq \lambda(D);$$

$$\lambda(C \sqsubseteq D) = \lambda(C) \sqsubseteq \lambda(D);$$

$$\lambda(C \twoheadrightarrow D) = \{\lambda(C) \sqsubseteq \lambda(D), \lambda(\neg D) \sqsubseteq \lambda(\neg C)\};$$

$$\lambda(C(a)) = \lambda(C)(a); \lambda(R(a)) = \lambda(R)(a),$$

where C and D are concepts, a is an individual, and R is a role.

Theorem 1 ([28]). $\mathcal{O} \models_4 \alpha$ if and only if $\overline{\mathcal{O}} \models_2 \lambda(\alpha)$ for an ontology \mathcal{O} , where \models_2 is the entailment in \mathcal{ALC} .

2.2. The Logic of Minimal Knowledge and Negation as Failure

The Logic of Minimal Knowledge and Negation as Failure (MKNF) [27] has been proposed as a unifying framework for different nonmonotonic formalisms, such as default logic, autoepistemic logic, and logic programming [33]. It is a variant of first-order modal logic with two modal operators: **K** and **not**.

Let Σ be a signature that consists of constants and function symbols, first-order predicates, and the binary equality predicate \approx . A first-order atom $P(t_1, \dots, t_l)$ is an MKNF formula, where P is a first-order predicate and t_i are first-order terms. Other MKNF formulae are built over Σ by using standard connectives in first-order logic and two modal operators as follows: **true**, $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\exists x : \varphi$, **K** φ , **not** φ . Moreover, formulae **false**, $\varphi_1 \vee \varphi_2$, $\varphi_1 \supset \varphi_2$, $\forall x : \varphi$, $\varphi_1 \equiv \varphi_2$, $t_1 \approx t_2$ and $t_1 \not\approx t_2$ are abbreviations, respectively, for formulae \neg **true**, $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\neg\varphi_1 \vee \varphi_2$, $\neg(\exists x : \neg\varphi)$, $(\varphi_1 \supset \varphi_2) \wedge (\varphi_2 \supset \varphi_1)$, $\approx(t_1, t_2)$, $\neg(t_1 \approx t_2)$. First-order atoms of the form $t_1 \approx t_2$ and $t_1 \not\approx t_2$ are called *equalities* and *inequalities*, respectively, and have a predefined interpretation [10]. Formulae of the form **K** φ (**not** φ) are called *modal K-atoms* (**not-atoms**); modal **K-atoms** and **not-atoms** are called *modal atoms*. An MKNF formula φ is called *closed* if it contains no free variables, and called *ground* if it is without any variables. An MKNF formula φ is called *modally closed* if it is closed and all modal operators are applied to closed subformulae. Formula $\varphi[t/x]$ is obtained from φ by substituting the term t for the variable x .

Let Δ be a *universe*, which contains an infinite supply of constants, besides constants occurring in the formulae. Just like in first-order logic, a first-order interpretation I over Σ and Δ assigns an object $a^I \in \Delta$ to each constant $a \in \Sigma$, a function $f^I : \Delta^n \rightarrow \Delta$ to each function $f \in \Sigma$ and a relation $P^I \subseteq \Delta^n$ to each predicate $P \in \Sigma$. Moreover, it interprets \approx as equality predicate, that is to say, $\approx(t_1, t_2)$ iff $t_1 = t_2$. Unlike in first-order logic, for each element $\alpha \in \Delta$, the signature Σ is required to contain a special constant n_α , called a *name*, such that $n_\alpha^I = \alpha$.

An *MKNF structure* is a triple (I, M, N) , where I is a first-order interpretation over Σ and Δ , while M and N are nonempty sets of first-order interpretations. *Satisfaction* of a closed MKNF formula φ is defined inductively as follows:

$$\begin{array}{ll}
(I, M, N) \models \mathbf{true} & \text{for each structure } (I, M, N) \\
(I, M, N) \models P(t_1, \dots, t_l) & \text{iff } P(t_1, \dots, t_l) \in I \\
(I, M, N) \models \neg\varphi & \text{iff } (I, M, N) \not\models \varphi \\
(I, M, N) \models \varphi_1 \wedge \varphi_2 & \text{iff } (I, M, N) \models \varphi_1 \text{ and } (I, M, N) \models \varphi_2
\end{array}$$

$$\begin{aligned}
(I, M, N) \models \exists x : \varphi & \quad \text{iff } (I, M, N) \models \varphi[n_\alpha/x] \text{ for some } \alpha \in \Delta \\
(I, M, N) \models \mathbf{K}\varphi & \quad \text{iff } (J, M, N) \models \varphi \text{ for all } J \in M \\
(I, M, N) \models \mathbf{not}\varphi & \quad \text{iff } (J, M, N) \not\models \varphi \text{ for some } J \in N
\end{aligned}$$

An *MKNF interpretation* M over a universe Δ is a nonempty set of first-order interpretations. M *satisfies* a closed MKNF formula φ , written $M \models \varphi$, iff $(I, M, M) \models \varphi$ for every $I \in M$. M is an MKNF *model* of φ if: (i) $(I, M, M) \models \varphi$ for each $I \in M$; (ii) for each set of first-order interpretations M' such that $M' \supsetneq M$, we have $(I', M', M) \not\models \varphi$ for some $I' \in M'$. An MKNF formula φ is *MKNF satisfiable* if an MKNF model of φ exists; otherwise φ is *MKNF unsatisfiable*. Furthermore, $\varphi \models_{\text{MKNF}} \psi$ if and only if $M \models \psi$ for each MKNF model M of φ .

Note that the definition of MKNF model indicates the preference of the maximal set M that satisfies φ . The bigger the MKNF model is, the less knowledge we get from the knowledge base. In fact, if $M_1 \subseteq M_2$, then M_1 satisfies φ if M_2 satisfies φ .

Example 3. Consider the MKNF formula $\varphi = \mathbf{not} q \supset \mathbf{K}p$, where p and q are two propositional atoms. $M = \{\{p\}, \{p, q\}\}$ is the MKNF model of φ .

As argued in [33], the MKNF semantics has two undesirable properties. One is counterintuitive semantics caused by an arbitrary universe, and the other one is different constants in different interpretations, which leads to the counterintuitive semantics of existential quantification. Therefore, the standard name assumption was adopted in [33].

Definition 1. (*Standard Name Assumption [33]*) A first-order interpretation I over a signature Σ employs the standard name assumption if

- (1) the universe Δ of I contains all constants of Σ and a countably infinite number of additional constants called parameters;
- (2) $t^I = t$ for each ground term t constructed using the function symbols from Σ and the constants from Δ ; and
- (3) the predicate \approx is interpreted in I as a congruence relation – that is, \approx is reflexive, symmetric, transitive, and allows for the replacement of equals by equals [10].

Therefore, Herbrand first-order interpretations, in which each constant is interpreted by itself, are used to replace the first-order interpretations in

MKNF interpretations. Moreover, it has been proved in [10] that each first-order formula is satisfiable iff it is satisfiable in a model that employs the standard name assumption.

2.3. Hybrid MKNF Knowledge Bases

Hybrid MKNF, based on the MKNF logic, is a mature approach for integrating Description Logics and Logic Programming, proposed by Boris Motik and Riccardo Rosati [32, 33]. Hybrid MKNF knowledge bases consist of a finite number of MKNF rules and a decidable description logic knowledge base which can be translated to first-order logic.

More concretely, the approach based on Hybrid MKNF knowledge bases is applicable to any first-order fragment \mathcal{DL} that satisfies these conditions: (i) each knowledge base $\mathcal{O} \in \mathcal{DL}$ can be translated to a formula $\pi(\mathcal{O})$ of function-free first-order logic with equality (see [1] for standard translation for Description Logic axioms), (ii) it supports ABox-assertions of the form $P(t_1, \dots, t_l)$, where P is a predicate and each t_i is a constant of \mathcal{DL} , and (iii) satisfiability checking and instance checking (i.e., checking entailments of the form $\mathcal{O} \models P(t_1, \dots, t_l)$) are decidable. Note that description logics around OWL satisfy these conditions. The restriction to function-free first-order logic guarantees the decidability of the language. Therefore, in the rest of the paper, we will not allow function symbols in hybrid MKNF knowledge bases.

Definition 2 ([33]). A formula ξ_G is a grounding of a nonground formula ξ if ξ_G is obtained from ξ by replacing its free variables with constants. A set of generalized atoms \mathcal{GA} is a set of first-order formulae such that, if $\xi \in \mathcal{GA}$, then $\xi_G \in \mathcal{GA}$ for each grounding ξ_G of ξ . A generalized atom is ground if it does not contain free variables.

Definition 3. Let \mathcal{O} be a DL knowledge base. An MKNF rule r has the following form, where $\alpha_i, \beta_i, \gamma_i$ are function-free generalized atoms:

$$\mathbf{K}\alpha_1 \vee \dots \vee \mathbf{K}\alpha_n \leftarrow \mathbf{K}\beta_{n+1} \wedge \dots \wedge \mathbf{K}\beta_m \wedge \mathbf{not}\gamma_{m+1} \wedge \dots \wedge \mathbf{not}\gamma_k \quad (5)$$

The sets $\{\mathbf{K}\alpha_i\}$, $\{\mathbf{K}\beta_i\}$, $\{\mathbf{not}\gamma_i\}$ are called the rule head, the positive body and the negative body, respectively. An MKNF rule r is nondisjunctive if $n = 1$; r is positive if $m = k$; r is a fact if $m = k = 0$. A program \mathcal{P} is a finite set of MKNF rules. A hybrid MKNF knowledge base \mathcal{K} is a pair $(\mathcal{O}, \mathcal{P})$.

Mapping Concepts to FOL
$\pi_y(\top, X) = \top, \pi_y(\perp, X) = \perp$ $\pi_y(A, X) = A(X), \pi_y(\neg C, X) = \neg \pi_y(C, X)$ $\pi_y(C_1 \sqcap C_2, X) = \pi_y(C_1, X) \wedge \pi_y(C_2, X)$ $\pi_y(C_1 \sqcup C_2, X) = \pi_y(C_1, X) \vee \pi_y(C_2, X)$ $\pi_y(\exists R.C, X) = \exists y : R(X, y) \wedge \pi_x(C, y)$ $\pi_y(\forall R.C, X) = \forall y : R(X, y) \rightarrow \pi_x(C, y)$
Mapping Axioms to FOL
$\pi(C(a)) = \pi_y(C, a), \pi(R(a, b)) = R(a, b)$ $\pi(C \sqsubseteq D) = \forall x : \pi_y(C, x) \rightarrow \pi_y(D, x), \pi(a \approx b) = a \approx b$
<p>where X is a meta variable and is substituted by actual variable, and π_x is defined as π_y by substituting x and x_i for y and y_i, respectively.</p>

Table 2: Translation of \mathcal{ALC} into FOL

To obtain a practically useful formalism, the language \mathcal{GA} should at least include the standard function-free first-order atoms of the form $P(t_1, \dots, t_l)$, negative literals of the form $\neg P(t_1, \dots, t_l)$, and conjunctive queries over \mathcal{DL} . In this section, we refer to these three types of generalized atoms.

Hybrid MKNF knowledge bases, as defined above, are not of the form of MKNF knowledge bases. In order to semantically interpret hybrid MKNF knowledge bases, the transformation π (see Table 2) that translates description logic expressions to first-order formulae is extended to MKNF rules and hybrid MKNF knowledge bases as follows:

Definition 4. Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid knowledge base. We extend π to r of form (5), \mathcal{P} , and \mathcal{K} as follows, where x is the vector of the free variables of MKNF rule r :

$$\pi(r) = \forall x : (\mathbf{K}\alpha_1 \vee \dots \vee \mathbf{K}\alpha_n \subset \mathbf{K}\beta_{n+1} \wedge \dots \wedge \mathbf{K}\beta_m \wedge \mathbf{not}\gamma_{m+1} \wedge \dots \wedge \mathbf{not}\gamma_k)$$

$$\pi(\mathcal{P}) = \bigwedge_{r \in \mathcal{P}} \pi(r) \quad \pi(\mathcal{K}) = \mathbf{K}\pi(\mathcal{O}) \wedge \pi(\mathcal{P})$$

\mathcal{K} is satisfiable if and only if an MKNF model of $\pi(\mathcal{K})$ exists, and \mathcal{K} entails a closed MKNF formula φ , written $\mathcal{K} \models \varphi$ if and only if $\pi(\mathcal{K}) \models_{MKNF} \varphi$.

To ensure that the MKNF logic is decidable, *DL-safety* is introduced as a restriction to MKNF rules.

Definition 5 ([33]). We assume that the signature Σ contains a subset $\Sigma_{DL} \subseteq \Sigma$ such that $\approx \in \Sigma_{DL}$. We call the predicates in Σ_{DL} DL-predicates and assume that \mathcal{DL} refers only to such predicates; furthermore, we call the predicates in $\Sigma \setminus \Sigma_{DL}$ non-DL-predicates.

A generalized atom ξ is a DL-atom if it contains only predicates from Σ_{DL} ; furthermore, ξ is a non-DL-atom if it is of the form $(\neg)P(t_1, \dots, t_n)$ for P a non-DL-predicate. A modal atom $\mathbf{K}\xi$ or $\mathbf{not}\xi$ is a DL-atom or non-DL-atom if ξ has the respective property.

An MKNF rule is DL-safe if each modal atom in it is either a DL- or a non-DL-atom, and if every variable in r occurs in the body of r in some non-DL-atom of the form $\mathbf{K}\beta$. A hybrid MKNF knowledge base \mathcal{K} is DL-safe if each rule $r \in \mathcal{P}$ is DL-safe.

In the rest of this paper, without explicitly stating it, hybrid MKNF knowledge bases are considered to be DL-safe.

Definition 6. Given a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. The ground instantiation of \mathcal{K} is the knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, where \mathcal{P}_G is obtained from \mathcal{P} by replacing each rule r of \mathcal{P} with a set of rules substituting each variable in r with constants from \mathcal{K} in all possible ways.

Grounding the knowledge base \mathcal{K} ensures that rules in \mathcal{P} apply only to objects that occur in \mathcal{K} . And it was shown in [33] that the MKNF models of \mathcal{K} and \mathcal{K}_G coincide.

Example 4. Consider a DL knowledge base \mathcal{O} consisting of axioms (1) and (4) from Example 2, and the set \mathcal{P} consisting of the rules (6) and (7) below. Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. Predicates from \mathcal{O} start with an uppercase letter and others with a lowercase letter.

$$\mathbf{K}Fly(x) \leftarrow \mathbf{K}animal(x), \mathbf{K}Bird(x), \mathbf{not}Penguin(x) \tag{6}$$

$$\mathbf{K}Penguin(Tweety) \leftarrow \tag{7}$$

The knowledge base \mathcal{K} describes correctly the statement in Example 1 – that is, most birds can fly, with some exceptions, such as penguins. The exception is expressed by the **not** operator, and closed world reasoning is used for Penguin in (6). Note that the non-DL-atom animal ensures the DL-safety of rule (6).

Hybrid MKNF knowledge bases provide a paradigm for representing data sources on the web using rules and description logics simultaneously. Local closed world reasoning in the knowledge bases bridges the rules and DLs, and accordingly overcomes the expressive limitation of rules and DLs, and enhances the expressivity.

However, real knowledge bases will be distributed and multi-authored. It is unreasonable to require every knowledge base to be logically consistent. Inconsistencies may arise when rules and DLs are reconciled in hybrid MKNF knowledge bases, even if the rule-part and DL-part are consistent if taken alone.

Example 5. Consider a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. The DL part \mathcal{O} consists of axioms (1) to (3) from Example 2, and $\mathcal{P} = \{\mathbf{K}\neg\text{Fly}(x) \leftarrow \mathbf{K}\text{Penguin}(x)\}$. Note that \mathcal{O} and \mathcal{P} are consistent knowledge bases, respectively. However, the combination causes the inconsistency of $\text{Fly}(\text{Tweety})$. Therefore \mathcal{K} has no MKNF model.

In Example 5, classical reasoning broke down due to the inconsistency of $\text{Fly}(\text{Tweety})$. Some useful information will be lost, e.g., $\text{Penguin}(\text{Tweety})$. In order to handle these problems, we will present a paraconsistent semantics for hybrid MKNF knowledge bases.

3. Para-MKNF Knowledge Bases with Paraconsistent Semantics

For distinguishing the two hybrid MKNF knowledge bases with stable model semantics and paraconsistent semantics, we call the latter para-MKNF knowledge base. Now we describe its syntax and semantics.

Syntactically, para-MKNF knowledge bases hardly differ from hybrid MKNF KBs, and similarly a para-MKNF KB \mathcal{K} has two components: \mathcal{O} and \mathcal{P} , where \mathcal{O} is an \mathcal{ALC} KB and \mathcal{P} is a nonempty set of MKNF rules that differ only slightly from the ones in Definition 3. In the previous section, we have mentioned that there are three kinds of implication connectives in four-valued logic. However, in our paper, we will employ inclusion implication [2], as recent research [28, 30] has shown that it has desirable properties which the other two lack.²

²In particular, as shown in [30], inclusion implication scales much better in implementations, and it also preserves polynomial time complexity for important DLs such as those underlying the Web Ontology Language OWL [31].

Definition 7. Let \mathcal{O} be a DL knowledge base. L is a literal if it is an atom P , or of the form $\neg P$, where P is an atom. An MKNF rule r has the following form, where H_i, A_i, B_i are first-order function-free literals:

$$\mathbf{K}H_1 \vee \dots \vee \mathbf{K}H_n \leftarrow \mathbf{K}A_{n+1} \wedge \dots \wedge \mathbf{K}A_m \wedge \mathbf{not}B_{m+1} \wedge \dots \wedge \mathbf{not}B_k \quad (8)$$

The sets $\{\mathbf{K}H_i\}$, $\{\mathbf{K}A_i\}$, $\{\mathbf{not}B_i\}$ are called the rule head, the positive body and the negative body, respectively. An MKNF rule r is nondisjunctive if $n = 1$; r is positive if $m = k$; r is a fact if $m = k = 0$. A program \mathcal{P} is a finite set of MKNF rules. A para-MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ is a pair of a DL knowledge base \mathcal{O} and a program \mathcal{P} .

Note that, for simplicity, we substitute literals for first-order formulae in MKNF rules. In our paradigm, negative literals have the same status as positive literals, and we consider a modified version of Herbrand first-order interpretations, namely the set of ground literals occurring in \mathcal{K} , and call them *paraconsistent Herbrand first-order interpretations*. Negative literals are allowed in paraconsistent Herbrand first-order interpretations, while Herbrand first-order interpretations are restricted to first-order atoms.

Definition 8. A four-valued (paraconsistent) MKNF structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ consists of a paraconsistent Herbrand first-order interpretation \mathcal{I} and two nonempty sets of paraconsistent Herbrand first-order interpretations \mathcal{M} and \mathcal{N} . A nonempty set of paraconsistent Herbrand first-order interpretations \mathcal{M} is called a paraconsistent MKNF interpretation.

\mathcal{I} is supposed to interpret first-order formulae, while \mathcal{M} and \mathcal{N} are used to evaluate modal \mathbf{K} -atoms and modal \mathbf{not} -atoms, respectively. Every MKNF formula is assigned to an element in the bilattice \mathcal{FOUR} .

Definition 9. Let $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ be a paraconsistent MKNF structure, and $\{\mathbf{t}, \mathbf{f}, \perp, \top\}$ with partial order \leq_k be the set of truth values. We evaluate MKNF formulae inductively as follows:

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(L(t_1, \dots, t_l)) = \begin{cases} \mathbf{t} & \text{iff } L(t_1, \dots, t_l) \in \mathcal{I} \text{ and } \neg L(t_1, \dots, t_l) \notin \mathcal{I} \\ \mathbf{f} & \text{iff } L(t_1, \dots, t_l) \notin \mathcal{I} \text{ and } \neg L(t_1, \dots, t_l) \in \mathcal{I} \\ \top & \text{iff } L(t_1, \dots, t_l) \in \mathcal{I} \text{ and } \neg L(t_1, \dots, t_l) \in \mathcal{I} \\ \perp & \text{iff } L(t_1, \dots, t_l) \notin \mathcal{I} \text{ and } \neg L(t_1, \dots, t_l) \notin \mathcal{I} \end{cases}$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\neg\varphi) = \begin{cases} \mathbf{t} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{f} \\ \mathbf{f} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t} \\ \top & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) = \top \\ \perp & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) = \perp \end{cases}$$

$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\mathbf{true}) = \top$, for each paraconsistent structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$

$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\mathbf{false}) = \perp$, for each paraconsistent structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1 \wedge \varphi_2) = (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1) \wedge (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_2)$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1 \vee \varphi_2) = (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1) \vee (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_2)$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\exists x : \varphi) = \bigvee_{\alpha \in \Delta} (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi[n_\alpha/x])$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\forall x : \varphi) = \bigwedge_{\alpha \in \Delta} (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi[n_\alpha/x])$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\mathbf{K}\varphi) = \bigwedge_{\mathcal{J} \in \mathcal{M}} (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi)$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1 \supset \varphi_2) = \begin{cases} \mathbf{t} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_1) \in \{\mathbf{f}, \perp\} \\ (\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi_2) & \text{otherwise} \end{cases}$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\mathbf{not}\varphi) = \begin{cases} \mathbf{t} & \text{iff } \exists \mathcal{J} \in \mathcal{N} \text{ s. t. } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{f} \\ & \text{and no other } \mathcal{J} \in \mathcal{N}, \text{ s. t. } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \perp \\ \mathbf{f} & \text{iff } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t} \text{ for all } \mathcal{J} \in \mathcal{N}; \\ & \text{or } \exists \mathcal{J} \in \mathcal{N} \text{ such that } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t}, \\ & \text{and for other } \mathcal{J} \in \mathcal{N}, (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \top \\ \top & \text{iff } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \perp \text{ for some } \mathcal{J} \in \mathcal{N} \\ \perp & \text{iff } (\mathcal{J}, \mathcal{M}, \mathcal{N})(\varphi) = \top \text{ for all } \mathcal{J} \in \mathcal{N} \end{cases}$$

Moreover, as defined in classical semantics, formulae $\varphi_1 \equiv \varphi_2$, $t_1 \approx t_2$ and $t_1 \not\approx t_2$ are abbreviations, respectively, for formulae $(\varphi_1 \supset \varphi_2) \wedge (\varphi_2 \supset \varphi_1)$, $\approx (t_1, t_2)$, $\neg(t_1 \approx t_2)$.

As in two-valued semantics, the evaluation of $\mathbf{not}\varphi$ basically follows the “mirror” idea of $\mathbf{K}\varphi$, except for the case for φ being evaluated in some to true and in some to false.

Definition 10. Let $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ be a paraconsistent MKNF structure. Paraconsistent satisfaction of closed MKNF formulae is defined inductively as follows:

$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \mathbf{true}$	for each paraconsistent structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 P(t_1, \dots, t_l)$	iff $P^{\mathcal{I}}(t_1, \dots, t_l) \in \{\mathbf{t}, \top\}$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \neg\varphi$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{f}, \top\}$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_1 \wedge \varphi_2$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_i, i = 1, 2$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_1 \vee \varphi_2$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_1$ or $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_2$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \exists x : \varphi$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi[n_\alpha/x]$ for some $\alpha \in \Delta$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \forall x : \varphi$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi[n_\alpha/x]$ every $\alpha \in \Delta$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_1 \supset \varphi_2$	iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \not\models_4 \varphi_1$ or $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \varphi_2$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \mathbf{K}\varphi$	iff $(\mathcal{J}, \mathcal{M}, \mathcal{N}) \models_4 \varphi$ for all $\mathcal{J} \in \mathcal{M}$
$(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models_4 \mathbf{not}\varphi$	iff $(\mathcal{J}, \mathcal{M}, \mathcal{N}) \not\models_4 \varphi$ for some $\mathcal{J} \in \mathcal{N}$

It can be easily verified that Definition 9 of paraconsistent semantics is compatible with Definition 10 of paraconsistent satisfaction, and we will work mostly with the latter. For a closed MKNF formula φ , a paraconsistent MKNF interpretation \mathcal{M} *paraconsistently satisfies* φ , written $\mathcal{M} \models_4 \varphi$, iff $(\mathcal{I}, \mathcal{M}, \mathcal{M}) \models_4 \varphi$ for each $\mathcal{I} \in \mathcal{M}$.

Definition 11. A paraconsistent MKNF interpretation \mathcal{M} is a paraconsistent MKNF model of a given closed MKNF formula φ , written $\mathcal{M} \models_{MKNF}^4 \varphi$ if and only if the following two conditions are satisfied :

- (1) \mathcal{M} paraconsistently satisfies φ ;
- (2) for each paraconsistent MKNF interpretation \mathcal{M}' , such that $\mathcal{M}' \supseteq \mathcal{M}$, we have $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \not\models_4 \varphi$, for some $\mathcal{I}' \in \mathcal{M}'$.

For a para-MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, \mathcal{K} is *paraconsistently MKNF satisfiable* iff a paraconsistent MKNF model of $\pi(\mathcal{K})$ exists, where $\pi(\mathcal{K})$ is defined as in Section 2.3. φ *para-MKNF entails* ϕ , written $\varphi \models_{MKNF}^4 \phi$, iff $\mathcal{M} \models_4 \phi$ for each paraconsistent MKNF model \mathcal{M} of φ .

When a para-MKNF knowledge base \mathcal{K} has consistent MKNF models, we distinguish these consistent MKNF models as *preferred MKNF models*. A consistent MKNF model \mathcal{M} means that for all $\mathbf{K}\xi$ such that $\mathcal{M} \models_4 \mathbf{K}\xi$, no $\mathbf{K}\xi$ and $\mathbf{K}\neg\xi$ hold simultaneously. A para-MKNF knowledge base \mathcal{K} is *consistent* if there exists a preferred MKNF model. The preferred MKNF model coincides with the MKNF model under the two-valued semantics.

Example 6. Consider the para-MKNF knowledge base \mathcal{K} in Example 5. \mathcal{K} has a paraconsistent MKNF model $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \{Penguin(Tweety), Fly(Tweety), \neg Fly(Tweety), Bird(Tweety)\}\}$.

Paraconsistent semantics in our paradigm is faithful. That is to say, the semantics yields the paraconsistent semantics for DLs according to [28, 29, 30] when no rules are present, and the p-stable model of LP from [38] when the DL-component is empty.

In order to show this property, we need to recall the notion of *p-stable model* of an answer set program.

An *extended disjunctive program* P is a finite set of clauses of the form

$$L_1 \vee \dots \vee L_l \leftarrow L_{l+1} \wedge \dots \wedge L_m \wedge notL_{m+1} \wedge \dots \wedge notL_n \quad (9)$$

where the L_i are literals, $1 \leq i \leq n$. P is called a *positive extended disjunctive program* if $m = n$.

Definition 12. (*Paraconsistent Semantics of ASP [38]*) Let P be an extended disjunctive program and I be a subset of the Herbrand base of P . The reduct of P w.r.t. I is the positive extended disjunctive program P^I such that a clause

$$L_1 \vee \dots \vee L_l \leftarrow L_{l+1} \wedge \dots \wedge L_m$$

is in P^I iff there is a ground clause of form (9) from P such that $\{L_{m+1}, \dots, L_n\} \cap I = \emptyset$. For a positive extended disjunctive program P , an interpretation I is a model of P if I satisfies every ground clause from P , and a p-minimal model if there exists no model J of P such that $J \subset I$.

Then I is called a *paraconsistent stable model* (shortly, p-stable model) of P if I is a p-minimal model of P^I .

Proposition 1. Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a para-MKNF knowledge base, φ a closed first-order formula, and A a ground literal.

- If $\mathcal{P} = \emptyset$, then $\mathcal{K} \models_{MKNF}^4 \varphi$ iff $\mathcal{O} \models_4 \varphi$.
- If $\mathcal{O} = \emptyset$, then $\mathcal{K} \models_{MKNF}^4 A$ iff $\mathcal{P} \models_4 A$, where $\mathcal{P} \models_4 A$ means for all the p-stable models I of \mathcal{P} , that $I \models_4 A$.

PROOF. Case 1: $\mathcal{P} = \emptyset$. (Necessity) \mathcal{K} consists only of DL axioms. Thus $\mathcal{K} \models_{MKNF}^4 \varphi$ equals to $\mathbf{K}\pi(\mathcal{O}) \models_{MKNF}^4 \varphi$, which means $\mathcal{M} \models_4 \varphi$ for each paraconsistent MKNF model \mathcal{M} of $\mathbf{K}\pi(\mathcal{O})$. Therefore, $\mathcal{I} \in \mathcal{M}$ iff \mathcal{I} is a paraconsistent model of $\pi(\mathcal{O})$. From $\mathcal{M} \models_4 \varphi$ we can infer $\pi(\mathcal{O}) \models_4 \varphi$, which is the first-order form of $\mathcal{O} \models_4 \varphi$.

(Sufficiency) Similarly, the sufficiency can easily be proved by contradiction.

Case 2: $\mathcal{O} = \emptyset$. \mathcal{K} consists only of MKNF rules. $\mathcal{K} \models_{MKNF}^4 \varphi$ if and only if $\pi(\mathcal{P}) \models_{MKNF}^4 \varphi$. We denote the program of the form (9) corresponding to \mathcal{P} by P . First we have a property (\dagger): \mathcal{M} is a paraconsistent MKNF model of \mathcal{K} iff I is a p-stable model of program P such that $I = \bigcap_{\mathcal{I} \in \mathcal{M}} \mathcal{I}$. Then $\pi(\mathcal{P}) \models_{MKNF}^4 \varphi$ infers that for every paraconsistent MKNF model \mathcal{M} of $\pi(\mathcal{P})$, $\mathcal{M} \models_4 \varphi$. Then we have $I \models_4 \varphi$, where $I = \bigcap_{\mathcal{I} \in \mathcal{M}} \mathcal{I}$. From (\dagger), we can infer that the p-stable model I of program P is one-to-one corresponding to the paraconsistent MKNF model \mathcal{M} of \mathcal{P} . Then we obtain that $P \models_4 \varphi$. The converse can be proved similarly.

Proof of (\dagger): (Necessity) Let \mathcal{M} be a paraconsistent MKNF model of \mathcal{K} and $I = \bigcap_{\mathcal{I} \in \mathcal{M}} \mathcal{I}$. For every rule r in P^I of form (9) such that $\{L_{m+1}, \dots, L_n\} \cap I = \emptyset$, if $\{L_{l+1}, \dots, L_m\} \subseteq I$, then $\{L_{l+1}, \dots, L_m\} \subseteq \bigcap_{\mathcal{I} \in \mathcal{M}} \mathcal{I}$, and then $\mathcal{M} \models_4 \mathbf{K}L_i$, for every i , $l+1 \leq i \leq m$. Since \mathcal{M} is the paraconsistent MKNF model of \mathcal{P} , we can infer that there exists L_j , $1 \leq j \leq l$, such that $\mathcal{M} \models_4 \mathbf{K}L_j$, then we obtain that $L_j \in I$ and I satisfies every rule in P^I . Next we prove the minimality of I . Suppose there exists an interpretation J , such that $J \subset I$ and J satisfies P^I . Let $\mathcal{M}' = \{\mathcal{J}' \mid \mathcal{J}' \supseteq J\}$, then $\mathcal{M} \subset \mathcal{M}'$. For every MKNF rule r of form (8), if $\mathcal{M}' \models_4 \mathbf{not}L_j$, $m+1 \leq j \leq k$, and $\mathcal{M}' \models_4 \mathbf{K}L_i$, $l+1 \leq i \leq m$, then $L_j \notin J$, for every $m+1 \leq j \leq k$, and $L_i \in J$, for every $l+1 \leq i \leq m$. Since J satisfies P^I , we obtain that there exists L_i , $1 \leq i \leq l$, such that $L_i \in J$. Thus $\mathcal{M}' \models_4 \mathbf{K}L_i$, and $\mathcal{M}' \models_4 \mathcal{P}$, which contradicts the fact that \mathcal{M} is the paraconsistent MKNF model of \mathcal{K} . Hence I is a p-stable model of P . (Sufficiency) Sufficiency can be proved similarly. We omit the details. \square

For a para-MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, let \mathcal{P}_G be the set of rules obtained from \mathcal{P} by replacing in each rule all variables with all constants from \mathcal{K} in all possible ways; the knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ is called the *ground instantiation* of \mathcal{K} .

As argued in [33], for a DL-safe hybrid knowledge base \mathcal{K} , the two-valued MKNF models of \mathcal{K}_G and \mathcal{K} coincide. In our paradigm, the same conclusion holds for paraconsistent MKNF models.

4. Transformation from Para-MKNF Knowledge Bases to Hybrid MKNF Knowledge Bases

Y. Ma et al. [28] have proposed a transformation λ from $\mathcal{ALC4}$ to \mathcal{ALC} . In this section, we extend λ to MKNF rules and then transform para-MKNF knowledge bases to hybrid MKNF knowledge bases based on the transformation operator.

Given a para-MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, the transformation operator λ transforms every axiom in \mathcal{O} as presented in Section 2.1. In this paper, MKNF rules are made up from $\mathbf{K}A_i$, $\mathbf{not}B_j$ and some connectors, where A_i and B_j are literals. We first define the transformation on modal atoms.

- If $\varphi = \mathbf{K}\psi$, then $\lambda(\varphi) = \mathbf{K}\lambda(\psi)$, where ψ is a literal;
- If $\varphi = \mathbf{not}\psi$, then $\lambda(\varphi) = \mathbf{not}\lambda(\psi)$, where ψ is a literal.

Based on this, MKNF rules of form (8) are transformed to $\lambda(\mathbf{K}H_1) \vee \dots \vee \lambda(\mathbf{K}H_n) \leftarrow \lambda(\mathbf{K}A_{n+1}) \wedge \dots \wedge \lambda(\mathbf{K}A_m) \wedge \lambda(\mathbf{not}B_{m+1}) \wedge \dots \wedge \lambda(\mathbf{not}B_k)$. Then the para-MKNF knowledge base \mathcal{K} is transformed inductively to a hybrid MKNF knowledge base, denoted by $\bar{\mathcal{K}}$. We say $\bar{\mathcal{K}}$ is *classically induced* by the para-MKNF knowledge base \mathcal{K} , if all the axioms and rules in $\bar{\mathcal{K}}$ are exactly the transformations of the axioms and rules in \mathcal{K} . We also extend λ to MKNF formulae such that modal operators \mathbf{K} and \mathbf{not} are restricted to literals.

- If $\varphi = P(t_1, \dots, t_l)$, then $\lambda(\varphi) = P_+(t_1, \dots, t_l)$, where $P(t_1, \dots, t_l)$ is a first-order atom occurring in \mathcal{K} and $P_+(t_1, \dots, t_l)$ is a new first-order atom;
- If $\varphi = \neg P(t_1, \dots, t_l)$, then $\lambda(\varphi) = P_-(t_1, \dots, t_l)$, where $P_-(t_1, \dots, t_l)$ is a new first-order atom;
- If $\varphi = \varphi_1 \wedge \varphi_2$, then $\lambda(\varphi) = \lambda(\varphi_1) \wedge \lambda(\varphi_2)$, where φ_1 and φ_2 are two MKNF formulae;
- If $\varphi = \varphi_1 \vee \varphi_2$, then $\lambda(\varphi) = \lambda(\varphi_1) \vee \lambda(\varphi_2)$;
- If $\varphi = \exists x : \psi$, then $\lambda(\varphi) = \exists x : \lambda(\psi)$;
- If $\varphi = \forall x : \psi$, then $\lambda(\varphi) = \forall x : \lambda(\psi)$;

- If $\varphi = \neg\neg\psi$, then $\lambda(\varphi) = \lambda(\psi)$;
- If $\varphi = \neg(\varphi_1 \wedge \varphi_2)$, then $\lambda(\varphi) = \lambda(\neg\varphi_1) \vee \lambda(\neg\varphi_2)$;
- If $\varphi = \neg(\varphi_1 \vee \varphi_2)$, then $\lambda(\varphi) = \lambda(\neg\varphi_1) \wedge \lambda(\neg\varphi_2)$;
- If $\varphi = \neg(\exists x : \psi)$, then $\lambda(\varphi) = \forall x : \lambda(\neg\psi)$;
- If $\varphi = \neg(\forall x : \psi)$, then $\lambda(\varphi) = \exists x : \lambda(\neg\psi)$;
- If $\varphi = \neg\mathbf{K}\psi$, then $\lambda(\varphi) = \mathbf{K}\lambda(\neg\psi)$, where ψ is a literal;
- If $\varphi = \neg\mathbf{not}\psi$, then $\lambda(\varphi) = \mathbf{not}\lambda(\neg\psi)$, where ψ is a literal.

An MKNF interpretation of $\overline{\mathcal{K}}$ can be induced by a paraconsistent interpretation of \mathcal{K} . First of all, we define the interpretation structure of $\overline{\mathcal{K}}$.

Definition 13 (Classical Induced MKNF Structure). *Let $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ be a paraconsistent MKNF structure of a para-MKNF knowledge base \mathcal{K} , and $\overline{\mathcal{K}}$ be the classical induced knowledge base of \mathcal{K} . The classical induced MKNF structure of $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, written $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})$, is defined as follows:*

1. $\overline{\Delta} = \Delta$, where $\overline{\Delta}$ represents the universe of the classical induced interpretation $\overline{\mathcal{I}}$;
2. for a first-order atom $P(t_1, \dots, t_l)$,

$$P_+^{\overline{\mathcal{I}}}(t_1, \dots, t_l) = \begin{cases} \mathbf{t} & \text{iff } P^{\mathcal{I}}(t_1, \dots, t_l) \in \{\mathbf{t}, \top\} \\ \mathbf{f} & \text{iff } P^{\mathcal{I}}(t_1, \dots, t_l) \in \{\mathbf{f}, \perp\} \end{cases} \quad (10)$$

$$P_-^{\overline{\mathcal{I}}}(t_1, \dots, t_l) = \begin{cases} \mathbf{t} & \text{iff } \neg P^{\mathcal{I}}(t_1, \dots, t_l) \in \{\mathbf{t}, \top\} \\ \mathbf{f} & \text{iff } \neg P^{\mathcal{I}}(t_1, \dots, t_l) \in \{\mathbf{f}, \perp\} \end{cases} \quad (11)$$

and we call $\overline{\mathcal{I}}$ a classical induced interpretation;

3. $\overline{\mathcal{M}}$ is a nonempty set corresponding to \mathcal{M} and consists of classical induced interpretations $\overline{\mathcal{I}}$ such that $\mathcal{I} \in \mathcal{M}$. $\overline{\mathcal{N}}$ is a nonempty set corresponding to \mathcal{N} and consists of classical induced interpretations $\overline{\mathcal{I}}$ such that $\mathcal{I} \in \mathcal{N}$.

Conversely, given an MKNF structure of a hybrid MKNF knowledge base $\overline{\mathcal{K}}$, we can define the four-valued induced MKNF structure of $\overline{\mathcal{K}}$ easily. Particularly, when $\overline{\mathcal{K}}$ is consistent, the four-valued induced MKNF structure coincides with the original MKNF structure.

Definition 14. Given a hybrid MKNF knowledge base \mathcal{K} , let $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})$ be a MKNF structure of a hybrid MKNF knowledge base $\overline{\mathcal{K}}$. The four-valued induced MKNF structure of $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})$, written $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, is defined as follows:

1. $\Delta = \overline{\Delta}$, where Δ represents the universe of the four-valued induced interpretation \mathcal{I} ;
2. for a first-order atom $P(t_1, \dots, t_l)$,

$$P^{\mathcal{I}}(t_1, \dots, t_l) = \begin{cases} \mathbf{t} & \text{iff } P_+(t_1, \dots, t_l) \in \overline{\mathcal{I}} \text{ and } P_-(t_1, \dots, t_l) \notin \overline{\mathcal{I}} \\ \mathbf{f} & \text{iff } P_+(t_1, \dots, t_l) \notin \overline{\mathcal{I}} \text{ and } P_-(t_1, \dots, t_l) \in \overline{\mathcal{I}} \\ \top & \text{iff } P_+(t_1, \dots, t_l) \in \overline{\mathcal{I}} \text{ and } P_-(t_1, \dots, t_l) \in \overline{\mathcal{I}} \\ \perp & \text{iff } P_+(t_1, \dots, t_l) \notin \overline{\mathcal{I}} \text{ and } P_-(t_1, \dots, t_l) \notin \overline{\mathcal{I}} \end{cases}$$

and we call \mathcal{I} a four-valued induced interpretation;

3. \mathcal{M} is a nonempty set corresponding to $\overline{\mathcal{M}}$ and consists of four-valued induced interpretations \mathcal{I} such that $\overline{\mathcal{I}} \in \overline{\mathcal{M}}$. \mathcal{N} is a nonempty set corresponding to $\overline{\mathcal{N}}$ and consists of four-valued induced interpretations \mathcal{I} such that $\overline{\mathcal{I}} \in \overline{\mathcal{N}}$.

Lemma 1. For a paraconsistent MKNF structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ of a para-MKNF knowledge base \mathcal{K} and any MKNF formula ϕ such that modal operators \mathbf{K} and \mathbf{not} are restricted to literals, we have

$$(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\phi) = \begin{cases} \mathbf{t} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})\phi \in \{\mathbf{t}, \top\} \\ \mathbf{f} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})\phi \in \{\mathbf{f}, \perp\} \end{cases} \quad (12)$$

$$(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\neg\phi) = \begin{cases} \mathbf{t} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})\neg\phi \in \{\mathbf{t}, \top\} \\ \mathbf{f} & \text{iff } (\mathcal{I}, \mathcal{M}, \mathcal{N})\neg\phi \in \{\mathbf{f}, \perp\} \end{cases} \quad (13)$$

PROOF. Let φ be a literal. From Definition 13, it is easy to verify that φ satisfies the equations (12) and (13). First of all, we prove the equations (12) and (13) for $\mathbf{K}\varphi$ and $\mathbf{not}\varphi$ inductively.

Case 1: $\phi = \mathbf{K}\varphi$.

Equation (12): $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\mathbf{K}\varphi) = \mathbf{t}$ iff $(\overline{\mathcal{J}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\varphi) = \mathbf{t}$ for every $\overline{\mathcal{J}} \in \overline{\mathcal{M}}$. From the assumption, $(\overline{\mathcal{J}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\varphi) = \mathbf{t}$ for every $\overline{\mathcal{J}} \in \overline{\mathcal{M}}$, iff $(\mathcal{J}, \mathcal{M}, \mathcal{N})\varphi \in \{\mathbf{t}, \top\}$ for every $\mathcal{J} \in \mathcal{M}$, which means that $(\mathcal{I}, \mathcal{M}, \mathcal{N})\mathbf{K}\varphi \in \{\mathbf{t}, \top\}$. Then we obtain that $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}})\lambda(\mathbf{K}\varphi) = \mathbf{t}$ iff $(\mathcal{I}, \mathcal{M}, \mathcal{N})\mathbf{K}\varphi \in \{\mathbf{t}, \top\}$.

Equation (13): $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\neg \mathbf{K}\varphi) = \mathbf{t}$ or \top iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{K}\varphi) = \mathbf{f}$ or \top . If $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{K}\varphi) = \mathbf{f}$, then there exists an interpretation $\mathcal{J} \in \mathcal{M}$ such that $\varphi^{\mathcal{J}} = \mathbf{f}$ and no $\mathcal{J}' \in \mathcal{M}$ such that $\varphi^{\mathcal{J}'} = \perp$ or \mathbf{t} . Then $\neg\varphi^{\mathcal{J}} = \mathbf{t}$ or \top for each $\mathcal{J} \in \mathcal{M}$ and hence $\lambda(\neg\varphi)^{\overline{\mathcal{J}}} = \mathbf{t}$ for each $\overline{\mathcal{J}} \in \overline{\mathcal{M}}$. Therefore, $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \lambda(\neg \mathbf{K}\varphi) = \mathbf{t}$. If $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{K}\varphi) = \top$, then for each $\mathcal{J} \in \mathcal{M}$ such that $\varphi^{\mathcal{J}} = \top$, and hence for each $\mathcal{J} \in \mathcal{M}$, $\neg\varphi^{\mathcal{J}} = \top$. Thus for each $\overline{\mathcal{J}} \in \overline{\mathcal{M}}$, $\lambda(\neg\varphi)^{\overline{\mathcal{J}}} = \mathbf{t}$. Therefore, $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \lambda(\neg \mathbf{K}\varphi) = \mathbf{t}$. The other side can be proved similarly. We omit the details here.

Case 2: $\phi = \mathbf{not}\varphi$.

Equation (12): $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \lambda(\mathbf{not}\varphi) = \mathbf{t}$ iff $(\overline{\mathcal{J}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \lambda(\varphi) = \mathbf{f}$ for some $\overline{\mathcal{J}} \in \overline{\mathcal{M}}$. From the assumption, $(\mathcal{J}, \mathcal{M}, \mathcal{N}) \varphi \in \{\mathbf{f}, \perp\}$ for some $\mathcal{J} \in \mathcal{M}$, which means that $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \mathbf{not}\varphi \in \{\mathbf{t}, \top\}$ from Definition 9. Therefore we obtain that $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \lambda(\mathbf{not}\varphi) = \mathbf{t}$ iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \mathbf{not}\varphi \in \{\mathbf{t}, \top\}$.

Equation (13): $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\neg \mathbf{not}\varphi) = \mathbf{t}$ or \top iff $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{not}\varphi) = \mathbf{f}$ or \top . If $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{not}\varphi) = \mathbf{f}$, then there exists at least one $\mathcal{J} \in \mathcal{N}$ such that $\varphi^{\mathcal{J}} = \mathbf{t}$ and for other $\mathcal{J}' \in \mathcal{M}$, $\varphi^{\mathcal{J}'} = \top$. Then there exists $\mathcal{J} \in \mathcal{N}$ such that $\neg\varphi^{\mathcal{J}} = \mathbf{f}$ and hence there exists $\overline{\mathcal{J}} \in \overline{\mathcal{N}}$ such that $\lambda(\neg\varphi)^{\overline{\mathcal{J}}} = \mathbf{f}$. Therefore $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \mathbf{not}\lambda(\neg\varphi) = \mathbf{t}$. If $(\mathcal{I}, \mathcal{M}, \mathcal{N}) (\mathbf{not}\varphi) = \top$, then there exists a $\mathcal{J} \in \mathcal{N}$ such that $\varphi^{\mathcal{J}} = \perp$. Then there exists $\mathcal{J} \in \mathcal{N}$ such that $\neg\varphi^{\mathcal{J}} = \perp$, and hence there exists $\overline{\mathcal{J}} \in \overline{\mathcal{N}}$ such that $\lambda(\neg\varphi)^{\overline{\mathcal{J}}} = \mathbf{f}$. Therefore, $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{N}}) \mathbf{not}\lambda(\neg\varphi) = \mathbf{t}$. Similarly, the other side can be proved. We omit the details.

It can be easily verified that equations (12) and (13) hold for MKNF formulae of the other forms. We omit the proofs here. \square

From Lemma 1, we can get an important conclusion as follows:

Theorem 2. *For a para-MKNF knowledge base \mathcal{K} and a closed MKNF formula φ , we have $\mathcal{K} \models_{MKNF}^4 \varphi$ iff $\overline{\mathcal{K}} \models_{MKNF} \lambda(\varphi)$.*

PROOF. (Necessity) Suppose \mathcal{M} is a paraconsistent MKNF model of \mathcal{K} , we only need to prove that the classical induced interpretation $\overline{\mathcal{M}}$ of \mathcal{M} is an MKNF model of $\overline{\mathcal{K}}$. $\mathcal{M} \models_{MKNF}^4 \pi(\mathbf{K})$ implies $\mathcal{M} \models_{MKNF}^4 \pi(\mathcal{P})$ and $\mathcal{M} \models_{MKNF}^4 \mathbf{K}\pi(\mathcal{O})$.

Let α be an axiom in \mathcal{O} . $\mathcal{M} \models_{MKNF}^4 \mathbf{K}\pi(\mathcal{O})$ implies $\mathcal{I} \models_4 \pi(\alpha)$ for each $\mathcal{I} \in \mathcal{M}$. Then $\overline{\mathcal{I}} \models_2 \pi(\lambda(\alpha))$ by Theorem 1, and hence $\overline{\mathcal{M}} \models \mathbf{K}\pi(\lambda(\alpha))$. Therefore, $\overline{\mathcal{M}} \models \overline{\mathcal{O}}$.

For MKNF rules of form (8), $(\mathcal{I}, \mathcal{M}, \mathcal{M}) \models_4 r$ iff $(\mathcal{I}, \mathcal{M}, \mathcal{M}) \models_4 \mathbf{K}H_i$, for some $1 \leq i \leq n$, or $(\mathcal{I}, \mathcal{M}, \mathcal{M}) \not\models_4 \mathbf{K}A_j$, for some $n + 1 \leq j \leq m$, or

$(\mathcal{I}, \mathcal{M}, \mathcal{M}) \models_4 \mathbf{not}B_t$, for some $m + 1 \leq t \leq k$, each case corresponding to the case $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{M}}) \models \lambda(\mathbf{K}H_i)$, $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{M}}) \not\models \lambda(\mathbf{K}A_j)$, or $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{M}}) \not\models \lambda(\mathbf{not}B_t)$, respectively. Each of the cases means $(\overline{\mathcal{I}}, \overline{\mathcal{M}}, \overline{\mathcal{M}}) \models \lambda(r)$.

If there exists an MKNF interpretation $\overline{\mathcal{M}'} \supseteq \overline{\mathcal{M}}$ such that $(\overline{\mathcal{I}'}, \overline{\mathcal{M}'}, \overline{\mathcal{M}'}) \models \lambda(r)$, then this MKNF interpretation structure must satisfy all the induced axioms and rules, which contradicts the maximality of $\overline{\mathcal{M}}$.

(Sufficiency) For any MKNF interpretation $\overline{\mathcal{M}}$ of $\overline{\mathcal{K}}$, let \mathcal{M} be the four-valued semantics of $\overline{\mathcal{M}}$. Similarly, we can prove that \mathcal{M} is an MKNF model of \mathcal{K} if $\overline{\mathcal{M}}$ is an MKNF model of $\overline{\mathcal{K}}$. We omit the details here. \square

Note that the transformation operator is linear. Thus from Theorem 2, we can conclude that the data complexity of our paradigm is not higher than that of classical reasoning³.

5. Characterization of Paraconsistent MKNF Models

In this section we present several fixpoint characterizations of paraconsistent MKNF models, each suitable to different kinds of rules. According to the discussion in Section 3, a para-MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ has exactly the same paraconsistent MKNF models as \mathcal{K}_G . Therefore in the rest of the paper, we only consider grounded knowledge bases \mathcal{K}_G .

Definition 15. *Let \mathcal{K}_G be a ground para-MKNF knowledge base. The set of \mathbf{K} -atoms of \mathcal{K}_G , written $\mathbf{KA}(\mathcal{K}_G)$, is the smallest set that contains (1) all ground \mathbf{K} -atoms occurring in \mathcal{P}_G , and (2) a modal atom $\mathbf{K}\xi$ for each ground modal atom $\mathbf{not}\xi$ occurring in \mathcal{P}_G . Furthermore, $\mathbf{HA}(\mathcal{K}_G)$ is the subset of $\mathbf{KA}(\mathcal{K}_G)$ that contains all \mathbf{K} -atoms occurring in the head of some rule in \mathcal{P}_G .*

As argued in [33], MKNF models of \mathcal{K}_G are determined by subsets of $\mathbf{HA}(\mathcal{K}_G)$. The same holds for paraconsistent MKNF models.

Example 7. *We consider the following para-MKNF knowledge base $\mathcal{K}_G^{ex} = (\mathcal{O}^{ex}, \mathcal{P}_G^{ex})$, where \mathcal{O}^{ex} consists of axiom (14) and \mathcal{P}_G^{ex} consists of MKNF rules (15)-(17):*

$$q \supset r \tag{14}$$

³Data complexity of reasoning in MKNF KBs estimates the performance of reasoning algorithms measured in terms of ABox assertions and MKNF rules.

$$\mathbf{K}q \leftarrow \mathbf{not}p \quad (15)$$

$$\mathbf{K}s \leftarrow \mathbf{not}r \quad (16)$$

$$\mathbf{K}\neg q \leftarrow \quad (17)$$

Let $\mathcal{M}_G^{ex} = \{\mathcal{I} \mid \mathcal{I} \models_4 \{q \wedge \neg q \wedge (q \supset r)\}\}$. It can be verified that \mathcal{M}_G^{ex} is a paraconsistent MKNF model of \mathcal{K}_G^{ex} .

Definition 16. Let \mathcal{K}_G be a ground para-MKNF knowledge base, and P_h a subset of $HA(\mathcal{K}_G)$. The objective knowledge of P_h w.r.t. $HA(\mathcal{K}_G)$ is the first-order theory $OB_{\mathcal{O}, P_h}$ defined by

$$OB_{\mathcal{O}, P_h} = \{\pi(\mathcal{O})\} \cup \{\xi \mid \mathbf{K} \xi \in P_h\}.$$

In Example 7, let $P_h^{ex} = \{\mathbf{K}q, \mathbf{K}\neg q\}$. The paraconsistent MKNF model \mathcal{M}_G^{ex} equals $\{\mathcal{I} \mid \mathcal{I} \models_4 OB_{\mathcal{O}^{ex}, P_h^{ex}}\}$.

Definition 17. For a paraconsistent MKNF interpretation \mathcal{M} and a set of ground \mathbf{K} -atoms S , the subset of S paraconsistently induced by \mathcal{M} is the set $\{\mathbf{K} \xi \in S \mid \mathcal{M} \models_4 \xi\}$.

Lemma 2. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground para-MKNF knowledge base, \mathcal{M} be a paraconsistent MKNF model of \mathcal{K}_G , and P_h be the subset of $HA(\mathcal{K}_G)$ paraconsistently induced by \mathcal{M} . Then \mathcal{M} coincides with the set of paraconsistent MKNF interpretations $\mathcal{M}' = \{\mathcal{I} \mid \mathcal{I} \models_4 OB_{\mathcal{O}, P_h}\}$.

PROOF. Given $\mathcal{I} \in \mathcal{M}$, since \mathcal{M} is a paraconsistent MKNF model of \mathcal{K}_G , we clearly have $\mathcal{I} \models_4 \mathcal{O}$. For ξ such that $\mathbf{K}\xi \in P_h$, $\mathcal{M} \models_4 \xi$ from the fact that P_h is paraconsistently induced by \mathcal{M} . Then $\mathcal{I} \models_4 \xi$, thus $\mathcal{I} \in \mathcal{M}'$, and $\mathcal{M} \subseteq \mathcal{M}'$.

Suppose $\mathcal{I}' \in \mathcal{M}'/\mathcal{M}$. From the definition of \mathcal{M}' , $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathcal{O}$. For every $r_G \in \mathcal{P}_G$ of form (8) in Section 3, if $\mathcal{M} \not\models_4 \mathbf{K}H_j$, for every $1 \leq j \leq n$, then either $\mathcal{M} \not\models_4 \mathbf{K}A_j$, for some $n+1 \leq j \leq m$; or $\mathcal{M} \not\models_4 \mathbf{not}B_j$, for some $m+1 \leq j \leq k$. Since $\mathcal{M} \subseteq \mathcal{M}'$, $\mathcal{M} \not\models_4 \mathbf{K}A_j$ implies $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \not\models_4 \mathbf{K}A_j$; and $(\mathcal{I}, \mathcal{M}, \mathcal{M}) \not\models_4 \mathbf{not}B_j$ implies $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \not\models_4 \mathbf{not}B_j$. Then $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 r_G$. If $\mathcal{M} \models_4 \mathbf{K}H_j$, for some $1 \leq j \leq n$, then $\mathbf{K}H_j \in P_h$, from the definition of \mathcal{M}' , $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathbf{K}H_j$. Therefore $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 r_G$. This contradicts the assumption that \mathcal{M} is a paraconsistent MKNF model of \mathcal{K}_G . \square

Accordingly, every paraconsistent MKNF model is represented by a subset of $\text{HA}(\mathcal{K}_G)$, and conversely each subset of $\text{HA}(\mathcal{K}_G)$ corresponds to a paraconsistent MKNF interpretation of \mathcal{K}_G . We will get the paraconsistent MKNF models of \mathcal{K}_G by finding proper subsets of $\text{HA}(\mathcal{K}_G)$.

5.1. Positive Rules

A positive MKNF rule has the form

$$\mathbf{K}H_1 \vee \dots \vee \mathbf{K}H_n \leftarrow \mathbf{K}A_1 \wedge \dots \wedge \mathbf{K}A_m \quad (18)$$

where H_i and A_i are literals occurring in \mathcal{K}_G .

To search for the appropriate P_h , we define a fixpoint operator.

Definition 18. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P})$ be a ground positive para-MKNF knowledge base and $\mathbb{S} \in 2^{2^{\text{HA}(\mathcal{K}_G)}}$. A mapping $\mathfrak{T}_{\mathcal{K}_G} : 2^{2^{\text{HA}(\mathcal{K}_G)}} \rightarrow 2^{2^{\text{HA}(\mathcal{K}_G)}}$ is defined as

$$\mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}) = \bigcup_{\mathbb{S} \in \mathbb{S}} T_{\mathcal{K}_G}(\mathbb{S}),$$

where the mapping $T_{\mathcal{K}_G} : 2^{\text{HA}(\mathcal{K}_G)} \rightarrow 2^{2^{\text{HA}(\mathcal{K}_G)}}$ is defined as follows.

- If $\text{OB}_{\mathcal{O}, \mathbb{S}} \models_4 A_i, 1 \leq i \leq m$ for some ground integrity constraint $\leftarrow \mathbf{K}A_1 \wedge \dots \wedge \mathbf{K}A_m$ in \mathcal{P}_G , then $T_{\mathcal{K}_G}(\mathbb{S}) = \emptyset$.
- Otherwise, $T_{\mathcal{K}_G}(\mathbb{S}) = \{\mathbf{Q} \subseteq \text{HA}(\mathcal{K}_G) \mid \mathbf{Q} = \mathbb{S} \cup \mathbf{R} \cup \mathbf{H}\}$, where $\mathbf{R} = \{\mathbf{K}H_i \mid \text{for each ground MKNF rule } C_j \text{ of form (18) in } \mathcal{P}_G \text{ such that } \text{OB}_{\mathcal{O}, \mathbb{S}} \models_4 A_t \text{ for each } 1 \leq t \leq m\}$, and $\mathbf{H} = \{\mathbf{K}\xi \in \text{HA}(\mathcal{K}_G) \mid \text{OB}_{\mathcal{O}, \mathbb{S}} \models_4 \xi\}$.

Given a set $\mathbb{S} \subseteq \text{HA}(\mathcal{K}_G)$, for each ground rule C_j in \mathcal{P}_G such that $\text{OB}_{\mathcal{O}, \mathbb{S}} \models_4 A_t$ for each $1 \leq t \leq m$, we choose one rule head in every such MKNF rule C_j , this collection is the set \mathbf{R} . Since the MKNF rule is disjunctive, there may be various choices when we choose the rule head, and hence $T_{\mathcal{K}_G}(\mathbb{S})$ is the set collecting all the choices.

The definition of operator $T_{\mathcal{K}_G}$ differs from the fixpoint operator in ASP. In fact, we substitute the set \mathbb{S} for $\text{OB}_{\mathcal{O}, \mathbb{S}}$, which means that the DL-part and the rule part in the knowledge base \mathcal{K}_G can directly affect each other when reasoning with \mathcal{K}_G . This is the original intension of combining DLs and rules. Moreover, $\mathfrak{T}_{\mathcal{K}_G}$ is defined on $2^{2^{\text{HA}(\mathcal{K}_G)}}$, not on $2^{\text{HA}(\mathcal{K}_G)}$, indirectly leading to the fact that $\mathfrak{T}_{\mathcal{K}_G}$ is not monotonic. The following example gives evidence of the non-monotonicity of $\mathfrak{T}_{\mathcal{K}_G}$.

Example 8. Consider a para-MKNF knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, where $\mathcal{O} = \{p, p \supset f\}$ and $\mathcal{P} = \{\mathbf{K}\neg c \vee \mathbf{K}\neg f \leftarrow \mathbf{K}p; \mathbf{K}e \leftarrow \mathbf{K}\neg f\}$.

Let $\mathbb{S}_1, \mathbb{S}_2$ be subsets of $2^{\text{HA}(\mathcal{K}_G)}$, such that $\mathbb{S}_1 = \emptyset$, and $\mathbb{S}_2 = \{\{\mathbf{K}\neg f\}, \{\mathbf{K}\neg c\}\}$. Clearly, $\mathbb{S}_1 \subset \mathbb{S}_2$. However, we cannot conclude $\mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}_1) \subset \mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}_2)$.

In fact, $\mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}_1) = \mathfrak{T}_{\mathcal{K}_G}(\emptyset) = \{\{\mathbf{K}\neg f\}, \{\mathbf{K}\neg c\}\}$. $\mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}_2) = T_{\mathcal{K}_G}(\{\mathbf{K}\neg f\}) \cup T_{\mathcal{K}_G}(\{\mathbf{K}\neg c\})$, in which $T_{\mathcal{K}_G}(\{\mathbf{K}\neg f\}) = \{\{\mathbf{K}\neg f, \mathbf{K}e\}, \{\mathbf{K}\neg c, \mathbf{K}\neg f, \mathbf{K}e\}\}$, and $T_{\mathcal{K}_G}(\{\mathbf{K}\neg c\}) = \{\{\mathbf{K}\neg c\}, \{\mathbf{K}\neg f, \mathbf{K}\neg c\}\}$. Then $\mathfrak{T}_{\mathcal{K}_G}(\mathbb{S}_2) = \{\{\mathbf{K}\neg c, \mathbf{K}\neg f, \mathbf{K}e\}, \{\mathbf{K}\neg f, \mathbf{K}e\}, \{\mathbf{K}\neg c\}, \{\mathbf{K}\neg f, \mathbf{K}\neg c\}\}$.

Since $\mathfrak{T}_{\mathcal{K}_G}$ is not monotonic, then the Knaster-Tarski theorem does not apply. Therefore we employ a procedure that is also used in [38] to compute the fixpoint of $\mathfrak{T}_{\mathcal{K}_G}$.

$$\begin{aligned}\mathfrak{T}_{\mathcal{K}_G} \uparrow 0 &= \emptyset \\ \mathfrak{T}_{\mathcal{K}_G} \uparrow n + 1 &= \mathfrak{T}_{\mathcal{K}_G}(\mathfrak{T}_{\mathcal{K}_G} \uparrow n) \\ \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi &= \bigcup_{\alpha < \varpi} \bigcap_{\alpha \leq n < \varpi} \mathfrak{T}_{\mathcal{K}_G} \uparrow n\end{aligned}$$

where n is a successor ordinal and ϖ is a limit ordinal.

Lemma 3. $\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$ is a fixpoint.

PROOF. Let $I \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$, suppose $I \notin \mathfrak{T}_{\mathcal{K}_G}(\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi)$, then there is no set $J \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$ such that $I \in \mathfrak{T}_{\mathcal{K}_G}(\{J\})$. That is to say, for a set J such that $I \in \mathfrak{T}_{\mathcal{K}_G}(\{J\})$, we have $J \notin \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$. Then for any α there exists $\alpha \leq n < \varpi$ such that $J \notin \mathfrak{T}_{\mathcal{K}_G} \uparrow n$, which implies $I \notin \mathfrak{T}_{\mathcal{K}_G} \uparrow n + 1$. This contradicts the fact that $I \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$. Conversely, let $I \in \mathfrak{T}_{\mathcal{K}_G}(\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi)$, then there exists $J \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$ such that $I \in \mathfrak{T}_{\mathcal{K}_G}(\{J\})$. $J \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$ implies there exists $\alpha < \varpi$ such that $J \in \mathfrak{T}_{\mathcal{K}_G} \uparrow n$ for each $\alpha \leq n < \varpi$. Then $I \in \mathfrak{T}_{\mathcal{K}_G} \uparrow n$, for each $\alpha + 1 \leq n < \varpi$. Therefore $I \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi$. \square

Example 9. Continue to consider the knowledge base \mathcal{K} from Example 5. $\mathfrak{T}_{\mathcal{K}_G} \uparrow 0 = \emptyset$, $\mathfrak{T}_{\mathcal{K}_G} \uparrow 1 = \{\{\mathbf{K}\neg \text{Fly}(\text{Tweety})\}\}$, $\mathfrak{T}_{\mathcal{K}_G} \uparrow 2 = \mathfrak{T}_{\mathcal{K}_G} \uparrow 1$. Then $\mathfrak{T}_{\mathcal{K}_G} \uparrow 1$ is a fixpoint of the operator $\mathfrak{T}_{\mathcal{K}_G}$.

Lemma 4. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a para-MKNF knowledge base, and P_h be the subset of $\text{HA}(\mathcal{K}_G)$ paraconsistently induced by a paraconsistent MKNF interpretation \mathcal{M} of \mathcal{K}_G . Then $\mathcal{M} \models_4 \mathcal{K}_G$ if and only if $P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$.

PROOF. (Sufficiency) If $P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$, then $\{\mathbf{K}A_1, \dots, \mathbf{K}A_m\} \subseteq P_h$ implies that there exists a rule head $\mathbf{K}H_i$ such that in $\mathbf{K}H_i \in P_h$. Since P_h is the subset of $\text{HA}(\mathcal{K}_G)$ paraconsistently induced by \mathcal{M} , we infer $\mathcal{M} \models_4 \mathbf{K}A_j, 1 \leq j \leq m$, then $\mathcal{M} \models_4 \mathbf{K}H_i$. Therefore, $\mathcal{M} \models_4 \mathcal{K}_G$.

(Necessity) If $\mathcal{M} \models_4 \mathcal{K}_G$, then \mathcal{M} paraconsistently satisfies all the MKNF rules in the program \mathcal{P} . If $\mathcal{M} \models_4 \mathbf{K}A_j, 1 \leq j \leq m$, which means $\{\mathbf{K}A_1, \dots, \mathbf{K}A_m\} \subseteq P_h$, then there exist $i, 1 \leq i \leq n$, $\mathcal{M} \models_4 \mathbf{K}H_i$, which means that $\mathbf{K}H_i \in P_h$. Therefore, $P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$. \square

$P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$ indicates that implication " \leftarrow " is closed w.r.t. P_h . That is to say, if $\{\mathbf{K}A_1, \dots, \mathbf{K}A_m\} \subseteq P_h$, then there must exist a rule head $\mathbf{K}H_i$ in P_h , which is equivalent to $\mathcal{M} \models_4 \mathcal{P}_G$. Lemma 4 means that P_h corresponds to the paraconsistent MKNF model of \mathcal{K}_G .

Let $\gamma(\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi) = \{\mathbf{S} \mid \mathbf{S} \in \mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi, \text{ and } \mathbf{S} \in \mathfrak{T}_{\mathcal{K}_G}(\{\mathbf{S}\})\}$, and $\min(\mathbf{S}) = \{\mathbf{S} \mid \text{there exists no } \mathbf{Q} \in \mathbf{S} \text{ such that } \mathbf{Q} \subset \mathbf{S}\}$.

Theorem 3. *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground para-MKNF knowledge base and \mathcal{M} be a paraconsistent MKNF model of \mathcal{K}_G . Then \mathcal{M} coincides with $\mathcal{M}' = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P_h}\}$, where P_h is an element of the set $\mathbb{Q} = \min(\gamma(\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi))$.*

PROOF. If $P_h \in \mathbb{Q}$, then $P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$. From Lemma 4, $\mathcal{M}' \models_4 \mathcal{K}_G$. Suppose there exists an MKNF interpretation \mathcal{M}'' such that $\mathcal{M}'' \supsetneq \mathcal{M}'$ and $(\mathcal{I}, \mathcal{M}'', \mathcal{M}') \models_4 \mathcal{K}_G$ for each $\mathcal{I} \in \mathcal{M}''$. Let P'_h be the subset of $\text{HA}(\mathcal{K}_G)$ that is paraconsistently induced by \mathcal{M}'' , and $\mathcal{M}_1 = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P'_h}\}$. Then $(\mathcal{I}, \mathcal{M}_1, \mathcal{M}') \models_4 \mathcal{K}_G$ for each $\mathcal{I} \in \mathcal{M}_1$. From Lemma 4, $(\mathcal{I}, \mathcal{M}_1, \mathcal{M}') \models_4 \mathcal{K}_G$ implies $P'_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P'_h\})$. Then $P'_h \in \gamma(\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi)$, and $\mathcal{M}' \supsetneq \mathcal{M}$ implies $P'_h \subsetneq P_h$, which contradicts the minimality of P_h . Therefore, the claim holds.

If \mathcal{M} is a paraconsistent MKNF model of \mathcal{K}_G , let $P_h \subseteq \text{HA}(\mathcal{K}_G)$ be the subset of $\text{HA}(\mathcal{K}_G)$ that is paraconsistently induced by \mathcal{M} . From Lemma 2, $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P_h}\}$, and from Lemma 4, $P_h \in \mathfrak{T}_{\mathcal{K}_G}(\{P_h\})$. Suppose P_h is not in \mathbb{Q} , then there exists P'_h , such that $P'_h \subsetneq P_h$. Let $\mathcal{M}' = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P'_h}\}$, $\mathcal{M}' \supsetneq \mathcal{M}$. From the definition of \mathcal{M}' , $\mathcal{M}' \models_4 \mathcal{O}$. For a rule r_G , if $\text{OB}_{\mathcal{O}, P'_h} \models_4 A_i, 1 \leq i \leq m$, then there exists a rule head $\mathbf{K}H_j \in P'_h, 1 \leq j \leq n$, thus $\mathcal{M}' \models_4 r_G$ and then $\mathcal{M}' \models_4 \mathcal{K}_G$, which contradicts the fact that \mathcal{M} is a paraconsistent MKNF model of \mathcal{K}_G . \square

When \mathcal{P}_G is nondisjunctive, in which MKNF rules are of the following form

$$\mathbf{K}H \leftarrow \mathbf{K}A_1 \wedge \dots \wedge \mathbf{K}A_m \quad (19)$$

where H_i and A_i are literals occurring in \mathcal{K}_G , the rule head in each rule in \mathcal{P}_G has at most one element. If $\text{OB}_{\mathcal{O}, P_h}$ paraconsistently satisfies the rule body, then there is at most one candidate to be added to the set \mathbf{R} . In this case, the set \mathbf{Q} has only one element.

Proposition 2. *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground para-MKNF knowledge base, where \mathcal{P}_G is a positive nondisjunctive program. Then \mathcal{K}_G has a unique paraconsistent MKNF model.*

In fact, the fixpoint operator can be simplified when \mathcal{P}_G is nondisjunctive.

Definition 19. *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground nondisjunctive positive para-MKNF knowledge base. The fixpoint operator $\mathbb{T}_{\mathcal{K}_G}: 2^{\text{HA}(\mathcal{K}_G)} \rightarrow 2^{\text{HA}(\mathcal{K}_G)}$ is defined as follows:*

$$\begin{aligned} \mathbb{T}_{\mathcal{K}_G}(\mathbf{S}) = \{ & \mathbf{KH} \mid \text{for each } r_G \in \mathcal{P}_G \text{ of the form (19) such that} \\ & \text{OB}_{\mathcal{O}, \mathbf{S}} \models_4 A_t \text{ for each } 1 \leq t \leq m\} \cup \\ & \{ \mathbf{K}\xi \in \text{HA}(\mathcal{K}_G) \mid \text{OB}_{\mathcal{O}, \mathbf{S}} \models_4 \xi \} \end{aligned}$$

It can be easily verified that $\mathbb{T}_{\mathcal{K}_G}$ is monotonic. Then it follows from the Knaster-Tarski's theorem that $\mathbb{T}_{\mathcal{K}_G}$ has a unique least fixpoint $\mathbb{T}_{\mathcal{K}_G}^\infty$. $\mathbb{T}_{\mathcal{K}_G}^\infty$ can be computed as in ASP by setting $S_0 = \emptyset$ and $S_i = \mathbb{T}_{\mathcal{K}_G}(S_{i-1})$. If \mathcal{P}_G is a finite set, $S_n = S_{n+1} = \dots = \mathbb{T}_{\mathcal{K}_G}^\infty$.

Proposition 3. *For a ground nondisjunctive positive para-MKNF knowledge base \mathcal{K}_G , $\mathfrak{T}_{\mathcal{K}_G} \uparrow \varpi = \{\mathbb{T}_{\mathcal{K}_G}^\infty\}$. That is to say, the paraconsistent MKNF model of \mathcal{K}_G is $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, \mathbb{T}_{\mathcal{K}_G}^\infty}\}$.*

The constructing idea of the operator $\mathbb{T}_{\mathcal{K}_G}$ is the same as the one of the operator $\mathfrak{T}_{\mathcal{K}_G}$. Then Proposition 3 follows.

5.2. General Rules

For general rules, we cannot apply the fixpoint operator directly to the rules, since reasoning with the modal **not** operator is nonmonotonic. Therefore, we transform each general program to a positive program, and then compute the fixpoint of the transformed program.

Definition 20. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground para-MKNF knowledge base. Then its transformation is defined as \mathcal{K}_G^* obtained by replacing each general rule in \mathcal{P}_G with the following positive MKNF rule.

$$\mathbf{K}\mu_1 \vee \dots \vee \mathbf{K}\mu_n \vee \mathbf{K}B_{m+1} \vee \dots \vee \mathbf{K}B_k \leftarrow \mathbf{K}A_{n+1} \wedge \dots \wedge \mathbf{K}A_m, \quad (20)$$

$$\mathbf{K}H_i \leftarrow \mathbf{K}\mu_i \quad \text{for } 1 \leq i \leq n, \quad (21)$$

$$\leftarrow \mathbf{K}\mu_i \wedge \mathbf{K}B_j \quad \text{for } 1 \leq i \leq n, m+1 \leq j \leq k, \quad (22)$$

$$\mathbf{K}\mu_i \leftarrow \mathbf{K}H_i \wedge \mathbf{K}\mu_j \quad \text{for } 1 \leq i, j \leq n. \quad (23)$$

This transformation, following Chiaki Sakama and Katsumi Inoue [38], is originally introduced in [19] in a different form, in order to compute *answer sets* for any class of function-free logic programs, including the extended disjunctive program of form (9).

The modal operator \mathbf{K} can be understood as "believe". The intuition of the transformed clauses is that if A_{n+1}, \dots, A_m are believed to be true, then there exists a literal H_i ($1 \leq i \leq n$), such that H_i is believed to be true via μ_i when B_{m+1}, \dots, B_k are not believed to be true; otherwise, there exists a literal B_j ($m+1 \leq j \leq k$), such that B_j is believed to be true.

Given a set \mathbb{S}^* that is a subset of $2^{\text{HA}(\mathcal{K}_G^*)}$, $\Phi(\mathbb{S}^*) = \{\mathbb{S}^* \cap \mathbf{K}\mathbf{A}(\mathcal{K}_G) \mid \mathbb{S}^* \in \mathbb{S}^*\}$.

Theorem 4. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground para-MKNF knowledge base, then each paraconsistent MKNF model of \mathcal{K}_G equals $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \mathbf{O}\mathbf{B}_{\mathcal{O}, P_h}\}$, where P_h is an element of the set $\mathbb{Q} = \Phi(\min(\gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi)))$.

PROOF. First of all, we prove every element in \mathbb{Q} corresponds to a paraconsistent MKNF model of \mathcal{K}_G . Suppose that $P_h \in \mathbb{Q}$, then there exists $P_h^* \in \min(\gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi))$, such that $P_h = P_h^* \cap \mathbf{K}\mathbf{A}(\mathcal{K}_G)$. Let $\mathcal{M}^* = \{\mathcal{I} \mid \mathcal{I} \models_4 \mathbf{O}\mathbf{B}_{\mathcal{O}, P_h^*}\}$. From Theorem 3, \mathcal{M}^* is a paraconsistent MKNF model of \mathcal{K}_G^* . $\mathcal{M} \models_4 \mathcal{O}$ by definition of \mathcal{M} . For each MKNF rule r_G of form (8) in \mathcal{P}_G , if $\mathcal{M} \models_4 \mathbf{K}A_i$ for $n+1 \leq i \leq m$ and $\mathcal{M} \models_4 \mathbf{not}B_j$, for $m+1 \leq j \leq k$, then from $P_h \subseteq P_h^*$, we have $\mathcal{M}^* \models_4 A_i$ for $n+1 \leq i \leq m$ and $\mathbf{K}B_j \notin P_h$ for $m+1 \leq j \leq k$. $\mathbf{K}B_j \in \mathbf{K}\mathbf{A}(\mathcal{K}_G)$ implies $\mathbf{K}B_j \notin P_h^*$. Since \mathcal{M}^* paraconsistently satisfies each rule in \mathcal{P}_G^* , then there exists $\mu_l, 1 \leq l \leq n$, such that $\mathcal{M}^* \models_4 \mathbf{K}\mu_l$, and by rule (21) $\mathcal{M}^* \models_4 \mathbf{K}H_l$. $\mathbf{K}H_l \in \mathbf{K}\mathbf{A}(\mathcal{K}_G)$ implies $\mathbf{K}H_l \in P_h$. Therefore, $\mathcal{M} \models_4 \mathbf{K}H_l$, for some $1 \leq l \leq n$. Then $\mathcal{M} \models_4 \mathcal{K}_G$.

Suppose there exists a paraconsistent MKNF interpretation $\mathcal{M}'' \supseteq \mathcal{M}$ such that $(\mathcal{I}'', \mathcal{M}'', \mathcal{M}) \models_4 \mathcal{K}_G$. Let P'_h be the subset of $\text{HA}(\mathcal{K}_G)$ paraconsistently induced by \mathcal{M}'' , and $\mathcal{M}' = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P'_h}\}$. Then we can infer that $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathcal{K}_G$. Clearly, $P'_h \subsetneq P_h$. For simplicity let $P_h \setminus P'_h = \{\mathbf{K}\xi\}$, $P'_h = P_h \setminus \{\mathbf{K}\xi, \mathbf{K}\mu_i\}$, in which $\mathbf{K}\xi \leftarrow \mathbf{K}\mu_i$, and $P'_h = P'_h \cap \text{KA}(\mathcal{K}_G)$. Let $\mathcal{M}^{*} = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P_h^*}\}$. Obviously, $(\mathcal{I}, \mathcal{M}^*, \mathcal{M}^*) \models_4 \mathcal{O}$. For a rule r_G^* of form (20), it corresponds to a rule r_G of form (8) in \mathcal{P}_G . If $(\mathcal{I}, \mathcal{M}^*, \mathcal{M}^*) \models_4 \mathbf{KA}_i$ for each $n+1 \leq i \leq m$, then $\mathbf{KA}_i \in P'_h$. $\mathbf{KA}_i \in \text{KA}(\mathcal{K}_G)$ implies $\mathbf{KA}_i \in P_h$. Then $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathbf{KA}_i$. From $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathcal{K}_G$, we know $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 r_G$. In case of (i), if $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \text{not}B_j$, for each $m+1 \leq j \leq k$, which implies $\mathbf{KB}_j \notin P'_h$ and $\mathcal{M} \models_4 \text{not}B_j$, for each $m+1 \leq j \leq k$, then there exist $H_l, 1 \leq l \leq n$ such that $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \models_4 \mathbf{KH}_l$, which means $\mathbf{KH}_l \in P'_h \subseteq P_h^*$. On the other hand, $P'_h \subseteq P_h \subseteq P_h^*$ implies $\mathbf{KA}_i \in P_h \subseteq P_h^*$ for each $n+1 \leq i \leq m$, since \mathcal{M}^* is a paraconsistent MKNF model of \mathcal{K}_G , there exists a $\mu_v, 1 \leq v \leq n$, such that $\mathcal{M}^* \models_4 \mathbf{K}\mu_v$, since there exists \mathbf{KH}_l such that $\mathbf{KH}_l \in P'_h \subseteq P_h \subseteq P_h^*$, then from rule (23), $\mathbf{K}\mu_l \in P_h^*$. $\mathbf{KH}_l \in P'_h$ means \mathbf{KH}_l is not $\mathbf{K}\xi$, then $\mathbf{K}\mu_l \in P_h^*$. Therefore $(\mathcal{I}, \mathcal{M}^*, \mathcal{M}^*) \models_4 r_G^*$. In case of (ii), if $(\mathcal{I}', \mathcal{M}', \mathcal{M}) \not\models_4 \text{not}B_j$, for some $j, m+1 \leq j \leq k$, which implies $\mathbf{KB}_j \in P'_h \subseteq P_h^*$. Then $(\mathcal{I}, \mathcal{M}^*, \mathcal{M}^*) \models_4 r_G^*$. For other rules r_G^* of form (21), (22), and (23) in P_G^* , it can be easily inferred $(\mathcal{I}, \mathcal{M}^*, \mathcal{M}^*) \models_4 r_G^*$ by the fact that $P_h^* \subsetneq P_h$. Then it contradicts that \mathcal{M}^* is a paraconsistent MKNF model of \mathcal{K}_G^* . Therefore every element in \mathbb{Q} is a paraconsistent MKNF model of \mathcal{K}_G .

Conversely, suppose \mathcal{M}_1 is the paraconsistent MKNF model of \mathcal{K}_G . From Lemma 2 we imply that $\mathcal{M}_1 = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P_h}\}$, where P_h is the subset of $\text{HA}(\mathcal{K}_G)$ paraconsistently induced by \mathcal{M}_1 . For each MKNF rule, let $\mathbf{S}_\mu = \bigcup_r \{\mathbf{K}\mu_l \mid \mathcal{M}_1 \models_4 A_i, \text{ for each } n+1 \leq i \leq m, \mathcal{M}_1 \not\models_4 B_j, \text{ for each } m+1 \leq j \leq k, \text{ and } \mathcal{M}_1 \models_4 H_l\}$. Let $\mathbf{S} = P_h \cup \mathbf{S}_\mu$. Then, \mathbf{S} satisfies all the MKNF rules in P_G^* and $\mathbf{S} \in \gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi)$ by the construction. Now we define $\mathbf{S}^* = P_h \cup \mathbf{S}'$, in which \mathbf{S}' is the minimal subset of \mathbf{S}_μ such that each $\mathbf{K}\mu_l$ is chosen in a way that satisfies the MKNF rules in P_G^* . Now we need to prove \mathbf{S}^* to be minimal in $\gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi)$. Assume that $\mathbf{S}_1^* \in \gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi)$ such that $\mathbf{S}_1^* \subset \mathbf{S}^*$. Since \mathbf{S}^* is defined to be the minimal set with respect to elements in \mathbf{S}_μ , then $\mathbf{S}_1^* \cap \text{KA}(\mathcal{K}_G) \subset \mathbf{S}^* \cap \text{KA}(\mathcal{K}_G)$, and thus there exists $\mathbf{K}\xi \in \mathbf{S}^* \setminus \mathbf{S}_1^*$, and $\mathbf{K}\xi \in P_h$. Let $P'_h = P_h \setminus \mathbf{K}\xi = \mathbf{S}_1^* \cap \text{KA}(\mathcal{K}_G)$ and $\mathcal{M}'_1 = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, P'_h}\}$. Then from $\mathbf{S}_1^* \in \gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi)$, we can easily infer that $(\mathcal{I}, \mathcal{M}'_1, \mathcal{M}_1) \models_4 \mathcal{K}_G$, which contradicts the fact that \mathcal{M}_1 is the paraconsistent MKNF model of

\mathcal{K}_G . Then $\mathbf{S}^* \in \min(\gamma(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow \varpi))$. That is to say, for each paraconsistent MKNF model \mathcal{M} of \mathcal{K}_G , there exists an element \mathbf{S} in \mathbb{Q} such that $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, \mathbf{S}}\}$. \square

Example 10. Consider a ground knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, where $\mathcal{O} = \{p, p \supset b, b \supset f\}$, $\mathcal{P}_G = \{\mathbf{K}\neg f \vee \mathbf{K}\neg c \leftarrow \mathbf{K}p; \mathbf{K}a \leftarrow \mathbf{K}\neg c; \mathbf{K}f \leftarrow \mathbf{K}b \wedge \mathbf{not}p\}$, and p, b, f, c, a are atoms. By Definition 20, \mathcal{P}_G is transformed to \mathcal{P}_G^* :

$$\begin{aligned} \mathbf{K}\neg f \vee \mathbf{K}\neg c &\leftarrow \mathbf{K}p \\ \mathbf{K}a &\leftarrow \mathbf{K}\neg c \\ \mathbf{K}\mu \vee \mathbf{K}p &\leftarrow \mathbf{K}b \\ &\leftarrow \mathbf{K}\mu \wedge \mathbf{K}p \\ \mathbf{K}f &\leftarrow \mathbf{K}\mu \end{aligned}$$

We compute the fixpoint by applying the procedure presented in section 5.1 to the knowledge base $\mathcal{K}_G^* = (\mathcal{O}, \mathcal{P}_G^*)$. By evaluating $\mathfrak{T}_{\mathcal{K}_G^*} \uparrow n + 1 = \mathfrak{T}_{\mathcal{K}_G^*} \mathfrak{T}_{\mathcal{K}_G^*} \uparrow n$ recursively, $\min(\mathfrak{T}_{\mathcal{K}_G^*} \uparrow n) = \{\{\mathbf{K}\neg c, \mathbf{K}a, \mathbf{K}p, \mathbf{K}f\}, \{\mathbf{K}\neg f, \mathbf{K}p, \mathbf{K}f\}\}$. Since the computation process is tedious, we omit it here. Also, it can be easily verified that $\mathcal{M}_1 = \{\mathcal{I} \mid \mathcal{I} \models_4 \{p, b, f, \neg c, a\}\}$, and $\mathcal{M}_2 = \{\mathcal{I} \mid \mathcal{I} \models_4 \{p, b, f, \neg f\}\}$ are two paraconsistent MKNF models of \mathcal{K}_G . Note that \mathcal{M}_1 is the preferred MKNF model of \mathcal{K}_G , and \mathcal{K}_G is a consistent knowledge base.

5.3. Stratified Rules

In this section, we discuss the class of stratified program, which are nondisjunctive, but can contain **not**-atoms. Stratified rules are of form (24), and they can be separated into strata, each of which can be evaluated separately.

$$\mathbf{K}H \leftarrow \mathbf{K}A_1 \wedge \dots \wedge \mathbf{K}A_m \wedge \mathbf{not}B_{m+1} \wedge \dots \wedge \mathbf{not}B_k \quad (24)$$

Definition 21 ([33]). Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground nondisjunctive MKNF knowledge base, and let $\sigma: \text{KA}(\mathcal{K}_G) \rightarrow \mathbb{N}^+$ be a function assigning an integer to each **K**-atom. For a modal atom $\mathbf{K}\xi$, the sets of **K**-atoms $[\mathbf{K}\xi]^\uparrow$ and $[\mathbf{K}\xi]^\downarrow$ are defined as follows:

$$[\mathbf{K}\xi]^\uparrow = \{\mathbf{K}\varphi \mid \mathbf{K}\varphi \in \text{HA}(\mathcal{K}_G) \text{ such that } \sigma(\mathbf{K}\varphi) \leq \sigma(\mathbf{K}\xi)\}$$

$$[\mathbf{K}\xi]^\downarrow = \text{HA}(\mathcal{K}_G) \setminus [\mathbf{K}\xi]^\uparrow$$

We say that σ is a *stratification* of \mathcal{P}_G if the following conditions hold:

- (1) For each rule $r_G \in \mathcal{P}_G$ of the form (24), $\sigma(\mathbf{K}H) \geq \sigma(\mathbf{K}B_i)$ for each $1 \leq i \leq m$, and $\sigma(\mathbf{K}H) > \sigma(\mathbf{K}B_i)$ for each $m + 1 \leq i \leq k$;
- (2) For each atom $\mathbf{K}\xi \in \text{KA}(\mathcal{K}_G)$ and each subset $S_h \subseteq [\mathbf{K}\xi]^\downarrow$, if $\text{OB}_{\mathcal{O}, S_h} \not\models_4 \xi$, then $\text{OB}_{\mathcal{O}, S_h \cup [\mathbf{K}\xi]^\uparrow} \not\models_4 \xi$.

The program \mathcal{P}_G is *stratified* if a stratification σ exists. A stratification σ partitions \mathcal{P}_G into *strata* $\mathcal{P}_G^1, \dots, \mathcal{P}_G^\zeta$ as follows:

$$\mathcal{P}_G^i = \{r_G \in \mathcal{P}_G \mid \sigma(\mathbf{K}H) = i, \text{ where } \mathbf{K}H \text{ is the head atom of } r_G\}$$

This sequence is often identified with σ and is called a *stratification*.

A nonground MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ is *stratified* if $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ is stratified.

Let P and N be disjoint subsets of $\text{KA}(\mathcal{K}_G)$. The program $\mathcal{P}_G[\text{not}, P, N]$ is obtained from \mathcal{P}_G by replacing each **not** ξ with \top if $\mathbf{K}\xi \in N$ and $\mathbf{K}\neg\xi \in N$; with **t** if $\mathbf{K}\xi \in N$ and $\mathbf{K}\neg\xi \notin N$; with **f** if $\mathbf{K}\xi \in P$ and $\mathbf{K}\neg\xi \notin N$; with \perp if $\mathbf{K}\xi \in P$ and $\mathbf{K}\neg\xi \in P$.

A paraconsistent MKNF model of a stratified MKNF knowledge base can be computed by processing strata sequentially.

Definition 22. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground stratified para-MKNF knowledge base and $\mathcal{P}_G^1, \dots, \mathcal{P}_G^\zeta$ a stratification of \mathcal{P}_G . The sequence of subsets U_0, \dots, U_ζ of $\text{HA}(\mathcal{K}_G)$ is inductively defined as follows:

$$U_0 = \emptyset; \tag{25}$$

$$P_i = \{\mathbf{K}\xi \in \text{HA}(\mathcal{K}_G) \mid \sigma(\xi) < i \text{ and } \text{OB}_{\mathcal{O}, U_{i-1}} \models_4 \xi\}; \tag{26}$$

$$N_i = \{\mathbf{K}\xi \in \text{HA}(\mathcal{K}_G) \mid \sigma(\xi) < i \text{ and } \text{OB}_{\mathcal{O}, U_{i-1}} \not\models_4 \xi\}; \tag{27}$$

$$\chi_i = \{\mathbf{K}\xi \leftarrow \mid \mathbf{K}\xi \in U_{i-1}\} \cup \mathcal{P}_G^i[\text{not}, P_i, N_i]; \tag{28}$$

$$U_i = \mathbb{T}_{G_i}^\infty, \tag{29}$$

where $G_i = (\mathcal{O}, \chi_i)$, and $1 \leq i \leq \zeta$ for equations (26) - (29). We define $U_{\mathcal{K}_G}^\infty = U_\zeta$.

Theorem 5. Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a stratified ground para-MKNF knowledge base, $U_{\mathcal{K}_G}^\infty$ be a set obtained from Definition 22 using any stratification, and $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, U_{\mathcal{K}_G}^\infty}\}$.

- If $\mathcal{M} \neq \emptyset$, then \mathcal{M} is a paraconsistent MKNF model of \mathcal{K}_G .
- If \mathcal{K}_G has a paraconsistent MKNF model, then this model is equal to \mathcal{M} .

PROOF. Let $\mathcal{P}_G^1, \dots, \mathcal{P}_G^\zeta$ be a stratification of \mathcal{P}_G used to compute $U_{\mathcal{K}_G}^\infty$. Moreover, let $\mathcal{P}_G^{\leq i} = \bigcup_{j \leq i} \mathcal{P}_G^j$, let $\mathcal{K}_G^{\leq i} = (\mathcal{O}, \mathcal{P}_G^{\leq i})$, and $\mathcal{M}_i = \{\mathcal{I} \mid \mathcal{I} \models_4 \text{OB}_{\mathcal{O}, U_i}\}$, for $0 \leq i \leq \zeta$. Each G_i is a positive nondisjunctive program, so by Proposition 2, it has at most a paraconsistent MKNF model corresponding to \mathcal{M}_i by Proposition 3.

Since each χ_i contains MKNF rules $\mathbf{K}\xi \leftarrow$ such that $\mathbf{K}\xi \in U_{i-1}$, then $U_{i-1} \subseteq U_i$. For each $\mathbf{K}\xi \in \text{KA}(\mathcal{K}_G)$, $\sigma(\mathbf{K}\xi) = i$, and each $j > i$, we have $\mathcal{M}_i \models_4 \mathbf{K}\xi$ if and only if $\mathcal{M}_j \models_4 \mathbf{K}\xi$. We denote this property by (\dagger) . In fact, if $\text{OB}_{\mathcal{O}, U_i} \models_4 \xi$, then $\text{OB}_{\mathcal{O}, U_j} \models_4 \xi$, since $U_i \subseteq U_j$. If $\text{OB}_{\mathcal{O}, U_j} \models_4 \xi$, but $\text{OB}_{\mathcal{O}, U_i} \not\models_4 \xi$. Since $U_j \setminus U_i \subseteq [\mathbf{K}\xi]^\uparrow$, we have $\text{OB}_{\mathcal{O}, U_i \cup [\mathbf{K}\xi]^\uparrow} \models_4 \xi$, which contradicts Condition (2) of the definition of stratification. Therefore, $\text{OB}_{\mathcal{O}, U_i} \models_4 \xi$ if and only if $\text{OB}_{\mathcal{O}, U_j} \models_4 \xi$, which is equivalent to property (\dagger) .

We now prove by induction on $1 \leq i \leq \zeta$ that both claims of this theorem hold for $\mathcal{K}_G^{\leq i}$ and \mathcal{M}_i . The base case $i = 0$ holds obviously, so we consider the induction step.

(Claim 1) Assume that $\mathcal{M}_i \neq \emptyset$, then $\mathcal{M}_{i-1} \neq \emptyset$ as well. By induction assumption \mathcal{M}_{i-1} is a paraconsistent MKNF model of $\mathcal{K}_G^{\leq i-1}$. Consider each MKNF rule $r_G \in \mathcal{P}_G^{\leq i-1}$, for each rule head $\mathbf{K}H$ and each rule body $\mathbf{K}A_t$ ($1 \leq t \leq m$) and $\text{not}B_j$ ($m+1 \leq j \leq k$), we have $\sigma(\mathbf{K}H) < i$, $\sigma(\mathbf{K}A_t) < i$ ($1 \leq t \leq m$) and $\sigma(\mathbf{K}B_j) < i$ ($m+1 \leq j \leq k$). Then we have $\mathcal{M}_i \models_4 \mathbf{K}\xi$ if and only if $\mathcal{M}_{i-1} \models_4 \mathbf{K}\xi$ by property (\dagger) , where $\xi \in \{H, A_t, B_j\}$ ($1 \leq t \leq m$ and $m+1 \leq j \leq k$). Since $\mathcal{M}_{i-1} \models_4 r_G$, we have $\mathcal{M}_i \models_4 r_G$. Consider each MKNF rule $r_G \in \mathcal{P}_G^i$, for each negative rule body $\text{not}B_j$ ($m+1 \leq j \leq k$), $\sigma(\mathbf{K}B_j) < i$ ($m+1 \leq j \leq k$). By property (\dagger) , we have $\mathcal{M}_i \models_4 r_G$ if and only if $\mathcal{M}_i \models_4 r_G[\text{not}, P_i, N_i]$. Thus we conclude $\mathcal{M}_i \models_4 \mathcal{P}_G^i$, then $\mathcal{M}_i \models_4 \mathcal{K}_G^{\leq i}$.

Assume that there exists a paraconsistent MKNF interpretation $\mathcal{M}'_i \supset \mathcal{M}_i$ such that $(\mathcal{I}'_i, \mathcal{M}'_i, \mathcal{M}_i) \models_4 \mathcal{K}_G^{\leq i}$ for each $\mathcal{I}'_i \in \mathcal{M}'_i$. For each $\mathbf{K}\xi \in \text{KA}(\mathcal{K}_G)$, such that $\sigma(\mathbf{K}\xi) < i$, we have $\mathcal{M}'_i \models_4 \mathbf{K}\xi$ if and only if $\mathcal{M}_{i-1} \models_4 \mathbf{K}\xi$.

Rules	$\mathcal{DL} = \phi$	$\mathcal{DL} \in \text{PTime}$	$\mathcal{DL} \in \text{coNP}$
Non-disjunctive positive	PTime	PTime	coNP
Stratified	PTime	PTime	Δ_2^p
General	coNP	coNP	Π_2^p
Disjunctive positive	coNP/ Π_2^p	coNP/ Π_2^p	coNP/ Π_2^p
Disjunctive general	Π_2^p	Π_2^p	Π_2^p

Table 3: Data Complexity of Instance Checking in Para-MKNF KBs. \mathcal{DL} indicates the base DL.

Otherwise, \mathcal{M}_{i-1} is not the paraconsistent MKNF model of $\mathcal{K}_G^{\leq i-1}$. But then $(\mathcal{I}'_i, \mathcal{M}'_i, \mathcal{M}_i) \models_4 \chi_i$, which contradicts the fact that \mathcal{M}_i is the paraconsistent MKNF model of G_i . Therefore, \mathcal{M}_i is a paraconsistent MKNF model of $\mathcal{K}_G^{\leq i}$.

(Claim 2) Assume that $\mathcal{K}_G^{\leq i}$ has a paraconsistent MKNF model \mathcal{M}'_i . For each $\mathbf{K}\xi \in \text{KA}(\mathcal{K}_G)$, such that $\sigma(\mathbf{K}\xi) < i$, we have $\mathcal{M}'_i \models_4 \mathbf{K}\xi$ if and only if $\mathcal{M}_{i-1} \models_4 \mathbf{K}\xi$. Otherwise, \mathcal{M}_{i-1} is not the paraconsistent MKNF model of $\mathcal{K}_G^{\leq i-1}$. But then \mathcal{M}'_i is an MKNF model of G_i . Therefore, $\mathcal{M}_i = \mathcal{M}'_i$.⁴

As discussed above, an algorithm for computing paraconsistent MKNF models follows from the fixpoint operators. From Theorem 2 in Section 4, we obtain that the data complexity in our paradigm is not higher than that for the classical knowledge base. Therefore, we obtain Table 3 from [33], which presents the data complexity of instance checking in different para-MKNF knowledge bases.

6. Related Work

A well-founded semantics [21] for hybrid MKNF has been proposed for better efficiency of reasoning. It extends two-valued semantics to three valued semantics, and is able to detect inconsistencies occurring in the knowledge base, however, cannot handle inconsistencies as such. Since [21] is restricted to nondisjunctive rules, it is less expressive than the four-valued semantics. The advantage of [21] is that when the considered DL is of polynomial data complexity, the combined approach remains polynomial.

⁴This proof is only a slight variation of the proof of Theorem 4.19 in [33], which we include for completeness of our treatment.

There are alternatives to four-valued paraconsistent semantics for DLs, due to the inherent limitation of four-valued logic. In fact, quasi-classical semantics [41], based on quasi-classical logic [18], is a paraconsistent semantics for \mathcal{ALC} to handle inconsistencies. A weak semantics was proposed, which was actually identical to $\mathcal{ALC}4$. The problem that Modus Ponens, Modus Tollens, and Disjunctive Syllogisms fail, which is inherent in $\mathcal{ALC}4$, also occurs in the weak semantics. A strong semantics which was built upon the weak semantics does not suffer from these problems.

Paradoxical description logic \mathcal{ALC}_{LP} [42] is an extension of \mathcal{ALC} with semantics of logics of paradox [36]. Paradoxical entailment satisfies the excluded middle rule and intuitive equivalence that are not valid in four-valued logics. Moreover, \mathcal{ALC}_{LP} has strong inferential power than $\mathcal{ALC}4$ on assertions and material inclusions.

Nevertheless, the distinct advantage of four-valued paraconsistent semantics is allowing classical reasoners to derive sound but non-trivial conclusions from even inconsistent knowledge bases by embedding them into the classical framework.

Chiaki Sakama and Katsumi Inoue [38] proposed a paraconsistent stable semantics for extended disjunctive programs. They introduced a fixpoint operator for disjunctive programs, which is the inspiration of our work on the fixpoint operator. Moreover, by substituting a nine-valued lattice for \mathcal{FOUR} as truth value set, they proposed semi-stable models, which is also used in [9] to cope with instability. The program $P = \{a \leftarrow \mathbf{nota}\}$ has no paraconsistent stable model, but has a semi-stable model. This method deserves further discussion in our work, in which there is no paraconsistent MKNF model for the MKNF rule $\mathbf{K}a \leftarrow \mathbf{nota}$.

7. Conclusion

In this paper we have presented a paraconsistent semantics of hybrid MKNF knowledge bases that is sound w.r.t. the classical two-valued semantics defined in [33], which restricts to the paraconsistent semantics of extended disjunctive program [38] and to the paraconsistent semantics of OWL [28], when the DL-part and LP-part is empty, respectively. Furthermore, we characterized paraconsistent MKNF models via fixpoint operators, and showed that the complexity of our paradigm is not higher than that in [33].

There are a number of paths to further develop this work. First of all, in [21, 20], a well-founded semantics was introduced for hybrid MKNF knowledge bases which has better complexity properties, and our paraconsistent approach could be carried over to this paradigm. Moreover, inconsistency is not the only problem that occurs in the real world, some other problems such as vagueness and probabilistic uncertainty, which cannot be coped with by classical reasoners either, may deserve some discussions and research. Then it is necessary to extend fuzzy semantics and probabilistic semantics to hybrid MKNF knowledge bases. Finally, an even tighter paraconsistent and non-monotonic integration of OWL and rules could furthermore be investigated. In [4, 22, 23, 24], nominal schemas are introduced as an extension to DL-based ontology languages, which provide sufficient expressivity to incorporate rule-based modeling into ontologies. Inconsistency handling is also an interesting problem which deserves to be discussed in the context of nominal schemas.

Acknowledgments

The authors thank Matthias Knorr and the anonymous referees for their valuable comments, which improved the quality of this paper. The authors acknowledge the support provided by the National Basic Research Program of China [2010CB334706, 2011CB311808], and the National Natural Science Foundation of China [11071061]. The last named author acknowledges support by the National Science Foundation under award 1017225 “III: Small: TROn – Tractable Reasoning with Ontologies.”

References

- [1] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. F. (Eds.): *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Second Edition (2007)
- [2] Belnap, N. D.: *A Useful Four-valued Logic*. *Modern uses of multiple-valued logics*. 7–73 (1977)
- [3] Berners-Lee, T., Hendler, J., Lassila, O.: *The Semantic Web*. *Scientific American*. 284 (5), 35–43 (2001)

- [4] Carral Martinez, D., Hitzler, P.: Extending Description Logic Rules. In: Simperl, E. et al. (eds.) Proceedings of the 9th Extended Semantic Web Conference (ESWC2012). LNCS, vol. 7295, pp. 345–359. Springer (2012)
- [5] Carral Martinez, D., Krisnadhi, A. A., Hitzler, P.: Integrating OWL and rules: A syntax proposal for nominal schemas. In Proceedings of Proceedings of OWL: Experiences and Directions Workshop, Heraklion, Crete, Greece, May 27-28 (2012)
- [6] Donini, F. M., Lenzerini, M., Nardi, D., Schaerf, A.: AL-log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems (JIIS)*. 10(3), 227–252 (1998)
- [7] Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining Answer Set Programming with Description Logics for the Semantic Web. In Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR’04), pp. 141–151. AAAI Press (2004)
- [8] Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: Effective Integration of Declarative Rules with External Evaluations for Semantic Web Reasoning. In Proceedings of the 3rd European Semantic Web Conference (ESWC2006). LNCS, vol. 4011, pp. 273–287. Springer (2006)
- [9] Eiter, T., Fink, M., Moura, J.: Paracoherent Answer Set Programming. In Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR’10), pp. 486–496. AAAI Press (2010)
- [10] Fitting, M.: *First-Order Logic and Automated Theorem Proving*, 2nd Edition. Texts in Computer Science. Springer (1996)
- [11] Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming. In Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP’88), pp. 1070–1080. MIT Press (1988)
- [12] Grosz, B. N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In Proceedings of the 12th International World Wide Web Conference (WWW2003), pp. 48–57. ACM (2003)

- [13] Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A Framework for Handling Inconsistency in Changing Ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A.(Eds.) Proceedings of the 4th International Semantic Web Conference (ISWC2005). LNCS, vol. 3729, pp. 353–367. Springer (2005)
- [14] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (Eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation 27 October 2009 (2009), available from <http://www.w3.org/TR/owl2-primer/>
- [15] Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
- [16] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004, available from <http://www.w3.org/Submission/SWRL/>
- [17] Huang, S., Li, Q., Hitzler, P.: Paraconsistent Semantics for Hybrid MKNF Knowledge Bases. In: Gutierrez, C., Rudolph, S. (Eds.) Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR’11). LNCS, vol. 6902, pp. 93–107. Springer (2011).
- [18] Hunter, A.: Reasoning with Contradictory Information Using Quasi-classical Logic. *Journal of Logic Computation*. 10 (5), 677–703 (2000)
- [19] Inoue, K., Koshimura, M., Hasegawa, R.: Embedding Negation as Failure into a Model Generation Theorem Prover. In Proceedings of the 11th International Conference on Automated Deduction. LNCS, vol. 607, pp. 400–415. Springer (1992)
- [20] Knorr, M., Alferes, J. J., Hitzler, P.: A Coherent Well-founded Model for Hybrid MKNF Knowledge Bases. In: Ghallab, M., Spyropoulos, C. D., Fakotakis, N., and Avouris, N. M.(Eds.) Proceedings of the 18th European Conference on Artificial Intelligence (ECAI2008). Vol. 178, pp. 99–103. IOS Press, Patras, Greece (2008)
- [21] Knorr, M., Alferes, J., Hitzler, P.: Local Closed-world Reasoning with Description Logics under the Well-founded Semantics. *Artificial Intelligence* 175(9–10), 1528–1554 (2011)

- [22] Knorr, M., Hitzler, P., Maier, F.: Reconciling OWL and Non-monotonic Rules for the Semantic Web. In Proceedings of the 20th European Conference on Artificial Intelligence (ECAI2012), pp. 474–479. IOS Press (2012)
- [23] Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In: Polleres, Al, d’Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P.F. (Eds.) Reasoning Web. Semantic Technologies for the Web of Data. 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures. LNCS, vol. 6848, pp. 382-415. Springer (2011)
- [24] Krötzsch, M., Maier, F., Krisnadhi, A. A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Sadagopan, S., Ramamritham, K., Kumar, A., Ravindra, M., Bertino, E., Kumar, R. (eds.) Proceedings of the 20th International World Wide Web Conference (WWW2011), Hyderabad, India, March/April 2011, pp. 645–654. ACM, New York (2011)
- [25] Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable Rules for OWL 2. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (Eds.) Proceedings of the 7th International Semantic Web Conference (ISWC2008). LNCS, vol. 5318, pp. 649–664. Springer (2008)
- [26] Levy, A. Y., Rousset, M.-C.: Combining Horn Rules and Description Logics in CARIN. *Artificial Intelligence*. 104(1-2), 165–209 (1998)
- [27] Lifschitz, V.: Nonmonotonic Databases and Epistemic Queries. In: Mylopoulos, J. and Reiter, R. (Eds.) Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), pp. 381–386. Morgan Kaufmann Publishers (1991)
- [28] Ma, Y., Hitzler, P., Lin, Z.: Algorithms for Paraconsistent Reasoning with OWL. In: Proceedings of the 4th European Semantic Web Conference (ESWC2007). LNCS, vol. 4519, pp. 399–413. Springer (2007)
- [29] Maier, F.: Extending Paraconsistent SHOIQ. In: Hitzler, P., and Lukasiewicz, T. (Eds.) Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR’09). LNCS, vol. 6333, pp. 118–132. Springer (2010)

- [30] Maier, F., Ma, Y., Hitzler, P.: Paraconsistent OWL and Related Logics. Semantic Web journal. DOI : 10.3233/SW-2012-0066, 2012.
- [31] Motik, B., Cuenca Grau, B., Horrocks I., Wu, Z., Fokoue, A., Lutz, C., Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (Eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation 27 October 2009 , available from <http://www.w3.org/TR/owl2-profiles/>
- [32] Motik, B., Rosati, R.: A Faithful Integration of Description Logic with Logic Programming. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), Hyderabad, India, January 6-12, 2007, pp. 477–482. AAAI Press (2007)
- [33] Motik, B., Rosati, R.: Reconciling Description Logics and Rules. Journal of the ACM, 57(5), 1–61 (2010)
- [34] Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with rules. Journal of Web Semantics. 3(1), 41–60 (2005)
- [35] Patel-Schneider, P. F.: A Four-valued Semantics for Terminological Logics. Artificial Intelligence. 38, 319–351 (1989)
- [36] Priest, G.: Logic of Paradox. Journal of Philosophical Logic. 8, 219–241 (1979)
- [37] Rosati, R.: DL+log: Tight Integration of Description Logics and Disjunctive Datalog. In Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06), pp. 68–78. AAAI Press (2006)
- [38] Sakama, C., Inoue, K.: Paraconsistent Stable Semantics for extended disjunctive programs. Journal of Logic and Computation. 5, 265–285. Oxford University Press (1995)
- [39] Schlobach, S., Cornet, R.: Non-standard Reasoning Services for the Debugging of Description Logic Terminologies. In: Gottlob, G., Walsh, T. (Eds.) Proceedings of the 18th International Conference on Artificial Intelligence (IJCAI-03), pp. 355–362. Morgan Kaufmann Publishers (2003)

- [40] Van Gelder, A., Ross, K., Schlipf, J. S.: The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*. 38 (3), 620–650 (1991)
- [41] Zhang, X., Lin, Z.: Paraconsistent Reasoning with Quasi-classical Semantic in \mathcal{ALC} . In: Calvanese, D., Lausen, G.(eds.) *Proceedings of 2th International Conference on Web Reasoning and Rule Systems (RR 2008)*. LNCS, vol. 5341, pp. 222–229. Springer (2008)
- [42] Zhang, X., Lin, Z., Wang, K.: Towards a Paradoxical Description Logic for the Semantic Web. In: Link, S., Prade, H. (eds.): *Proceedings of the 6th International Symposium on Foundations of Information and Knowledge Systems (FoIKS2010)*. LNCS, vol. 5956, pp. 306–326 (2010)