Tobias Kaminski

The European Master's Program in Computational Logic
Master's Thesis

Efficient Paraconsistent Reasoning with Rules and Ontologies for the Semantic Web

Dissertação para obtenção do Grau de
Mestre em Lógica Computacional

Orientadores: João Leite,
CENTRIA, Universidade Nova de Lisboa
Matthias Knorr,
CENTRIA, Universidade Nova de Lisboa

Júri:

Presidente: Josẽ Jũlio Alferes

Arguentes: Michael Fink
Enrico Franconi
Matthias Knorr
João Leite

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2014

Efficient Paraconsistent Reasoning
with Rules and Ontologies
for the Semantic Web

Acknowledgements

I would like to thank all the people who gave me the possibility to complete this master's thesis. Firstly, I would like to express my highest gratitude for the guidance I received by my supervisors, João Leite and Matthias Knorr, and for many helpful comments which greatly helped me in developing and improving the work presented in this thesis. Further, I would like to thank the Joint Commission of the European Master's Program in Computational Logic for giving me the chance to study in this program, giving me financial support and providing excellent conditions for studying. Finally, I would like to thank my family and all my friends for their great support and encouragement.

Abstract

Ontologies formalized by means of Description Logics (DLs) and rules in the form of Logic Programs (LPs) are two prominent formalisms in the field of Knowledge Representation and Reasoning. While DLs adhere to the *Open World Assumption* and are suited for taxonomic reasoning, LPs implement reasoning under the *Closed World Assumption*, so that default knowledge can be expressed. However, for many applications it is useful to have a means that allows reasoning over an open domain and expressing rules with exceptions at the same time. Hybrid MKNF knowledge bases make such a means available by formalizing DLs and LPs in a common logic, the Logic of Minimal Knowledge and Negation as Failure (MKNF).

Since rules and ontologies are used in open environments such as the Semantic Web, inconsistencies cannot always be avoided. This poses a problem due to the *Principle of Explosion*, which holds in classical logics. Paraconsistent Logics offer a solution to this issue by assigning meaningful models even to contradictory sets of formulas. Consequently, paraconsistent semantics for DLs and LPs have been investigated intensively. Our goal is to apply the paraconsistent approach to the combination of DLs and LPs in hybrid MKNF knowledge bases.

In this thesis, a new six-valued semantics for hybrid MKNF knowledge bases is introduced, extending the three-valued approach by Knorr et al., which is based on the well-founded semantics for logic programs. Additionally, a procedural way of computing paraconsistent well-founded models for hybrid MKNF knowledge bases by means of an alternating fixpoint construction is presented and it is proven that the algorithm is sound and complete w.r.t. the model-theoretic characterization of the semantics. Moreover, it is shown that the new semantics is faithful w.r.t. well-studied paraconsistent semantics for DLs and LPs, respectively, and maintains the efficiency of the approach it extends.

Contents

List of Definitions

List of Propositions and Theorems

1

Introduction

For several decades, the development of expressive formalisms for knowledge representation, together with efficient reasoning services, has attracted researchers due to their ability of formalizing knowledge in subtle ways and to infer new knowledge from large amounts of data. In recent years, research on these topics has gained momentum partly because of its great promises for applications in *Semantic Web Technologies*. One vision of today's research in *Artificial Intelligence* (*AI*) is that it will be possible for computers to pose queries to and reason with data stored on the *World Wide Web*, so that the Web will essentially represent one large knowledge base. For this purpose, it is necessary to assign a formal semantics to data stored on the Web, which can be processed by computer programs. Such formalisms are commonly referred to by the notion of the emerging *Semantic Web*, and they are standardized by the *World Wide Web Consortium* (*W3C*).

The discipline of *Knowledge Representation and Reasoning* (*KRR*) is an important field of AI since it not only provides formalisms to represent data in a computer, but also mechanisms to obtain information that is only implicitly entailed by the data. *Description Logics* (*DLs*) and *Logic Programs* (*LPs*) are two major knowledge representation formalisms that have been investigated intensively in the past. As they are suited for reasoning in the Semantic Web, they have been respectively standardized by the W3C in the form of the ontology languages jointly named *Web Ontology Language* (*OWL*) and the *Rule Interchange Format* (*RIF*). DLs and LPs have different strengths and which of them is used as the logic for an information system depends on the context, i.e. the nature of the knowledge that has to be represented and the reasoning mechanisms that are required. On the one hand, DLs are appropriate for defining concepts and hierarchies among them. On the other hand, LPs are able to express rules with possible exceptions and preference orders over sets of rules. Consequently, the abilities of DLs and LPs can be viewed as being

complementary such that combining the two formalisms is an obvious next step in the advancement of KRR. Moreover, there are many real-world applications that require both of the described kinds of reasoning at the same time. For example, this could be the case for the customs' information system at a port that has to classify goods and execute according actions automatically (cf. Example 1.1)[1]. The endeavor of fully integrating DLs and LPs is realized by the development of so-called *hybrid knowledge bases*, which define an encompassing framework in which DLs as well as LPs can be formalized concurrently.

In the development of hybrid knowledge bases several challenges have to be faced, where some of them carry over from the separate formalisms and some new challenges emerge due to their integration. Firstly, it is desirable that a hybrid knowledge base is faithful w.r.t. the separate formalisms it integrates, i.e. it should behave as expected by a knowledge base designer who is acquainted with DLs and/or LPs. Simultaneously, the integration of DLs and LPs in a hybrid knowledge base should be as tight as possible, so that the trade-off of choosing between the two formalisms vanishes. Secondly, efficient algorithms are required for reasoning with hybrid knowledge bases since ontologies and rule bases may contain large amounts of data, which obviously also holds for their combination. This issue is even more crucial when considering the gigantic amount of data contained in the Web being viewed as one big knowledge base. Finally, a critical problem consists in the emergence of inconsistencies in hybrid knowledge bases due to the integration of different data sources, the occurrence of errors, or the participation of several users with potentially different views on the domain. Thus, contradictions are often unavoidable, especially when applications in an open environment such as the Semantic Web are taken into account, which contain very large amounts of interconnected data. Moreover, there can be complex interactions between the ontology component and the rule component of a hybrid knowledge base that can easily lead to contradictions. Under classical semantics, knowledge bases comprising contradictory information are no longer meaningful for reasoning since everything can be derived from them. Consequently, some strategy has to be implemented in a hybrid knowledge base to address the challenges posed by inconsistent knowledge bases, an undertaking that has only been addressed very rarely in the literature on hybrid knowledge bases so far. This thesis is about building a formalism for hybrid knowledge bases that fares well in all of these dimensions.

1.1 Hybrid Knowledge Bases: Bringing Two Worlds Together

As mentioned, DLs and LPs are two formalisms that approach the problem of representing and reasoning about knowledge from different directions, and exhibit different properties. DLs constitute decidable fragments of first-order logic. The constructs that can be used in a respective DL language are restricted, so that a certain computational

[1]For another concrete example of a real-world application that requires the expressive power of both DLs and LPs, and considers a medical knowledge base containing data about patients, refer to [KAH11].

complexity can be guaranteed for reasoning tasks such as *satisfiability* or *subsumption*. On the other hand, LPs consist of implications providing a rule-like structure and their semantics is usually defined over a finite domain. As a result, it is possible to work with the ground version of a logic program that contains variables.

A fundamental difference[2] between DLs and LPs lies in the fact that DLs employ the *Open World Assumption* (*OWA*), whereas LPs work under the *Closed World Assumption* (*CWA*) [KAH11]. While under the OWA everything that cannot be derived is considered to be unknown, the CWA implies that everything which is not known to be *true* is *false* by default. The adherence to each of these assumptions can be justified depending on the context and the kind of information system at hand. On the one hand, the OWA is useful when a knowledge base contains incomplete information such that missing information does not imply its falsity. On the other hand, under the CWA a *default negation operator*, usually denoted by **not**, can be introduced, which enables *non-monotonic reasoning* in LPs, i.e. knowledge derivable from an LP might not be derivable anymore after extending the program. The default negation operator is also called an *introspective operator* as it can introspect the LP to derive some fact based on the absence of information. On the contrary, there is normally no introspective operator in DLs and reasoning in them is monotonic[3]. However, adopting the OWA allows DLs to define concepts by quantifying over items in their domain which do not have to be explicitly known. Additionally, the *classical negation operator* $\neg$ is an integral constituent of DLs since they are based on first-order logic, so that it is usually possible to prove that some fact does not hold in them. Classical negation is a stronger form of negation than default negation since if some piece of knowledge cannot be proven to be *true*, its classical complement still cannot be assumed. Hybrid knowledge bases bring these two worlds together by enabling reasoning under the OWA and the CWA at the same time in different parts of a knowledge base. They normally consist of a *structural part* in form of an *ontology component* and a *rule-set component* sharing a common vocabulary that connects them.

The advantages of applying the OWA or the CWA respectively and the benefits of being able to combine reasoning under both of them in a hybrid knowledge base are demonstrated by means of the following example, which will be used as a running example throughout this thesis. The example is inspired by a realistic example presented by Slota et al. [SLS11], which in turn is inspired by a real-world use-case of hybrid knowledge bases. Here, the convention from Knorr et al. [KAH11] to start all predicate names occurring in the ontology component with upper-case letters, and those only appearing in the program component with lower-case letters, is adopted.

Example 1.1 (Customs risk assessment)**.** Consider the following hybrid knowledge base, which contains an ontology component formalized in DL-syntax in the upper part and a

[2] A discussion of further differences in the expressive power of DLs and LPs can be found in [EIKP08].

[3] This is true at least in the case of standard DL semantics. Non-monotonic extensions of DLs have been developed as well (e.g. in [DLNNS98; GGOP13]). However, they are not considered in this thesis since we achieve the non-monotonic behavior of the formalism developed here by providing an encompassing semantics for DLs and LPs.

logic program in the lower part. This hybrid knowledge base could be part of an information system used by a customs service to assess the risk exhibited by a certain good as well as to derive according actions that have to be taken on the basis of the classification of a good by the system. Note that this is in fact a hybrid knowledge base since the DL- and the LP-part share a common vocabulary and hence, interactions can occur between the two components.

$$
\begin{aligned}
\mathsf{ToxicChemical} &\sqsubseteq \exists\mathsf{Contains.ToxicSubstance} \\
\exists\mathsf{Contains.ToxicSubstance} &\sqsubseteq \mathsf{ProvenRisk} \\
\mathsf{ProvenRisk} \sqcup \mathsf{PotentialRisk} &\sqsubseteq \mathsf{Risk} \\
\mathsf{HasCertifiedForwarder} &\sqsubseteq \neg\mathsf{IsMonitored} \\
\mathsf{ToxicChemical(pesticide)} &
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{IsMonitored(x)} &\leftarrow \mathsf{good(x)}, \mathsf{Risk(x)}. \\
\mathsf{PotentialRisk(x)} &\leftarrow \mathsf{good(x)}, \mathbf{not}\,\mathsf{isLabelled(x)}. \\
\mathsf{resolvedRisk(x)} &\leftarrow \mathsf{good(x)}, \mathsf{IsMonitored(x)}. \\
\mathsf{good(food)} &\leftarrow \\
\mathsf{good(pesticide)} &\leftarrow \\
\mathsf{isLabelled(pesticide)} &\leftarrow
\end{aligned}
$$

Using this knowledge base, the good food can be derived to be a PotentialRisk by means of the second rule in the program, and the good Pesticide can be classified as a ProvenRisk due to the first and the second ontology axiom. A PotentialRisk and a ProvenRisk are both a kind of Risk according to the third ontology axiom. The difference between a PotentialRisk and a ProvenRisk is that a good can be derived to be a PotentialRisk by means of default reasoning, so that every good for which it is not known that it isLabelled is derived to be a PotentialRisk. Note that this can only be expressed because the program is interpreted under the CWA.

On the other side, a good can be derived to be a ProvenRisk by means of the ontology component if it Contains some ToxicSubstance. Note that, in contrast to the program component, not every individual referred to in the ontology component has to be explicitly stated. According to the second ontology axiom, everything that Contains some ToxicSubstance can be derived to be a ProvenRisk, even if the specific ToxicSubstance is unknown in this case. This is a result of the fact that the OWA is adopted by the ontology component.

Moreover, by means of the ontology component it is possible to prove that it does not hold for some good that it IsMonitored since classical negation is used in the fourth axiom. This can be important if the customs service has to be sure that it does not hold

for some good that it IsMonitored. Additionally, a reasoning algorithm can even provide an explanation for the falsity of IsMonitored in this case, namely that the good satisfies the concept description of HasCertifiedForwarder.

Further, note that the ontology and program component are tightly coupled in this example, so that it can first be derived by involving default negation that some good is a PotentialRisk, afterwards it can be deduced from the third axiom that the good is a Risk, and on the basis of this deduction an according rule can be applied which triggers that the particular good IsMonitored. ◊

1.2 Inconsistencies in Hybrid Knowledge Bases

In the hybrid knowledge base introduced in Example 1.1, inconsistencies can easily arise. For instance, assume that we add HasCertifiedForwarder(food) either as a fact to the program component or as a concept assertion to the ontology component. In this case, ¬isMonitored(food) can be derived by the fourth ontology axiom. But so can isMonitored(food) with the help of the first rule. As a result, the hybrid knowledge base does not have a model under a classical semantics, where only the truth values *true* and *false* are available, since assigning each of them to the atom isMonitored(food) does not satisfy the knowledge base. This poses a problem as the knowledge base is trivialized in this way, which means that everything is a logical consequence of it. This is due to the *Principle of Explosion*, which holds in classical logics and can be characterized in the following way.

Definition 1.2 (The Principle of Explosion)**.** Let ψ be a first-order formula and Φ a set of first-order formulas. If it is the case that $\Phi \models \psi$ and $\Phi \models \neg\psi$, then it holds that $\Phi \models \xi$ for every first-order formula ξ, where the operator $\models$ denotes the classical entailment operator of first-order logic.

The described behavior of hybrid knowledge bases under classical semantics is highly undesirable and renders the knowledge base useless if it contains contradictory information. To phrase it bluntly, one does not want that the gift one sends to a friend is classified as a toxic chemical by the customs' information system because a certified forwarder has forgotten to label some product. Furthermore, even an inconsistent knowledge base usually contains meaningful information that can be derived by logical rules not involving any contradictory knowledge. For example, after adding the program fact HasCertifiedForwarder(food) to the knowledge base in Example 1.1, Risk(Pesticide) is still derivable from the consistent part of the knowledge base alone. However, under classical semantics the whole knowledge base is spoiled by the inconsistency.

According to Damásio and Pereira [DP95], there are two main strategies for dealing with the problem of the explosive behavior observed in inconsistent knowledge bases. Which of the strategies is used partly depends on the viewpoint that is adopted regarding contradictions.

1. One can take the view that there are no true contradictions in the world and that contradictory information in a knowledge base always constitutes some kind of error, such that the knowledge base has to be "debugged". This is the goal of approaches in the areas of *Belief Revision* and *Ontology Repair*, which have been developed to remove contradictions from a knowledge base.

2. Contradictions can also be regarded as natural phenomena in realistic data, so that a logic should account for them. For this purpose, *Paraconsistent Logics* have been developed, in which the *Principle of Explosion* does not hold (or only partly holds).

Many approaches to belief revision have been developed in order to avoid inconsistencies when information is integrated or updated, often based on seminal work done by Alchourrón et al. [AGM85], who published the famous *AGM postulates* for belief revision. However, even how to revise DLs and LPs separately is a debated research question and revision algorithms with tractable computational complexity are difficult to obtain, even in the case of propositional knowledge bases (cf. [EG92]). In spite of the great challenges faced by revision approaches, work on revising combined formalisms for rules and ontologies has already been conducted (e.g. in [EFS13; SLS11]). On the other hand, reasoning with paraconsistent logics can be reduced to reasoning with its classical counterparts in many cases, where a tractable reasoning algorithm can often be obtained in this way. Besides, contradictions first have to be detected also for applying belief revision techniques, so that paraconsistent logics can be viewed as an intermediary step towards belief revision, according to Damásio [Dam96]. In our work, we will follow the secondly mentioned strategy for dealing with inconsistencies by providing a paraconsistent semantics for a certain hybrid knowledge base formalism.

In addition to the capability of detecting and reasoning with *inconsistent* knowledge bases provided by a paraconsistent semantics, it is often desirable that a paraconsistent formalism also supplies information about which facts can only be derived by involving contradictory information. For instance, suppose that, after adding the fact HasCertifiedForwarder(food) to the hybrid knowledge base in Example 1.1, we want to query the knowledge base for all goods that are a resolvedRisk. Under a paraconsistent semantics, resolvedRisk(food) can be derived. However, this derivation depends on the contradictory fact IsMonitored(food), i.e. since HasCertifiedForwarder(food) is *true*, ¬IsMonitored(food) can be derived as well and hence, we cannot be completely sure that IsMonitored(food) really holds. Therefore, we would still like to get the answer resolvedRisk(food) to our query, but together with the information that we have to be careful in using this fact. This information can be attained by "propagating" contradictions in some way in the inference steps executed to obtain the respective fact. In this way, it is possible to "encapsulate" the contradictory fragment of the information entailed by a knowledge base while still being able to use it for reasoning, so that no knowledge is lost by following this strategy (in contrast to belief revision approaches). Damásio and Pereira term the described technique *Contradiction Support Detection* [DP97].

1.3 Contributions and Outline

The work presented in this thesis builds upon a formalism for hybrid knowledge bases that has been developed by Knorr et al. [KAH08; KAH11], which exhibits a number of desirable properties and has already been implemented in the *NoHR*-plug-in for the ontology editor *Protégé* [IKL13]. The authors define a three-valued well-founded semantics for hybrid knowledge bases formalized in the *Logic of Minimal Knowledge and Negation as Failure* (*MKNF*). *Hybrid MKNF knowledge bases* constitute a *tight* and *flexible* integration of DLs and LPs, which means that both components interact in terms of derivable consequences and that it is possible to define predicates in the ontology and the program component simultaneously [MR10]. By basing their approach on the *Well-Founded Semantics* for logic programs, Knorr et al. are able to define a procedural characterization of their semantics, which uses an alternating fixpoint construction and provides an efficient means to compute the semantics of hybrid MKNF knowledge bases. Furthermore, their framework is faithful w.r.t. classical DL semantics and the original Well-Founded Semantics [KAH11]. Additionally, the three-valued semantics is sound w.r.t the previously developed two-valued semantics for hybrid MKNF knowledge bases, which is based on the *Stable Model Semantics* [MR10].

Although inconsistencies could already be detected in the framework of Knorr et al., their semantics is only defined for consistent hybrid MKNF knowledge bases and developing a paraconsistent Well-Founded Semantics for these knowledge bases has been an open problem so far. In this thesis, we present a paraconsistent version of the three-valued semantics for hybrid MKNF knowledge bases of Knorr et al., which allows full paraconsistent reasoning when an inconsistency occurs as a result of the interaction between the ontology and the program component, as well as when the ontology is inconsistent by itself. The contributions of this thesis are the following:

1. Our main contribution is the introduction of a new six-valued MKNF semantics, which is robust w.r.t. inconsistencies, implements Contradiction Support Detection regarding contradictions occurring in the program component, and differentiates between facts which are *false* by default and those which are *classically false*.

2. In order to compute paraconsistent models of our semantics, we adapt the alternating fixpoint construction used by Knorr et al. [KAH11] and show that the resulting algorithm is sound and complete w.r.t. the model-theoretic characterization of our semantics.

3. For ontology components formalized in the syntax of the DL $\mathcal{ALC}$, we prove that our semantics coincides with the paraconsistent DL semantics $\mathcal{ALC}4$ published by Maier et al. [MMH13] in case the program component is empty.

4. We show that if the ontology component of a hybrid MKNF knowledge base is restricted, our semantics corresponds to the semantics of $WFSX_p$ [Dam96] (in case

classical negation is only used in front of unary program atoms). As a result, our approach entails a new and concise model-theoretic characterization of $WFSX_p$.

5. We prove that for consistent hybrid MKNF knowledge bases, our semantics is sound w.r.t. the three-valued semantics of Knorr et al., and by transitivity also w.r.t. the two-valued semantics of Motik and Rosati [MR10].

6. Finally, we show that the efficiency of the three-valued approach developed by Knorr et al. carries over to our framework, so that reasoning with our semantics is tractable in case reasoning in the DL used for formalizing the ontology component is tractable.

The remainder of this thesis is structured as follows. The next chapter provides the background on paraconsistent logics for different knowledge representation formalisms and previous approaches to hybrid knowledge bases. In Chapter 3, the model-theory of our six-valued logic is developed and the faithfulness results regarding $\mathcal{ALC}4$ and the three-valued semantics of Knorr et al. are established. Chapter 4 describes the adaptation of the alternating fixpoint construction used by Knorr et al. to our approach and provides the proofs showing the correspondence to the model-theoretic characterization of our logic. Furthermore, the faithfulness result w.r.t. $WFSX_p$ is presented and the complexity of our approach is discussed in the end of this chapter. Finally, we conclude in Chapter 5 by reviewing two related approaches and discussing some ideas for future extensions of our framework.

2

Background

As this thesis integrates many different sub-fields of KRR, we start by providing a broad overview over the different formalisms developed in these sub-fields and their connections to our work. In this way, we place our work in the context of paraconsistent logics, their application to DLs and LPs, and previous approaches on hybrid knowledge bases. Moreover, some basic notions and definitions will be introduced in this chapter, which later chapters will draw on.

2.1 Paraconsistent Logics

As mentioned in the introduction, *paraconsistent logics* are those logics which still allow the derivation of meaningful knowledge from contradictory sets of formulas. According to Lang [Lan06], a logic is *paraconsistent* if there are sets of formulas of the logic that entail some formula and its classical negation and still have a model in the logic, and it is *fully paraconsistent* if the previous holds for all sets of formulas of the logic. Consequently, the *Principle of Explosion* is either partly or fully rejected in a paraconsistent logic. Generally, paraconsistent logics restrict the amount of consequences that can be derived from a set of formulas, so that they can be viewed to implement a "more conservative" form of reasoning, as noted by Carnielli and Marcos [CM01].

Paraconsistent logics that are used as the foundation for information systems, such as databases, and for knowledge representation formalisms in general, often employ some many-valued logic to obtain the paraconsistent behavior [GS00], such that in addition to the classical truth values *true* and *false*, at least one further truth value is introduced. Though, besides many-valued logics, a large number of other approaches to paraconsistent reasoning have been developed in the fields of philosophical and mathematical

logic, such as Jaśkowski's *Discussive Logic* [CD95; Jaś69], da Costa's *C-Logics* [Cos74] and Baten's *Inconsistency-Adaptive Logics* [Bat07]. For a comprehensive overview over these and other paraconsistent approaches, the reader is referred to the surveys by da Costa et al. [DCKB07] and Middelburg [Mid11].

In order to see how introducing an additional truth value can resolve the problem exemplified in section 1.2, recall that the facts isMonitored(food) and ¬isMonitored(food) can both be derived from the hybrid knowledge base in Example 1.1 after adding the fact HasCertifiedForwarder(food). Consequently, it is not possible to model the knowledge base under a two-valued semantics as assigning the truth value *true* to the proposition isMonitored(food) would not satisfy ¬isMonitored(food) and assigning the truth value *false* would not satisfy isMonitored(food) itself. Now, by having a third truth value available which is a designated truth value[1] of the logic and at the same time is its own complement, this third truth value can be assigned to the proposition isMonitored(food), which results in the knowledge base having a model[2]. As a result, not every proposition is entailed by the knowledge base under such a many-valued semantics anymore, but only those propositions that are satisfied by all models of the knowledge base that now interpret contradictory facts with the new truth value. This is to say, e.g. resolvedRisk(food) is still entailed by the inconsistent hybrid knowledge base, but ToxicChemical(food) cannot be derived anymore by means of the *Principle of Explosion*, just as it is the case for the consistent version of the hybrid knowledge base in Example 1.1. Basically, it is possible to isolate the inconsistency in this manner.

2.1.1 Belnap's Four-Valued Logic

As Maier et al. state [MMH13], the two-valued nature of classical logics ensures that the following two principles are satisfied in them:

- *The Law of the Excluded Middle* (*LEM*): every proposition must be either *true* or *false*. This principle is also called *tertium non datur*, which is Latin for "no third is given".
- *The Law of Non-Contradiction* (*LNC*): no proposition may be *true* and *false* at the same time. This leads to contradictory propositions not having a model and thus, implies the *Principle of Explosion* introduced in section 1.2, of which the principle *ex falso sequitur quodlibet* is the Latin equivalent.

By introducing further truth values into the semantics of a logic, a logic that does not comply with either one or both of these laws can be obtained. For instance, as Maier et al. remark [MMH13], the strong three-valued logic K_3 [Fit91b] devised by Kleene follows the *LNC* but not the *LEM*. On the other hand, Priest's three-valued *Logic of Paradox* [Pri79] obeys the *LEM*, but not the *LNC*.

[1] A designated truth value of a logic is a truth value such that if a formula is evaluated to this truth value in an interpretation, the interpretation is said to model the formula [Got14].

[2] Obviously, this also depends on the interpretation functions of the other operators w.r.t. the new truth value, which will be discussed in more detail later.

A logic which rejects both of the aforementioned principles is Belnap's four-valued logic [BJ77a]. The use of many-valued logics as a "practical tool for inference" [BJ77a] in computer science can be traced back to his article entitled "How a computer should think" and published in 1977 [BJ77b]. Accordingly, most of the work on paraconsistent semantics for logic programs and ontologies discussed in this thesis take Belnap's four-valued logic as a starting point for their approach. Belnap motivates the four truth values of his logic by pointing out that a computer can be in one of four epistemic states w.r.t. a proposition, depending on the information it has been told. It could either have been told that the proposition is *true*, that it is *false*, none of the previous, or both. While classically only the first two of these epistemic states are regarded, the latter two cases occur very often in real-world applications since a computer can be provided with contradictory and/or incomplete information. Belnap's insight was to take this issue seriously by implementing all of the four possible epistemic states in what he calls a "sophisticated question-answering system". Belnap denotes the four truth values of his logic by the symbols **T**, **F**, **None** and **Both**. We refer to them by the names *true*, *false*, *undefined* and *inconsistent*, and use the symbols **t**, **f**, **u** and **b**, respectively.

Belnap makes the observation that the four truth values of his logic naturally can be ordered into two different complete lattices[3], which he terms the *approximation lattice* $\mathbf{A4}$ and the *logical lattice* $\mathbf{L4}$, respectively. According to Belnap, the partial order induced by the approximation lattice can be read as "approximates the information in", in the sense that truth values that are greater in the order exhibit more information about the truth of a proposition [AA00]. Furthermore, he defines logical disjunction of two truth values as their least upper bound and conjunction as their greatest lower bound in the lattice $\mathbf{L4}$ [BJ77a].

Later, Belnap's four-valued logic has been generalized to so-called *bilattices* by Ginsberg [Gin88], which are algebraic structures simultaneously containing two partial orders [AA96]. In this way, the two lattices $\mathbf{A4}$ and $\mathbf{L4}$ introduced by Belnap can be combined into a single bilattice (which is often denoted by $\mathcal{FOUR}$ in the literature) with two partial orders, the *truth order* $\leq_t$ and the *knowledge order* $\leq_k$. On the basis of Ginsberg's work, Arieli and Avron give the following definition of a bilattice [AA00].

Definition 2.1 (Bilattice [AA00])**.** A *bilattice* is a structure $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ such that B is a non-empty set containing at least two elements; $(B, \leq_t)$ and $(B, \leq_k)$ are complement lattices; and $\neg$ is a unary operator on B that has the following properties:

(1) If $a \leq_t b$, then $\neg a \geq_t \neg b$,

(2) if $a \leq_k b$, then $\neg a \leq_k \neg b$, and

(3) $\neg\neg a = a$.

[3] A complete lattice consists of a set S and a partial order $\leq$, such that for every set $S' \subseteq S$ there is a least upper bound and a greatest lower bound in S [BJ77a].

According to Fitting [Fit91a], Belnap's logic constitutes the simplest logic that can be represented by means of a bilattice. Here, the truth order $\leq_t$ in the bilattice $\mathcal{FOUR}$ corresponds to the order in Belnap's lattice **L4** indicating the "degree of truth", and the knowledge $\leq_k$ order corresponds to the partial order in the lattice **A4** signifying a "measure of knowledge about truth" [AA00]. The bilattice $\mathcal{FOUR}$ together with the two orders is shown in Figure 2.1.

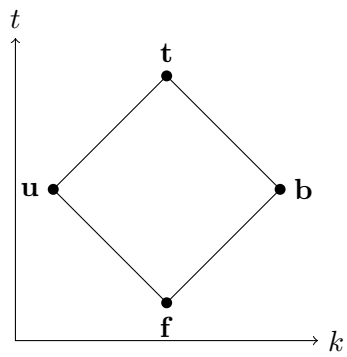

Figure 2.1: Bilattice $\mathcal{FOUR}$ of Belnap's four-valued logic [BJ77a], together with the truth- and knowledge order introduced by Ginsberg [Gin88].

Building on the work by Ginsberg, Fitting has applied the notion of bilattices to logic programming [Fit90; Fit91a], and his results have inspired several paraconsistent approaches in the field (e.g. [ADP02; Dam96; SI95]). For this, he narrows the definition of bilattices to *interlaced bilattices*, where an interlaced bilattice has to satisfy the additional condition that the join and meet operation of one partial order have to be monotone w.r.t. the join and meet operation of the other partial order [Fit91a]. Fitting denotes the join and meet operation w.r.t. the knowledge order $\leq_k$ by $\otimes$ and $\oplus$, and calls them *consensus* and *gullability* operator respectively since "$x \otimes y$ is the most information that x and y agree on" and $x \oplus y$ "accepts anything" that is stated by x or y [Fit90]. In addition to the negation operator $\neg$ of a bilattice, Fitting introduced a *conflation operator* which behaves like negation, "but with the roles of $\leq_k$ and $\leq_t$ switched around" [Fit91a]. Just like negation reverses truth and leaves the degree of knowledge unchanged, conflation reverses the degree of knowledge but does not change truth, according to Fitting. Alcântara et al. show that the *default negation* operator **not**, which is used in logic programs and will be discussed in more detail in section 2.3, can be defined in terms of a negation operator $\neg$ and a conflation operator $-$, namely by equating **not** A with $-\neg A$, given some atom A. Intuitively, this definition expresses that the default negation of a proposition has the meaning that there is no evidence (or it is not believed) that the proposition is *true* [ADP02].

2.2 Paraconsistent Description Logics

As discussed in the introduction in Chapter 1, even if we work under the assumption that the world itself is always consistent, inconsistencies can easily arise in knowledge bases containing facts about the world, due to a large number of pieces of information and interactions between them. This is even more true when considering DL knowledge bases formalized in OWL and intended for the use in the Semantic Web. In this case, knowledge bases are not only typically very large themselves, but also often contain data from a large number of sources that is constantly changing. For instance, this is the case when different data sources are integrated or when many users with possibly different views on the respective domain collaborate in editing a knowledge base. Contradictions cannot always be resolved in this scenario, therefore some way of inconsistency detection and reasoning from contradictory knowledge is essential for Semantic Web Technologies.

Naturally, approaches tackling the pressing issue posed by inconsistent ontologies have attracted a lot of attention recently. They often handle inconsistent knowledge bases by resorting to some sort of paraconsistent logic. In this line of research, e.g. Lang proposes paraconsistent semantics for the DLs $\mathcal{ALC}$ and $\mathcal{SHIQ}$ on the basis of a four-valued version of first-order logic that is translatable into classical first-order logic, and defines a semantics for the implication operator such that the *Deduction Theorem* holds and formulas can be reduced to *Negation Normal Form* [Lan06]. Moreover, Zhang et al. have developed paraconsistent semantics and tableau algorithms for several DLs on the basis of quasi-classical logic[4] [ZL08; ZQML09; ZXL09], which fulfill several important inference principles like *Modus Ponens* and *Disjunctive Syllogism*, and correspond to the standard semantics in the case of consistent formulas [ZQML09]. The same authors have also published a three-valued approach [ZLW10] that approximates classical reasoning more closely than many four-valued approaches, and extends the DL $\mathcal{ALC}$ with the semantics of Priest's *Logic of Paradox* [Pri79]. In the *Logic of Paradox*, a formula is allowed to be *true* and *false* at the same time, but cannot be neither *true* nor *false*, so that the LEM is enforced.

To the best of our knowledge, the most recent publication in the area of paraconsistent DL semantics has been made by Maier et al. in 2013 [MMH13], which is largely based on earlier work by the authors (cf. [MH09; MHL07; Mai10]). In this publication, a paraconsistent semantics for the expressive DL $\mathcal{SROIQ}$ and all less expressive DLs which are subsumed by $\mathcal{SROIQ}$, such as $\mathcal{SHIQ}$ and $\mathcal{ALC}$, is presented. Like many other approaches that define a paraconsistent DL semantics, the semantics of Maier et al. is based on Belnap's logic $\mathcal{FOUR}$, and the authors fittingly label the respective four-valued fragments $\mathcal{SROIQ}4$, $\mathcal{SHIQ}4$ and $\mathcal{ALC}4$. They show that $\mathcal{SROIQ}4$ is sound w.r.t. the classical two-valued semantics and that a $\mathcal{SROIQ}4$-knowledge base can be embedded into $\mathcal{SROIQ}$ such that consequences are preserved. Furthermore, Maier et al. demonstrate

[4] "[Quasi-classical logic] is a form of paraconsistent logic with a more expressive semantics than Belnap's four-valued logic, and unlike other paraconsistent logics, allows the connectives to appear to behave as classical connectives" [Hun02].

that models can be forced to be three-valued, excluding either the truth value *paraconsistent* or the truth value *undefined* from the possible evaluations of a formula, by adding additional axioms to the knowledge base. In the latter case, a "more classical" behavior of the logic can be achieved according to the authors. This strategy to obtain a deductively stronger logic is also pursued by Zhang et al. in using the *Logic of Paradox* as the basis for their three-valued framework [ZLW10].

In all of the previously mentioned approaches, the authors direct their attention to maintaining inference rules and classical equivalences that hold in the standard version of the respective formalism, in order to approximate standard reasoning in a paraconsistent setting. The amount of inference rules and classical equivalences that can be preserved in a paraconsistent logic depends a lot on the semantics of the implication operator. Correspondingly, Maier et al. discuss three distinct kinds of implication within their four-valued framework, which enable different inference rules to be applied [MMH13]. These implication operators are *internal implication* ($\supset$), which is *true* whenever the consequent is *true* or *inconsistent* and has the truth value of the consequent otherwise, *strong implication* ($\rightarrow$), which is defined in terms of $\supset$, and *equivalence* ($\leftrightarrow$). While *Modus Ponens*, the *Deduction Theorem*, *Identity*, and *Supraclassicality* can be used when *internal implication* is employed, the *strong implication* operator additionally enables the application of *Modus Tollens*, *Transposition* and *Strong Equivalence*, but not the *Deduction Theorem* (for the definition of the particular inference rules refer to [MMH13]). The authors also show that whenever an implication operator in the four-valued framework satisfies *Modus Ponens* and the *Deduction Theorem*, then it cannot satisfy several of the other inference rules, such as *Modus Tollens*, at the same time. Although ontology semantics with a stronger inference relation can be obtained by using other forms of implication, if only internal implication is used in the paraconsistent version of certain tractable DLs such as $\mathcal{EL}^{++}$, then formulas can be mapped to a sets of formulas interpreted under classical semantics, such that consequences are preserved, as well as tractability [MMH13]. This is a desirable property for applications of paraconsistent DL semantics since standard reasoners can be used in this case.

Next, we introduce description logics formally, by defining their syntax and their classical as well as four-valued semantics as published by Maier et al. We restrict the presentation of the DL semantics here to the case of $\mathcal{ALC}/\mathcal{ALC}4$ for the sake of a more concise presentation and to avoid special cases where the ontology has no model due to a combination of nominals and cardinality restrictions. For example, assertions such as $\geq (n{+}1)r.\{a_1, ..., a_n\}(b)$ do not have a model under the semantics of $\mathcal{SROIQ}4$ [MMH13]. Accordingly, Maier et al. state that $\mathcal{SROIQ}4$ could be viewed as being only "partially paraconsistent". Furthermore, we prove the correspondence of the semantics we assign to the ontology component in the hybrid approach developed in this thesis w.r.t. ontologies expressed in the syntax of $\mathcal{ALC}$.

2.2.1 Syntax and Semantics of $\mathcal{ALC}$ and $\mathcal{ALC}4$

The syntax of $\mathcal{ALC}$ and $\mathcal{ALC}4$ is identical and it is based on three disjoint sets of names.

- The set N_I contains all *individual names* denoting atomic entities in the domain.
- The set N_C contains all *concept names* denoting classes of individuals that share a common property.
- The set N_R contains all *role names* denoting binary relations between individuals.

Hereafter, we will assume that the three sets of names, also called the *signature* of a description logic, are given when speaking about the syntax of an ontology. Complex *concept descriptions* can be constructed inductively, starting from atomic concepts. First, all *atomic concepts* in N_C, the *bottom concept* $\bot$ and the *top concept* $\top$ are concept descriptions. If C, C_1 and C_2 are concept descriptions, then the *negation* $\neg C$, the *intersection* $C_1 \sqcap C_2$ and the *union* $C_1 \sqcup C_2$ are concept descriptions as well. Moreover, if r is a role name in N_R, then the *existential restriction* $\exists r.C$ and the *value restriction* $\forall r.C$ are concept descriptions, too.

Now, an ontology $\mathcal{O}$, formalized in the syntax of the DL $\mathcal{ALC}/\mathcal{ALC}4$, consists of two components, called a *TBox* and an *ABox*. A TBox is a set containing a finite number of *general concept inclusions* (*GCIs*) of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept descriptions. An ABox is a finite set of *concept assertions* of the form $C(a)$ and *role assertions* of the form $r(a_1, a_2)$, where C is a concept description, r is a role name, and a_1 and a_2 are individual names taken from the set N_I. GCIs, concept assertions and role assertions together are also often called *ontology axioms* (or just *axioms*).

In order to define the semantics of $\mathcal{ALC}$-formulas, we consider a non-empty *universe* of individuals Δ^I and the *interpretation function* $\cdot^I$, which is defined in the second column of Table 2.1. The *interpretation tuple* $I = \langle \Delta^I, \cdot^I \rangle$ *satisfies* a GCI, a concept assertion or a role assertion if the respective conditions in the box at the bottom of Table 2.1 are fulfilled. An interpretation I is a *model* of a TBox or an ABox if and only if it satisfies all axioms of the respective component. Finally, an interpretation I is a model of an ontology $\mathcal{O}$ if and only if it models the TBox and the ABox of $\mathcal{O}$. In the DL literature, a concept description C is said to be *satisfiable* w.r.t an ontology $\mathcal{O}$ if and only if there is a model of $\mathcal{O}$ such that C^I is not equivalent to the empty set in the corresponding interpretation. Moreover, an axiom is said to be a *logical consequence* of an ontology $\mathcal{O}$ if and only if it is satisfied in every model of $\mathcal{O}$. A comprehensive treatment of the theory and practice of DLs has been published by Baader et al. [BCMNP03].

For interpreting $\mathcal{ALC}$-formulas under a four-valued semantics, a four-valued interpretation $\mathcal{I}$ can be represented by a tuple $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ as before, where $\Delta^{\mathcal{I}}$ is still a non-empty universe of individuals, but the interpretation function $\cdot^I$ has to be replaced by a *paraconsistent interpretation function* $\cdot^{\mathcal{I}}$. The interpretation function of $\mathcal{ALC}4$ is defined in the third column of Table 2.1 and the resulting interpretations are called *4-interpretations* by Maier

Syntax	$\mathcal{ALC}$ Semantics	$\mathcal{ALC}4$ Semantics [MMH13]
$a \in N_I$	$a^I \in \Delta^I$	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
$A \in N_C$	$A^I \subseteq \Delta^I$	$\langle P, N \rangle$, where $P, N \subseteq \Delta^{\mathcal{I}}$
$r \in N_R$	$r^I \subseteq \Delta^I \times \Delta^I$	$\langle P, N \rangle$, where $P, N \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\top$	Δ^I	$\langle \Delta^{\mathcal{I}}, \emptyset \rangle$
$\bot$	$\emptyset$	$\langle \emptyset, \Delta^{\mathcal{I}} \rangle$
$C_1 \sqcap C_2$	$C_1^I \cap C_2^I$	$\langle P_1 \cap P_2, N_1 \cup N_2 \rangle$, where $C_i^{\mathcal{I}} = \langle P_i, N_i \rangle$
$C_1 \sqcup C_2$	$C_1^I \cup C_2^I$	$\langle P_1 \cup P_2, N_1 \cap N_2 \rangle$, where $C_i^{\mathcal{I}} = \langle P_i, N_i \rangle$
$\neg C$	$\Delta^I \setminus C^I$	$\langle N, P \rangle$, where $C^{\mathcal{I}} = \langle P, N \rangle$
$\exists r.C$	$\{x \in \Delta^I \mid \exists y.[(x,y) \in r^I \wedge y \in C^I]\}$	$\langle \{x \mid \exists y.[(x,y) \in p^+(r^{\mathcal{I}}) \wedge y \in p^+(C^{\mathcal{I}})]\}, \{x \mid \forall y.[(x,y) \in p^+(r^{\mathcal{I}}) \rightarrow y \in p^-(C^{\mathcal{I}})]\} \rangle$
$\forall r.C$	$\{x \in \Delta^I \mid \forall y.[(x,y) \in r^I \rightarrow y \in C^I]\}$	$\langle \{x \mid \forall y.[(x,y) \in p^+(r^{\mathcal{I}}) \rightarrow y \in p^+(C^{\mathcal{I}})]\}, \{x \mid \exists y.[(x,y) \in p^+(r^{\mathcal{I}}) \wedge y \in p^-(C^{\mathcal{I}})]\} \rangle$
$C(a)$	$a^I \in C^I$	$a^{\mathcal{I}} \in p^+(C^{\mathcal{I}})$
$r(a_1, a_2)$	$(a_1^I, a_2^I) \in r^I$	$(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in p^+(r^{\mathcal{I}})$
$C_1 \sqsubseteq C_2$	$C_1^I \subseteq C_2^I$	$p^+(C_1^{\mathcal{I}}) \subseteq p^+(C_2^{\mathcal{I}})$

Table 2.1: Syntax and semantics of $\mathcal{ALC}$ and $\mathcal{ALC}4$.

et al. According to the authors [MMH13], a *positive* and a *negative extension* has to be assigned to every concept description in order to extend the classical two-valued semantics of $\mathcal{ALC}$ to a four-valued logic. As a result, the $\mathcal{ALC}4$-semantics interprets a concept description C by a tuple such that $C^{\mathcal{I}} = \langle P, N \rangle$, where $\mathcal{I}$ is a 4-interpretation, and P and N are subsets of the corresponding universe $\Delta^{\mathcal{I}}$. Maier et al. also define two functions that map every concept description to its positive and negative extension, respectively, in the following manner: $p^+(C^{\mathcal{I}}) = P$ and $p^-(C^{\mathcal{I}}) = N$. In this way, it is possible to map each concept assertion $C(a)$ to one of Belnap's four truth values:

- $C(a)$ is *paraconsistent* in $\mathcal{I}$ iff $a^{\mathcal{I}} \in p^+(C^{\mathcal{I}})$ and $a^{\mathcal{I}} \in p^-(C^{\mathcal{I}})$,
- $C(a)$ is *true* in $\mathcal{I}$ iff $a^{\mathcal{I}} \in p^+(C^{\mathcal{I}})$ and $a^{\mathcal{I}} \notin p^-(C^{\mathcal{I}})$,
- $C(a)$ is *false* in $\mathcal{I}$ iff $a^{\mathcal{I}} \notin p^+(C^{\mathcal{I}})$ and $a^{\mathcal{I}} \in p^-(C^{\mathcal{I}})$, and
- $C(a)$ is *undefined* in $\mathcal{I}$ iff $a^{\mathcal{I}} \notin p^+(C^{\mathcal{I}})$ and $a^{\mathcal{I}} \notin p^-(C^{\mathcal{I}})$.

Likewise, role assertions can be evaluated to four different truth values by getting assigned a positive extension P and a negative extension N, where P and N are subsets of the cross product $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ in this case. The functions p^+ and p^- can be defined for roles in the same way as they have been defined for concept descriptions. We make use of this mapping when proving the faithfulness of our semantics w.r.t. $\mathcal{ALC}4$ in Chapter 3. On the basis of the paraconsistent interpretation function defined in Table 2.1, we can now define the satisfaction and entailment condition of the four-valued ontology semantics, in line with the definitions provided by Maier et al. [MMH13].

Definition 2.2 (Four-valued DL semantics [MMH13])**.** A 4-interpretation $\mathcal{I}$ is a *4-model* of an $\mathcal{ALC}4$ ontology $\mathcal{O}$, written $\mathcal{I} \models_4 \mathcal{O}$, if and only if $\mathcal{I}$ satisfies all axioms in the TBox and the ABox of $\mathcal{O}$ according to Table 2.1. An ontology $\mathcal{O}$ *4-entails* an axiom A w.r.t. $\mathcal{ALC}4$[5], written $\mathcal{O} \models_{\mathcal{ALC}4} A$, if and only if every 4-model of $\mathcal{O}$ is a 4-model of A.

While Maier et al. define three different kinds of GCIs, we just present the semantics of the *internal inclusion* operator of Maier et al. in Table 2.1 since only this operator will be used in the approach we present in Chapter 3. The authors equate their *internal inclusion* operator $\sqsubset$ with the standard DL inclusion operator $\sqsubseteq$, so that a common syntax for $\mathcal{ALC}$ and $\mathcal{ALC}4$ ontologies can be maintained [MMH13].

2.2.2 Removal of Truth Value Gaps and Gluts

As mentioned above, Maier et al. also discuss constraints on the truth values that can be added to their framework, which results in what they call the removal of truth value *gaps* and *gluts*, respectively. In this way, the formalism essentially becomes either three-valued, or two-valued if both are removed, so that classical reasoning can be simulated in $\mathcal{ALC}4$ [MMH13]. Practically, this can be done by adding further axioms to an $\mathcal{ALC}4$-ontology $\mathcal{O}$, which enforce the LEM or the LNC, respectively. The technique is adopted from Arieli [Ari04]. For selectively enforcing the LEM to hold under the four-valued ontolgy semantics, the following set of axioms has to be added to an $\mathcal{ALC}4$-ontology [MMH13]:

$$LEM(\mathcal{O}) =_{def} \{\top \sqsubseteq (A \sqcup \neg A) \mid A \in N_C\}$$

This formulation is a simplified version of the one stated by Maier et al. since we only consider $\mathcal{ALC}4$-ontologies here. In the case of $\mathcal{SROIQ}4$-ontologies, also self restrictions and nominals have to be taken into account. In order to enforce the LNC, another set of axioms has to be added to $\mathcal{O}$, namely:

$$LNC(\mathcal{O}) =_{def} \{(A \sqcap \neg A) \sqsubseteq \bot \mid A \in N_C\}$$

Here, enforcing the LEM is especially interesting because in this way a paraconsistent logic can be maintained, while at the same time the semantics of the resulting formalism is closer to the classical semantics of DLs. This is illustrated by the following example.

Example 2.3. According to Zhang et al., the following equivalence holds under the classical semantics of DLs, given two concepts C_1 and C_2 and a two-valued interpretation I:

$$I \models C_1 \sqsubseteq C_2 \text{ if and only if } I \models \neg C_1 \sqcup C_2(a) \text{ for all } a \in N_I.$$

This equivalence is often used to reduce reasoning over a TBox and an ABox to reasoning over only an ABox [ZLW10]. However, under the four-valued ontology semantics

[5] Maier et al. define entailment w.r.t. the more expressive DL $\mathcal{SROIQ}4$. We adapt their notation to the case of $\mathcal{ALC}4$.

presented by Maier et al., this equivalence does not hold in general. For example, this is the case if for some 4-interpretation $\mathcal{I}$ and some individual a it holds that $a^{\mathcal{I}} \notin p^+(C_1^{\mathcal{I}})$, $a^{\mathcal{I}} \notin p^-(C_1^{\mathcal{I}})$, $a^{\mathcal{I}} \notin p^+(C_2^{\mathcal{I}})$ and $a^{\mathcal{I}} \notin p^-(C_2^{\mathcal{I}})$, and for all other individuals a' it holds that $a'^{\mathcal{I}} \notin p^+(C_1^{\mathcal{I}})$ and $a'^{\mathcal{I}} \in p^-(C_1^{\mathcal{I}})$, i.e. $C_1(a)$ and $C_2(a)$ are *undefined* in $\mathcal{I}$, and $C_1(a')$ is *false* in $\mathcal{I}$ for all individuals a'. Then $\mathcal{I} \models C_1 \sqsubseteq C_2$ clearly holds due to the four-valued semantics of GCIs. At the same time, it is not the case that $\mathcal{I} \models \neg C_1 \sqcup C_2(a)$ holds for all $a \in N_I$ since the negation of the concept assertion $C_1(a)$, which is evaluated to *undefined* in $\mathcal{I}$, is also *undefined*, and $C_2(a)$ is evaluated to *undefined* in $\mathcal{I}$ as defined.

Now, if the 4-interpretation $\mathcal{I}$ additionally had to model the axiom $\top \sqsubseteq (C_1 \sqcup \neg C_1)$, then $C_1(a)$ could not be *undefined* in $\mathcal{I}$ anymore. In this way, the mentioned equivalence can still be used in a paraconsistent logic which obeys the LEM. $\Diamond$

Besides the equivalence discussed in Example 2.3, Zhang et al. show that several other logical equivalences hold in a three-valued logic which cannot be obtained by using a four-valued logic.

2.3 Semantics for Logic Programs with Classical Negation

Logic programming is a widely used knowledge representation formalism, which is based on declarative rules and uses the *default negation* operator **not** in interaction with the CWA to express non-monotonic information [BG94]. Thus, a default-negated proposition **not** A is *true* if and only if A is not known to be *true*. This differs fundamentally from the meaning of the classical negation operator $\neg$. LPs can be divided into different classes according to the expressiveness of their respective syntax. The most simple form of a non-monotonic LP is a so-called *normal logic program*, which does not allow disjunctions or default negation in the heads of rules. In addition to default negation, it is often useful to introduce classical negation in the syntax of logic programs in order to be able to express that some piece of knowledge is false because its classical negation can be derived, and not just because its positive version is not derivable[6]. Furthermore, paraconsistent semantics for LPs are only of interest when classical negation is also considered because no inconsistencies[7] can emerge when only default negation is used in a program. If a normal logic program additionally contains classically (or strongly) negated atoms, it is called an *extended logic program*. For the purpose of this thesis, we will restrict ourselves to extended logic programs and do not consider program rules containing disjunctions.

[6] A famous example that shows the usefulness of adding classical negation to LPs is attributed to John McCarthy [GL91]. Assume the policy that "The school bus may cross the railway tracks if no train is approaching." is formalized by the following rule using default negation: $Cross \leftarrow \mathbf{not}\, Train$. Now, it might be the case that knowledge about an approaching train is not available for some reason, so that the previous rule is not safe. In this situation, one would want to express that the rails can be crossed by the bus if there is evidence for the fact that no train is approaching. This can only be expressed by means of classical negation in the following way: $Cross \leftarrow \neg Train$.

[7] The term 'inconsistency' is understood in the classical sense here. There are LPs that do not have any models under certain semantics and do not contain classical negation. Such programs are said to be *incoherent* [SI95].

Next, we give the definition of the syntax of an extended logic program, which follows the presentation by Damásio [Dam96]. Though it is customary to make use of variables in the syntax of logic programs, usually just the ground version of a logic program is considered, i.e. the version where every rule that contains variables is replaced by all its ground instances such that variables are replaced by terms from the *Herbrand universe* of the program [GL88]. This is possible w.l.o.g. since the CWA is employed. The ground atoms in the resulting program form the *Herbrand base* of the program [BG94].

Definition 2.4 (Extended logic program)**.** An *extended logic program* Π consists of a set of rules of the form

$$H \leftarrow A_1, \ldots, A_n, \mathbf{not}\, B_{n+1}, \ldots, \mathbf{not}\, B_m,$$

where H and each element in the sets $\{A_i\}$ and $\{B_j\}$ is either a ground atom A or a classically negated ground atom $\neg A$. An atom A is called the *explicit complement* of the classically negated atom $\neg A$, and vice versa. The atoms and their explicit complements occurring in a program are called the *objective literals* (or *program atoms*) of Π and form its *extended Herbrand base* denoted by $\mathcal{H}^e_\Pi$. The atom H is called the *head* of the respective rule and the atoms in $\{A_i\}$ and $\{\mathbf{not}\, B_j\}$ together form its *body*. The program Π is called an *extended positive logic program* if and only if $n = m$ holds for all rules in Π.

Due to the declarative nature of logic programming, a large number of approaches that assign a semantics to LPs with classical negation can be found in the literature (refer to the paper by Apt and Bol [AB94] for a survey). Most of them are based on one of two, which arguably represent the two most popular ways of assigning a meaning to LPs because they exhibit a range of desirable properties, namely

- the *Stable Model Semantics* (*SMS*) of Gelfond and Lifschitz [GL88], and
- the *Well-Founded Semantics* (*WFS*) of Van Gelder et al. [GRS91a].

In general, it is straightforward to assign a model to a positive extended logic program Π by computing the fixpoint of the *immediate consequence operator*, usually denoted by T_Π, that collects all objective literals in the heads of those rules of which the objective literals in the body could already be derived [GRS91a]. The resulting model is a *minimal Herbrand model* and called the *least Herbrand model* of Π [GL88]. In order to obtain models of logic programs containing default negation, Gelfond and Lifschitz devise a transformation that yields a positive version of a logic program given a subset of the program atoms [GL88]. It is often called *GL-transformation* in the literature due to its inventors. Our definition matches their original definition, but is adapted to our notation.

Definition 2.5 (GL-transformation [GL88])**.** Let Π be an extended logic program. For any set M of atoms from Π, let Π_M be the program obtained from Π by deleting

(i) each rule that has a negative literal $\mathbf{not}\, B_i$ in its body with $B_i \in M$, and

(ii) all negative literals in the bodies of the remaining rules.

Gelfond and Lifschitz define a stable model of a program Π to be a subset M of the atoms occurring in Π such that M is the *least model* of the positive program Π_M [GL88]. The computation of the least model of the GL-transformation of a program Π w.r.t. a set of program atoms M is often expressed by means of the so-called *Gamma-operator* Γ, and abbreviated by $\Gamma_\Pi(M)$ [Dam96]. A set of program atoms M is a stable model of a program Π if and only if $M = \Gamma_\Pi(M)$ [GL88]. According to Dantsin et al., a normal logic program can have zero, one, or multiple stable models and checking whether a model exists is NP-complete, while reasoning with the SMS is CONP-complete[8] [DEGV01]. The intractability of the SMS intuitively results from the fact that a subset of the Herbrand base first has to be fixed before it is possible to check if the respective subset constitutes a stable model. In spite of the high complexity of logic programming under the SMS, the formalism is semantically stronger than many other semantics for LPs in that it allows to derive more *true* and *false* consequences [GRS91a; KAH11].

On the other hand, the WFS for LPs assigns a unique model to every normal logic program, the so-called *well-founded model*, and reasoning under the WFS is P-complete [DEGV01]. The price for the tractable complexity of reasoning algorithms for the WFS is that it constitutes a weaker semantics. Hereby, the WFS can be viewed as a generalization of the SMS to three-valued models, where a program atom can be interpreted with the additional truth value *undefined*. At the same time, the WFS is sound w.r.t. the SMS, which means that if an atom of a program is *true* or *false* in the well-founded model, then it is also *true* or *false* in every stable model of the program, respectively [GRS91b; KAH11]. Besides the lower complexity, the WFS has several other advantages over the SMS. For instance, the semantics is *cumulative* and *relevant* [Dix95], which means that it can be used for top-down query-answering where only the relevant part of the program has to be considered and tabling techniques can be used in this process [AKS13; KAH11]. These methods, which further increase the efficiency of the approach, have been implemented for the WFS in the proof procedure *SLG* [CW96], and in the procedure *SLX* [ADP94] for LPs with classical negation. Consequently, the WFS is well-suited as an underlying program semantics for hybrid knowledge bases since the latter should be able to reason efficiently with large amounts of data [KAH11].

So far, a program atom A and its explicit complement $\neg A$ are regarded to be independent syntactic entities and hence, algorithms for computing the stable models or the well-founded model of an extended logic program work like for programs that do not contain classical negation. Usually, a *noncontradiction-condition* is imposed on models of programs containing classical negation, so that A and $\neg A$ may not be part of a model simultaneously [PA92]. Alternatively, the Principle of Explosion is sometimes introduced explicitly into the semantics, by defining that if a pair of complementary literals is part of a minimal model of an extended positive logic program computed by the immediate consequence operator, then all other program atoms are part of the minimal model as

[8] Note that the complexity results obviously hold for extended logic programs as well since we merely treat classical negation as a syntactic construct so far.

well [GL91]. Furthermore, classical negation influences the truth evaluation induced by a model of a program under the respective semantics. For instance, Gelfond and Lifschitz assign the truth value *true* to an atom A if A is contained in a respective stable model[9], the truth value *false* if $\neg A$ is in a stable model, and the truth value *unknown* if neither A nor $\neg A$ is part of some stable model [GL91]. Damásio and Pereira coin the term "weak negation view" for the stance taken by approaches that treat atoms and their explicit complements basically as being independent (at least w.r.t. their procedural semantics) [DP98]. In contrast to this perspective, Pereira and Alferes postulate that classical and default negation should be related by means of the so-called *Coherence Principle*, which states that classical negation should imply default negation [PA92] and will be discussed in detail later. According to Damásio and Pereira, the semantics $WFSX_p$ developed by the authors, and its extensions, are the only approaches that take the Coherence Principle into account [DP98]. Naturally, contradictions cannot always be avoided regarding the model-theoretic characterizations of semantics that give meaning to extended logic programs, e.g. if A and $\neg A$ are both part of some stable model. Consequently, paraconsistent semantics for extended logic programs are required to give a non-trivial meaning to inconsistent logic programs.

2.3.1 Paraconsistent Semantics for Logic Programs

A number of paraconsistent semantics for extended positive logic programs have been developed, as well as some for extended logic programs extending either the SMS or the WFS. Many of these approaches are overviewed in a survey by Damásio and Pereira [DP98]. Likely the most encompassing approach to paraconsistent LPs can be found in a paper by Alcântara et al. [ADP02]. The authors have created a paraconsistent framework for LPs where, based on Fitting's work discussed in section 2.1.1, an arbitrary bilattice of truth values can be used as underlying algebraic structure [ADP02]. Furthermore, the authors implement the Coherence Principle in their formalism and, as a result, are able to show that a class of stable models, which they call *coherent answer sets*, is characterized by their approach as well.

A paraconsistent semantics for logic programs that extends the SMS has been developed by Sakama and Inoue [SI95]. Their semantics is defined for *extended disjunctive logic programs* and based on Belnap's four truth values. In order to compute all *paraconsistent stable models* of a program, the authors introduce a new fixpoint semantics, which works with a translation of a program into a semantically equivalent positive version of the program, called its *epistemic transformation*. Furthermore, the authors extend their approach to six truth values (cf. Figure 2.2), adding the truth values *suspiciously true* and *suspiciously false*, and extend their fixpoint semantics to a version that works with adorned objective literals in order to propagate the information that an atom has been derived from contradictory facts. In this way, Contradiction Support Detection is enabled to some degree.

[9] Gelfond and Lifschitz denote stable models by the term *answer sets* after introducing classical negation into their programs [GL91]. However, in the literature both terms are used interchangeably.

Moreover, the authors define *semi-stable models* on the basis of a nine-valued logic, illustrated in Figure 2.2, which can be assigned to programs in which an objective literal is implied by its default negation. Such programs do not have a stable model under the original SMS and are said to be *incoherent*, which can be viewed as a form of inconsistency according to Sakama and Inoue.

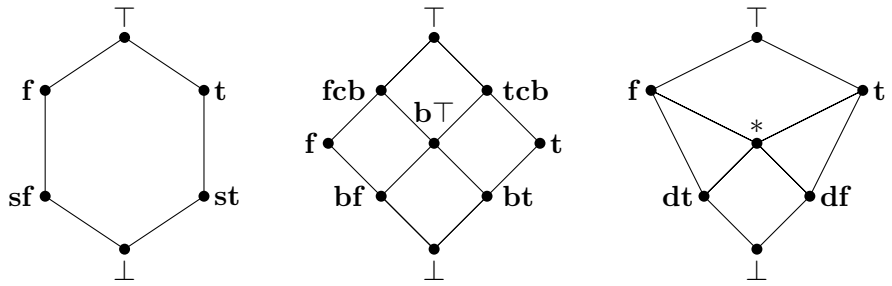

Figure 2.2: The lattice **VI** underlying the six-valued suspicious SMS of Sakama and Inoue [SI95]; the nine-valued bilattice **IX** on which the Semi-Stable Models of Sakama and Inoue are based [SI95]; and the bilattice **VII** of Sakama's seven-valued paraconsistent WFS [Sak92].

In case of the WFS, a paraconsistent extension has also been published by Sakama [Sak92]. It is based on a seven-valued bilattice, firstly introduced by Ginsberg [Gin88] and depicted in Figure 2.2, which makes it possible to represent default assumptions. The procedural definition of the *extended well-founded model* of an extended logic program Π mirrors the construction of the standard (i.e. non-extended) well-founded model, and Damásio and Pereira show that it can be obtained by computing the standard well-founded model of a simple transformation of Π [DP98]. Sakama considers *suspicious reasoning* in the context of paraconsistent WFS as well. For this, he extends the direct consequence operator used in the bottom-up computation of extended well-founded models such that every objective literal L is annotated by a set collecting those sets of objective literals from which L can be derived in the computation. In this way, derivations that involve contradictory facts can be distinguished. Damásio and Pereira state that this technique has the drawback that the annotation sets can become exponentially large w.r.t. the size of the program [DP98].

The Coherence Principle is not taken into account by the two previously described approaches and an incomplete form of suspicious reasoning is implemented as will be demonstrated in section 2.3.3. The paraconsistent extension of the WFS presented in the next section offers a solution for both of these shortcomings.

2.3.2 $WFSX_p$ and the Coherence Principle

We now discuss the *Paraconsistent Well-Founded Semantics with Explicit Negation* ($WFSX_p$) of Alferes et al. [ADP95; Alf93; Dam96] in more detail since the approach we develop is based on it. For this reason, here we introduce several notions that we will draw on in

Chapter 4 and Chapter 5. In doing so, we follow the presentation of the paper by Alferes et al. [ADP95]. The *well-founded model* of a program Π, denoted by $WFM(\Pi)$, can be represented by means of the union of two subsets of the extended Herbrand base $\mathcal{H}^e_\Pi$, which we denote by $T \cup \mathbf{not}\, F$[10]. Under the standard WFS the sets T and F are disjoint and a program atom A is defined to be *true* w.r.t. $WFM(\Pi)$ if and only if A is in $WFM(\Pi)$, *false* if and only if $\mathbf{not}\, A$ is in $WFM(\Pi)$, and *undefined* if and only if neither A nor $\mathbf{not}\, A$ is in $WFM(\Pi)$. The disjointness condition is dropped in $WFSX_p$ to enable paraconsistent interpretations of program atoms and the *paraconsistent well-founded model* of an extended logic program Π is denoted by $WFM_p(\Pi)$. An important property of $WFSX_p$ is that it satisfies the Coherence Principle, which states that classical negation implies default negation and can be formalized in the following way.

Definition 2.6 (Coherence Principle)**.** The paraconsistent well-founded model $WFM_p(\Pi)$ of an extended logic program Π satisfies the *Coherence Principle* iff $A \in WFM_p(\Pi)$ implies that $\mathbf{not}\, \neg A \in WFM_p(\Pi)$ and $\neg A \in WFM_p(\Pi)$ implies that $\mathbf{not}\, A \in WFM_p(\Pi)$, for every atom A and its explicit complement $\neg A$ in $\mathcal{H}^e_\Pi$.

The number of possible truth values that can be assigned to program atoms in $\mathcal{H}^e_\Pi$ by a paraconsistent well-founded model is increased to nine in the case of $WFSX_p$ [DP95]. The truth values can be ordered in a bilattice that has the same structure as the nine-valued bilattice of Sakama and Inoue shown in Figure 2.2. Nine truth values are implied by the fact that given an atom A and its explicit complement $\neg A$ appearing in a program Π, there are nine different ways in which they can occur in $WFM_p(\Pi)$, i.e. all subsets of $\{A, \neg A, \mathbf{not}\, A, \mathbf{not}\, \neg A\}$ that satisfy the Coherence Principle can be in $WFM_p(\Pi)$. Using a logic defined over the resulting bilattice $\mathcal{NINE}$, Damásio and Pereira give a model-theoretic characterization of $WFSX_p$ [DP95] by showing that every paraconsistent well-founded model is also a model in their nine-valued logic. However, to show the other direction, i.e. that nine-valued models correspond to paraconsistent well-founded models, the authors have to restrict the latter to so-called *supported models* and have to assume that the respective program does not contain infinite positive recursions [DP95]. Besides this model-based characterization, there are a number of other definitions of $WFSX_p$, e.g. by a *non-deterministic division* of programs, by a *compact bottom-up operator* or by an *alternating fixpoint construction* (cf. Damásio's PhD thesis [Dam96]). We present the latter definition here, which goes back on a construction introduced in [Gel89].

In order to implement the *Coherence Principle* in the alternating fixpoint construction of paraconsistent well-founded models, a program transformation has to be introduced, which has the effect that those objective literals of which the explicit complement is included in M cannot be derived by the operation $\Gamma_\Pi(M)$ presented in section 2.3. The following definition of this transformation is similar to Definition 3.2 in [ADP95].

[10]The prefixed set $\mathbf{not}\, F$ is the set of all elements in F prefixed by $\mathbf{not}$.

Definition 2.7 (Semi-normal version of a program [ADP95])**.** The *semi-normal version* of a program Π is the program Π_s obtained by adding to the (possibly empty) body of each rule with head H the default-negated explicit complement[11] of H.

The operator used in the alternating fixpoint construction of the paraconsistent well-founded model of a program Π represents a doubled version of the Γ-operator, so that the operator computes the result of $\Gamma_\Pi \Gamma_{\Pi_s}$ given a subset of $\mathcal{H}^e_\Pi$. Thus, the first application of the Γ-operator is done w.r.t. the semi-normal version of the respective program to ensure the Coherence Principle, while the operator applied to the resulting set of objective literals uses the original version of the program. When it is clear w.r.t. which program the operator $\Gamma_\Pi \Gamma_{\Pi_s}$ is applied (or it is not important), we will just write $\Gamma\Gamma_s$. While the Γ-operator itself is not monotonic, its doubled version $\Gamma\Gamma_s$ is monotonic for arbitrary sets of objective literals and Alferes et al. show that a fixpoint always exists [ADP95]. The authors provide the following constructive definition for the paraconsistent well-founded model of a program, which we have adapted to our notation.

Definition 2.8 (Paraconsistent well-founed model [ADP95])**.** Let Π be an extended logic program whose least fixpoint of the operator $\Gamma\Gamma_s$ is T. Then, the *paraconsistent well-founded model* of Π is $WMF_p(\Pi) = T \cup \mathbf{not}\,(\mathcal{H}^e_{\Pi_s} \setminus \Gamma_s T)$.

Intuitively, the set T contains those program atoms which are known to be "true", while the program atoms in $\mathcal{H}^e_{\Pi_s} \setminus \Gamma_s T$ are not known to be "true or undefined". In the original definition of the well-founded model it is ensured that an objective literal cannot be known to be "true" and not known to be "true or undefined" at the same time by imposing the condition that T must be a subset of $\Gamma_s T$. Note that the notions "true" and "undefined" do not refer to the truth values *true* and *undefined* of a truth valuation since an objective literal might actually be "true" and not "true or undefined" simultaneously in a paraconsistent well-founded model, which then corresponds to an evaluation to the truth value *inconsistent* in the logic $\mathcal{NINE}$. The least fixpoint of the operator $\Gamma\Gamma_s$ can be derived by means of the following *transfinite sequence* $\{I_\alpha\}$, which is directly adopted from Damásio [Dam96]:

$$\begin{aligned} I_0 &= \emptyset \\ I_{\alpha+1} &= \Gamma\Gamma_s \\ I_\delta &= \bigcup\{I_\alpha \mid \alpha < \delta\} \text{ for the limit ordinal } \delta \end{aligned}$$

According to Damásio, there is a smallest number λ such that I_λ is the least fixpoint of $\Gamma\Gamma_s$ [Dam96]. This result also shows that a paraconsistent well-founded model always exists and that it is unique. The author further notes that all objective literals that are obtained after any application of the operator $\Gamma\Gamma_s$ in the sequence are "true" in the resulting

[11] Thus, if H is of the form A, $\mathbf{not}\,\neg A$ is added to the body of the respective rule, and if H is of the form $\neg A$, $\mathbf{not}\,A$ is added.

paraconsistent well-founded model, and the objective literals that are not in the set obtained after an application of the operator Γ_s are those which are "false" in the model. Thus, two different sequences are implicitly present in the sequence $\{I_\alpha\}$, one increasing sequence maximizing the set of "true" objective literals, and one decreasing sequence minimizing the set of "true or undefined" objective literals, according to Damásio. The alternation between an increasing and a decreasing sequence gives the alternating fixpoint construction its name. The following example illustrates the computation of the least fixpoint of the operator $\Gamma\Gamma_s$ and the implementation of the Coherence Principle by using the semi-stable version of a program in the operator Γ_s.

Example 2.9 (Coherent alternating fixpoint construction)**.** Consider the following ground program Π.

$$\begin{array}{rcl} \mathsf{risk(food)} & \leftarrow & \mathbf{not}\,\mathsf{isLabelled(food)}. \\ \mathsf{isMon(food)} & \leftarrow & \mathsf{risk(food)}. \\ \neg\mathsf{isMon(food)} & \leftarrow & \mathsf{certForw(food)}. \\ \mathsf{certForw(food)} & \leftarrow & \end{array}$$

Here, isMon and certForw are abbreviations for isMonitored and hasCertifiedForwarder, respectively. The program expresses that a certain food is a risk if it cannot be proven that it isLabelled. Moreover, isMon(food) is *true* if food is a risk, and isMon(food) is *false* if certForw(food) holds. As certForw(food) is contained as a fact in the program and food can be proven to be a risk, it can also be proven that isMon(food) is *true* and *false*, simultaneously. The semi-normal version of Π, denoted by Π_s, is represented by the following ground program, according to Definition 2.7:

$$\begin{array}{rcl} \mathsf{risk(food)} & \leftarrow & \mathbf{not}\,\mathsf{isLabelled(food)}, \mathbf{not}\,\neg\mathsf{risk(food)}. \\ \mathsf{isMon(food)} & \leftarrow & \mathsf{risk(food)}, \mathbf{not}\,\neg\mathsf{isMon(food)}. \\ \neg\mathsf{isMon(food)} & \leftarrow & \mathsf{certForw(food)}, \mathbf{not}\,\mathsf{isMon(food)}. \\ \mathsf{certForw(food)} & \leftarrow & \mathbf{not}\,\neg\mathsf{certForw(food)}. \end{array}$$

According to Definition 2.8, the paraconsistent well-founded model of Π can be obtained by computing the least fixpoint of the operator $\Gamma\Gamma_s$. For this, we start with the empty set and apply the operator $\Gamma\Gamma_s$ repeatedly until a fixpoint is reached. The sequence presented above for the program Π is:

$$\begin{array}{rcl} I_0 & = & \emptyset \\ \Gamma_s(I_0) & = & \{\mathsf{risk(food)}, \mathsf{certForw(food)}, \mathsf{isMon(food)}, \neg\mathsf{isMon(food)}\} \\ I_1 = \Gamma\Gamma_s(I_0) & = & \{\mathsf{risk(food)}, \mathsf{certForw(food)}, \mathsf{isMon(food)}, \neg\mathsf{isMon(food)}\} \\ \Gamma_s(I_1) & = & \{\mathsf{risk(food)}, \mathsf{certForw(food)}\} \\ I_2 = \Gamma\Gamma_s(I_1) & = & \{\mathsf{risk(food)}, \mathsf{certForw(food)}, \mathsf{isMon(food)}, \neg\mathsf{isMon(food)}\} = I_1 \end{array}$$

All rules of the positive version of Π_s are contained in the GL-transformation Π_{sI_0} since I_0 is empty. The same is true for $\Pi_{\Gamma_s(I_0)}$ because the original version of the program is used and isLabelled(food) is not in $\Gamma_s(I_0)$. Hence, $\Gamma_s(I_0)$ and $\Gamma\Gamma_s(I_0)$ are equal. However, the second and the third rule are deleted from Π_s in the GL-transformation Π_{sI_1} as both isMon(food) and ¬isMon(food) could be derived in the previous iteration to be in the set T. Consequently, the Coherence Principle intervenes and removes both from the set of "true or undefined" objective literals. Since I_2 equals I_1, a fixpoint is reached after the second application of the operator $\Gamma\Gamma_s$. By Definition 2.8, the paraconsistent well-founded model of Π is

$$\begin{aligned} WFM_p(\Pi) = \{&\mathsf{risk(food)}, \mathsf{certForw(food)}, \mathsf{isMon(food)}, \neg\mathsf{isMon(food)}, \\ &\mathbf{not}\,\neg\mathsf{risk(food)}, \mathbf{not}\,\neg\mathsf{certForw(food)}, \mathbf{not}\,\mathsf{isMon(food)}, \mathbf{not}\,\neg\mathsf{isMon(food)}, \\ &\mathbf{not}\,\mathsf{isLabelled(food)}\}). \end{aligned}$$

Note that the program atoms isMon(food) and ¬isMon(food) are both in the set T and the set $\mathcal{H}^e_{\Pi_s} \setminus \Gamma_s T$ simultaneously, which makes them *inconsistent* under the nine-valued semantics employed by Damásio and Pereira. On the other side, the program atoms risk(food) and certForw(food) are *true*, while the program atom isLabelled(food) is interpreted to be *default false*. Further, note that the set $WFM_p(\Pi)$ satisfies Definition 2.6. ◊

Enforcing the Coherence Principle has the convenient side-effect that Contradiction Support Detection is already built into the alternating fixpoint construction. This property is the topic of the next section.

2.3.3 Contradiction Support Detection

As has already been hinted in the introduction, it is useful and might even be crucial for some applications that a paraconsistent semantics not only provides information about which facts are *inconsistent*, but also about those which can only be derived by involving *inconsistent* knowledge. In this way, one knows when to be "suspicious" about a piece of information because an error might have occurred. Hence, approaches applying so-called *suspicious reasoning* are closer to the idea of belief revision as inconsistencies are rather regarded as errors, and detecting "suspicious" information can be viewed as a form of debugging. Suspicious reasoning can be achieved by implementing some way of propagating inconsistencies into a logic, which results in what Damásio and Pereira call *Contradiction Support Detection* [DP97]. The authors give a general formal definition of dependence on contradiction in Definition 8, which is independent of the specific semantics used. We adapt it to our notation here.

Definition 2.10 (Dependence on contradiction [DP97])**.** Let Π be an extended logic program and $C = \{c, \neg c \mid c, \neg c \in \Pi\}$, the set of the contradictory facts in Π. We say that any objective literal L *depends on a contradiction* w.r.t. a semantics SEM if and only if there is

a set $S \subseteq C$ such that

$$SEM(\Pi) \cap \{L\} \neq SEM(\Pi - S) \cap \{L\}.$$

This means, an objective literal depends on a contradiction if and only if its truth value changes after removing only contradictory facts. Note that the definition is limited to contradictory facts, so that contradictory atoms which are derivable, but not contained as facts in the program, are not considered. Damásio and Pereira additionally state that the notion of dependence on contradiction can be made stronger by referring to the paraconsistent well-founded model of a program. In this case, an objective literal L is defined to depend on contradiction iff L and **not** L are both contained in the model [DP97].

Damásio and Pereira mention that in most work conducted on paraconsistent LP semantics the need for being able to detect support on contradictions has been "overlooked or not properly captured" [DP97]. Two approaches to paraconsistent LP semantics that integrate suspicious reasoning into their framework are those based respectively on the SMS [SI95] and the WFS [Sak92] which have been discussed in section 2.3.1. However, Damásio and Pereira remark that both approaches suppose that the truth of a default-negated program atom can never depend on a contradiction, which is not the case according to the authors and hence, makes their approach to Contradiction Support Detection incomplete [DP97]. They illustrate this issue by means of the following example.

Example 2.11 (Support on contradiction through default negation [DP97])**.** Consider the following extended logic program.

$$\begin{aligned} \mathsf{a} &\leftarrow \mathbf{not}\,\mathsf{b}. \\ \mathsf{b} &\leftarrow \mathbf{not}\,\mathsf{c}, \mathbf{not}\,\neg\mathsf{c}. \\ \mathsf{c} &\leftarrow \\ \neg\mathsf{c} &\leftarrow \end{aligned}$$

In this program, the program atom a clearly depends on a contradiction because it is not derivable anymore after removing one of the contradictory facts c and ¬c from the program. It is desirable that a paraconsistent semantics detects this dependence on contradiction. This is not the case for the paraconsistent SMS of Sakama and Inoue [SI95] and the paraconsistent WFS of Sakama [Sak92] because they do not propagate inconsistencies over default negation. ◊

In $WFSX_p$, Contradiction Support Detection is already built-in due to the enforcement of the Coherence Principle. For example, suppose that we add the program-rule

$$\mathsf{resolvedRisk(food)} \leftarrow \mathsf{isMon(food)}.$$

to the contradictory program in Example 2.9. Then, the program atom resolvedRisk(food)

is not part of the set $\Gamma_s(I_1)$ since isMon(food) is not derivable, i.e. it is also not in the set of "true or undefined" objective literals. However, simultaneously it is derivable by the operator $\Gamma\Gamma_s$ to be in the set of "true" objective literals since isMon(food) is contained in it, too. Consequently, the paraconsistent model of the program is

$$\begin{aligned} WFM_p(\Pi) = \{&\mathsf{risk(food)}, \mathsf{certForw(food)}, \mathsf{isMon(food)}, \neg\mathsf{isMon(food)}, \\ &\mathsf{resolvedRisk(food)}, \mathbf{not}\,\neg\mathsf{risk(food)}, \mathbf{not}\,\neg\mathsf{certForw(food)}, \mathbf{not}\,\mathsf{isMon(food)}, \\ &\mathbf{not}\,\mathsf{resolvedRisk(food)}, \mathbf{not}\,\neg\mathsf{resolvedRisk(food)}, \mathbf{not}\,\neg\mathsf{isMon(food)}, \\ &\mathbf{not}\,\mathsf{isLabelled(food)}\}) \end{aligned}$$

in this case and resolvedRisk(food) is evaluated to *true with contradictory belief* in the corresponding nine-valued model [DP95].

Though $WFSX_p$ is capable of handling situations as described in Example 2.11, it fails to detect dependence on contradiction when the program atom that should be interpreted to be *true with contradictory belief* occurs also in a rule with *undefined* body [DP97]. Damásio and Pereira provide the following example.

Example 2.12 (Failure to detect support on contradiction [DP97])**.** Consider the following extended logic program Π.

$$\begin{aligned} \mathsf{a} &\leftarrow \mathbf{not}\,\mathsf{b}. \\ \mathsf{a} &\leftarrow \mathsf{c}. \\ \mathsf{b} &\leftarrow \mathbf{not}\,\mathsf{b}. \\ \mathsf{c} &\leftarrow \\ \neg\mathsf{c} &\leftarrow \end{aligned}$$

In this program, the truth of a depends on the contradictory program atom c. However, $\mathbf{not}\,\mathsf{a}$ is not contained in $WFM_p(\Pi)$ because b is *undefined* due to the third rule and hence, the fact a is contained in the GL-transformation used by the operator Γ_s because of the first rule. ◊

In contrast to $WFSX_p$, this example is handled correctly by the approach of Sakama and Inoue [SI95] and the approach by Sakama [Sak92]. Damásio and Pereira show that a stronger form of Contradiction Support Detection can be obtained by using a transformation of the program in each step of the alternating fixpoint construction. For models computed by the resulting algorithm, Contradiction Support Detection is complete for arbitrary extended logic programs. In addition, Damásio and Pereira also demonstrate how the propagation of contradictions can be blocked in certain cases where the propagation is not desirable [DP97]. In our six-valued semantics, we implement the weaker form of Contradiction Support Detection and discuss an extension in the section about future work.

2.4 Hybrid Knowledge Bases

Besides hybrid approaches that couple rule and ontology formalisms within a single system, according to Knorr et al. [KAH11], a number of approaches exist that either enrich a DL with constructs for rules (e.g. the *Semantic Web Rules Language* (*SWRL*) [HP04]), or that investigate fragments of first-order logic which are suitable for expressing rules (e.g. *Description Logic Programs* (*DLP*) [GHVD03] and *Horn-*$\mathcal{SHIQ}$ [HMS05]). The latter two approaches have in common that they are based on first-order logic and thus, non-monotonic reasoning is not enabled by them. However, they have the advantage that standard DL-reasoners often can be applied straightforwardly to these frameworks [KAH11]. An overview over several approaches that combine rules and ontologies can be found in the papers by Hitzler and Parsia [HP09] and by Eiter et al. [EIKP08].

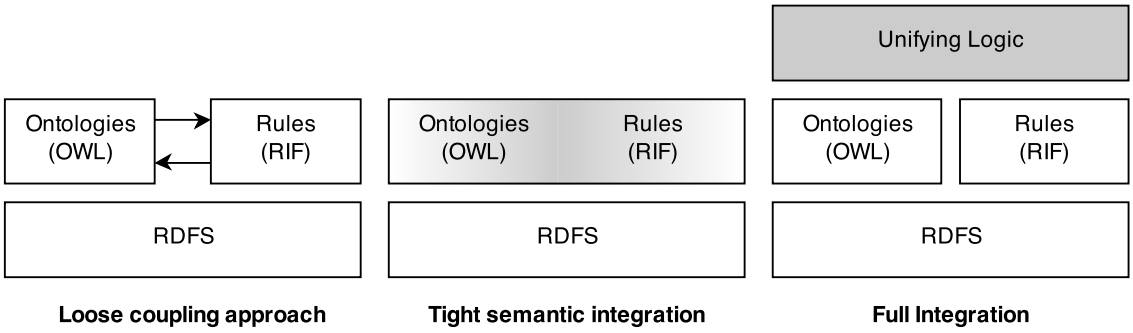

Figure 2.3: Illustration of the loose coupling, thight semantic integration and full integration approach to hybrid knowledge bases (adopted from Eiter et al. [EIKP08]).

As we have discussed in the introduction, a real integration of reasoning with DLs under the OWA and with LPs under the CWA is of high interest for many applications since it results in a formalism that is more expressive than each of the two formalisms alone. However, as Eiter et al. note, the combination of these two decidable fragments of first-order logic is highly non-trivial since a naive combination of a simple DL with a positive normal logic program is already undecidable [EIKP08]. Hence, a new semantic framework has to be developed by hybrid approaches that encompasses the expressiveness of DLs and LPs equally effective. Recently, a multitude of approaches has been developed in the area of hybrid knowledge bases that tackle the challenge of coupling DLs and LPs within a single framework, such as *description logic programs* (or *dl-programs*) [EILST08], $\mathcal{DL}$*+log* [Ros06], *Open Answer Set Programs* [HNV05] and *Hybrid MKNF knowledge bases* [MR10].

According to Eiter et al., approaches to hybrid knowledge bases can be grouped into three categories: *loose coupling*, *tight semantic integration*, and *full integration approaches* [EIKP08]. In the loose coupling approach, the DL- and the LP-component are interpreted under separate semantics dedicated to the respective formalism and an *interface mechanism* is implemented that allows exchange of information between the components in either one or both directions. In case of the tight semantic integration approach, the

knowledge base is interpreted by an *integrated model* that is composed of two parts sharing the same domain together with a definition of agreement between them, according to Eiter et al. Finally, the components of hybrid knowledge bases of the full integration approach share a common vocabulary and are interpreted within a single semantic framework [EIKP08]. The three degrees of tightness of hybrid knowledge base formalisms are illustrated in Figure 2.3, which is presented in a nearly identical form in [EIKP08]. The figure depicts the *Resource Description Framework Schema* (*RDFS*) as the basis of hybrid knowledge bases, providing a unified grammar for describing diverse Semantic Web formalisms. It shows further that only the full integration approach provides a *unifying logic* for rules and ontologies, which is also part of the conceived architecture of the Semantic Web, as illustrated in Figure 2.4.

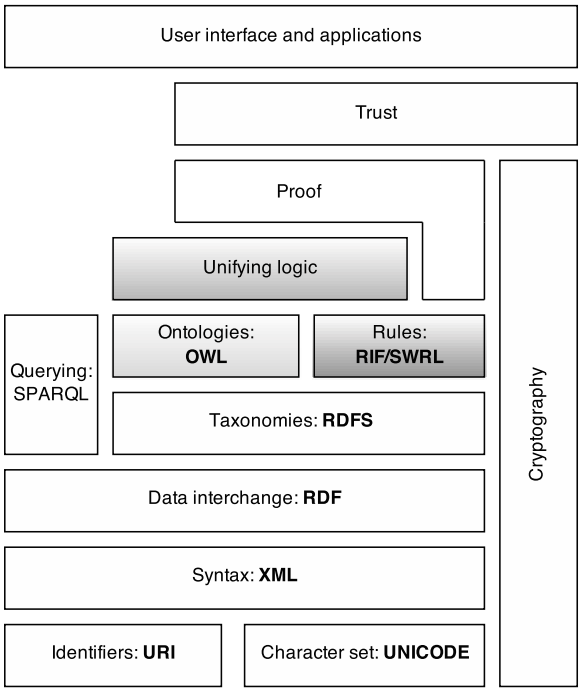

Figure 2.4: The Semantic Web Stack (like shown in [Wik14]).

Dl-programs, developed and investigated by Eiter et al., represent a loose coupling approach that allows for non-monotonic reasoning under either the SMS [EILST08] or the WFS [EILS11], where a DL component can be queried from within the program component. Program-atoms that pose a query to the ontology are termed *dl-atoms* by Eiter et al. Here, the DL component is viewed as an external oracle, so that it is interpeted by the usual DL-semantics and available reasoners can be used to answer queries from the LP- to the DL-component. On the other hand, in the version based on the SMS, an ASP solver can be used to compute the semantics of the program. In addition, dl-atoms can modify the ABox of the respective ontology by adding positive or negative assertions resulting from the program, so that the interaction between the components works in both directions [EIKP08]. The authors show that every positive dl-program has a least

Herbrand model, where the *Herbrand base* in this case consists of all program atoms instantiated with constants from the program component and all known individuals from the DL-component [EILST08]. Moreover, the *dlvhex* system [EIST06] provides an implementation of the approach, which, according to Eiter et al., has been used already for a series of applications, such as *ontology merging*, *bio-ontologies*, *web querying* [EIKP08].

The hybrid knowledge base formalism $\mathcal{DL}$+log of Rosati [Ros06], which integrates DLs with disjunctive Datalog, can be classified as a tight semantic integration approach [EIKP08]. Like dl-programs, the semantics of $\mathcal{DL}$+log knowledge bases is grounded in the SMS. *Rule predicates* and *classical predicates* are still distinguished in $\mathcal{DL}$+log, but the two components interact in terms of their semantics. For computing the semantics of a $\mathcal{DL}$+log knowledge base, the program component is reduced w.r.t. a given interpretation of the first-order predicates occurring in them in a first step, and afterwards the SMS is used for obtaining the stable models of the resulting program not containing classical predicates anymore [EIKP08]. In order to ensure decidability of the approach, the authors define a *weak safety* condition for program rules, which is less restrictive than the *DL-safety* condition usually imposed on hybrid MKNF knowledge bases (and thus, less restrictive than the condition we will present for our approach).

Hybrid MKNF knowledge bases, which we will study in depth in the remainder of this thesis, are a representative of the last category constituting a full integration approach. They have been introduced first by Motik and Rosati [MR06] and have already been extended in several ways in subsequent works (e.g. in [HHL14; HLH11; IKL13; KAH11; MR10; SLS11]).

2.4.1 Hybrid MKNF Knowledge Bases

Hybrid MKNF knowledge bases are formalized in the logic of *Minimal Knowledge and Negation as Failure* (*MKNF*), which represent a formalism for non-monotonic reasoning and have been introduced by Lifschitz in 1991 [Lif91]. As Lifschitz shows, the logic of MKNF is closely related to other non-monotonic formalisms such as Reiter's *Default Logic* [Rei80] and *Circumscription* [McC86]. It is an extension of *first-order logic with equality* that additionally provides two modal operators (also called epistemic operators by Lifschitz), a modal **K**- and a modal **not**-operator. These modal operators are able to introspect the knowledge base and according to Eiter et al. [EIKP08], a formula $\mathbf{K}\,\phi$ has the intuitive meaning that ϕ is necessarily known to hold, while **not** ϕ means that ϕ is not known to hold. Both operators are interpreted in a Kripke structure where every world is accessible from any other world (corresponding to the modal logic S5) [MR10]. We begin now by defining the syntax of general MKNF formulas here and present a restricted version for hybrid MKNF knowledge bases afterwards, before introducing a two- and a three-valued MKNF semantics in the next section. In what follows, we mainly rely on the definitions provided by Motik and Rosati [MR07; MR10], and by Knorr et al. [KAH08; KAH11].

The syntax of MKNF formulas is defined over a first-order signature $\Sigma = (\Sigma_c, \Sigma_f, \Sigma_p)$,

where the set Σ_c contains all constant symbols, the set Σ_f contains the function symbols of the logic and the set Σ_p contains the predicate symbols including $\approx$, the binary equality predicate. The symbols $\top$, $\bot$[12] and a *first-order atom* $P(t_1, \ldots, t_n)$, given a predicate symbol P and *first-order terms* t_i, are atomic *MKNF formulas*. Let φ, φ_1 and φ_2 be MKNF formulas. Then $\neg\varphi$, $\exists x : \varphi$, $\varphi_1 \wedge \varphi_2$, $\mathbf{K}\,\varphi$ and $\mathbf{not}\,\varphi$ are MKNF formulas, too. As in first-order logic, the formulas $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\neg\varphi_1 \vee \varphi_2$ and $\neg(\exists x : \neg\varphi)$ are abbreviated by $\varphi_1 \vee \varphi_2$, $\varphi_1 \supset \varphi_2$, and $\forall x : \varphi$, respectively [KAH11]. In the following, $\varphi[t_1/x_1, \ldots, t_n/x_n]$ stands for the formula that results from substituting the free variables x_i in φ by the terms t_i [KAH11]. A formula of the form $\mathbf{K}\varphi$ is called a *modal* **K**-*atom* and $\mathbf{not}\varphi$ is called a *modal* **not**-*atom*. Modal **K**-atoms and **not**-atoms are *modal atoms*, according to Motik and Rosati [MR10].

Hybrid MKNF knowledge bases are formalized by MKNF formulas that are constrained to a specific form. According to Knorr et al., they consist of a *program component* $\mathcal{P}$ represented by a finite set of rules containing modal **K**- and **not**-atoms, and an *ontology component* $\mathcal{O}$ formalized in some DL [KAH11]. Therefore, hybrid MKNF knowledge bases are implicitly parameterized by the respective DL language that is used to formalize the ontology component [MR10]. Motik and Rosati state that every DL that fulfills the following requirements can be utilized to express the ontology component of a hybrid MKNF knowledge base [MR07]:

1. Every ontology $\mathcal{O}$ expressed in the DL can be translated into a formula $\pi(\mathcal{O})$ of function-free first-order logic with equality,

2. the DL supports ABox-assertions of the form $P(a_1, \ldots, a_n)$ where P is a predicate and all a_i are constants of the DL, and

3. satisfiability checking and instance checking are decidable in the DL.

The subsequent definition of *MKNF rules* and hybrid MKNF knowledge bases is adapted from [KAH08]. Since we do not consider LPs where disjunctions are allowed in the head of a rule in our approach, we only provide the definition for non-disjunctive MKNF rules here. However, as Knorr et al. note, this can be partly compensated by the fact that the disjunction operator can be used in the ontology component (in case this is allowed by the respective DL used to formalize the ontology) [KAH11].

Definition 2.13 (Hybrid MKNF knowledge base [KAH08])**.** Let $\mathcal{O}$ be a DL-ontology. A function-free first-order atom $P(t_1, \ldots, t_n)$ over Σ such that P is $\approx$ or occurs in $\mathcal{O}$ is called a *DL-atom*; all other atoms are called *non-DL-atoms*. An *MKNF rule* r has the following form where H, A_i, and B_i are function-free first-order atoms:

[12] Usually, the formulas $\top$ and $\bot$ are equated with the formulas $a \vee \neg a$ and $a \wedge \neg a$, respectively. This is the case for the two- and three-valued MKNF semantics introduced in section 2.4.2, so that they are not explicitly treated in the model-theoretic definition of the respective semantics. For our purpose, we introduce $\top$ and $\bot$ as atomic MKNF formulas and explicitly assign a meaning to them in our six-valued semantics. This is necessary since e.g. the formula $a \wedge \neg a$ can be modeled by a paraconsistent interpretation that assigns the truth value *inconsistent* to the atom a.

$$\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots, \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m.$$

$\mathbf{K}\,H$ is called the *rule head*, and the sets $\{\mathbf{K}\,A_i\}$ and $\{\mathbf{not}\,B_j\}$ are called the *positive body* and the *negative body*, respectively. A rule r is *positive* if $m = 0$, and r is a *fact* if $n = m = 0$. A *program* $\mathcal{P}$ is a finite set of MKNF rules. A *hybrid MKNF knowledge base* (or *hybrid MKNF KB*) $\mathcal{K}$ is a pair $(\mathcal{O}, \mathcal{P})$.

To make LPs containing variables decidable, usually a *safeness condition* is imposed, which demands that all variables in a rule must occur in the positive body of the rule [Llo87]. However, the mentioned condition is not sufficient in the case of hybrid MKNF knowledge bases for the purpose of ensuring *domain independence*, according to Motik and Rosati [MR07]. As a result, a stronger safety condition has to be imposed that restricts the use of rules to those individuals that are explicitly introduced in the knowledge base. This condition has been introduced by Motik et al. [MSS05] and is expressed formally by means of the following definition from [MR07].

Definition 2.14 (DL-safety [MR07])**.** An MKNF rule r is *DL-safe* if every variable in r occurs in at least one non-DL-atom $\mathbf{K}\,B$ occurring in the body of r. A hybrid MKNF knowledge base $\mathcal{K}$ is DL-safe if all its rules are DL-safe.

The hybrid knowledge base we have introduced in Example 1.1 is DL-safe since the only variable x appearing in each program-rule also occurs in the non-DL-atom good(x) in all of the rules. According to Motik and Rosati, DL-safety can always be achieved by introducing a new predicate in the body of all rules which are not DL-safe that binds the variables occurring in the rule to all individuals that appear in the knowledge base [MR10]. Hence, we presuppose DL-safety when talking about hybrid MKNF knowledge bases in the remainder of this thesis.

Note that the syntax for hybrid MKNF knowledge bases that we have just introduced differs from the syntax of MKNF formulas. For this reason, it is necessary to define a translation of hybrid MKNF knowledge bases into their equivalent version expressed in first-order logic with equality and the two modal operators **K** and **not**. Given an ontology $\mathcal{O}$ expressed in some DL that satisfies the previously stated requirements, we denote the corresponding first-order translation (as defined e.g. in [BCMNP03]) by $\pi(\mathcal{O})$. The translation for hybrid MKNF knowledge bases is presented in the following definition, which represents an adaptation of the definitions provided in [KAH11; MR07].

Definition 2.15 (Translation of a hybrid MKNF knowledge base)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid MKNF knowledge base. We extend π to MKNF rules r, $\mathcal{P}$, and $\mathcal{K}$ as follows, where $\vec{x}$ is the vector of the free variables of r.

$$\pi(r) = \forall \vec{x} : (\mathbf{K}\,A_1 \wedge \ldots \wedge \mathbf{K}\,A_n \wedge \mathbf{not}\,B_1 \wedge \ldots \wedge \mathbf{not}\,B_m \supset \mathbf{K}\,H); \text{if } r \text{ is not a fact.}$$

$$\pi(r) = \top \supset \mathbf{K}\,H; \text{if } r \text{ is a fact.}$$

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

Note that r cannot contain any variables if it is a fact due to the DL-safety condition that we impose on hybrid MKNF knowledge bases. In the previous approaches that use a similar translation of hybrid MKNF knowledge bases into MKNF formulas (e.g. [KAH11; MR10]), the ontology component is also prefixed by the modal operator **K**, such that $\pi(\mathcal{K}) = \mathbf{K}\,\pi(\mathcal{O}) \wedge \pi(\mathcal{P})$. However, we leave the operator out in our definition. In terms of the semantics assigned to hybrid MKNF knowledge bases by Motik and Rosati, and Knorr et al., both definitions are equivalent as a consequence of Proposition 2 in [KAH11]. Like Knorr et al. [KAH11], we will refer to the translation of a knowledge base $\pi(\mathcal{K})$ just by $\mathcal{K}$ whenever there is no ambiguity.

Due to DL-safety, it is always possible to ground a hybrid MKNF knowledge base, such that its two- and three-valued models (which we will introduce in section 2.4.2) coincide with the models of its ground version. The grounding of a hybrid MKNF knowledge base is defined as follows, according to Knorr et al.

Definition 2.16 (Ground hybrid MKNF knowledge base [KAH11])**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid MKNF knowledge base. 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.

2.4.2 Two- and Three-Valued Hybrid MKNF Semantics

So far, two major approaches that assign a semantics to hybrid MKNF knowledge bases have been developed in the literature. The first approach has been published by Motik and Rosati when they introduced hybrid MKNF knowledge in [MR07], and it has been further elaborated in [MR10]. Their semantics employs two truth values and it is closely related to the SMS. Moreover, it has already been extended in order to allow for paraconsistent reasoning by Huang et al. [HLH11]. The second approach works with three truth values, is founded in the WFS for LPs and has been introduced by Knorr et al. in [KAH08]. A more extensive treatment of the three-valued semantics for hybrid MKNF knowledge bases has been published in [KAH11]. Corresponding to their respective base formalism, the latter approach is superior in terms of computational efficiency, while the former approach provides a stronger semantics (cf. the discussion w.r.t. the SMS and the WFS in Section 2.3).

Both, the two-valued and the three-valued MKNF semantics take first-order interpretations as basic constituents for assigning a meaning to hybrid MKNF knowledge bases. They are defined in the usual way as follows. A first-order interpretation I over a signature $\Sigma = (\Sigma_c, \Sigma_f, \Sigma_p)$ and a universe Δ assigns

- an element of the universe $a^I \in \Delta$ to every constant symbol a in Σ_c,
- a function $f^I : \Delta^n \to \Delta$ to every n-ary function symbol f in Σ_f, and

- a relation $P^I \subseteq \Delta^n$ to every n-ary predicate symbol P in Σ_p [MR10].

A variable-free term $t = f(s_1, \ldots, s_n)$ is interpreted recursively s.t. $t^I = f^I(s_1^I, \ldots, s_n^I)$. Additionally, Motik and Rosati point out that two different issues arise regarding the hybrid MKNF semantics when arbitrary universes are considered as in the original definition of the logic of MKNF by Lifschitz [Lif91] (for a discussion of these problems and concrete examples refer to [KAH11; MR10]). To avoid these problems, they impose the *Standard Name Assumption* on first-order interpretations applied in their framework. Here, we recite the definition of the Standard Name Assumption as presented in Definition 3.1 in [MR10] by Motik and Rosati.

Definition 2.17 (Standard Name Assumption [MR10])**.** 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 [Fit96].

In Proposition 3.2 of [MR10], the authors prove that satisfiability of a first-order formula w.r.t. a general first-order model and satisfiability w.r.t. a model that adheres to the Standard Name Assumption are equivalent. As a result, we assume in this thesis that the Standard Name Assumption is employed when talking about first-order interpretations. Moreover, in the following we suppose that a signature Σ and a universe Δ are implicitly given in this case as well.

Additionally, requiring DL-safety alone is not enough to ensure decidability of hybrid MKNF knowledge bases. Following the approaches in [KAH11; MR10], we constrain ourselves to function-free first-order logic, not allowing function symbols in hybrid MKNF knowledge bases, since only then decidability is guaranteed.

In order to provide a *possible world structure* for interpreting the modal **K**- and the modal **not**-operator respectively, Motik and Rosati define so-called *MKNF structures*, which are used to evaluate closed MKNF formulas in the two-valued semantics [MR10]. They are defined in the following manner.

Definition 2.18 (MKNF structure [MR10])**.** An *MKNF structure* is a triple (I, M, N), where I is a first-order interpretation, and M and N are non-empty sets of first-order interpretations.

The authors define satisfaction of a closed MKNF formula by an MKNF structure recursively w.r.t. the structure of the formula in the following way [MR07].

Definition 2.19 (Satisfaction w.r.t. an MKNF structure [MR07])**.** Let (I, M, N) be an MKNF structure and φ, φ_1 and φ_2 MKNF formulas. Then:

$$
\begin{array}{ll}
(I, M, N) \models P(t_1, \ldots, t_n) & \text{iff } (t_1^I, \ldots, t_n^I) \in P^I \\
(I, M, N) \models \neg\varphi & \text{iff } (I, M, N) \not\models \varphi \\
(I, M, N) \models \varphi_1 \wedge \varphi_2 & \text{iff } (I, M, N) \models \varphi_1 \text{ and } (I, M, N) \models \varphi_2 \\
(I, M, N) \models \exists x : \varphi & \text{iff } (I, M, N) \models \varphi[\alpha/x] \text{ for some } \alpha \in \Delta \\
(I, M, N) \models \mathbf{K}\varphi & \text{iff } (J, M, N) \models \varphi \text{ for all } J \in M \\
(I, M, N) \models \mathbf{not}\,\varphi & \text{iff } (J, M, N) \not\models \varphi \text{ for some } J \in N
\end{array}
$$

Note that closed MKNF formulas that do not contain any modal operator are solely interpreted w.r.t. the first component of an MKNF structure. The set M contains all *possible worlds*, represented by first-order interpretations, that are used for the evaluation of modal **K**-formulas, and N contains the possible worlds w.r.t. which the modal **not**-operator is evaluated. The interpretation of the other (first-order) operators is straightforward. It is necessary to interpret each kind of modal operator w.r.t. a different set of first-order interpretations in order to achieve the non-monotonic behavior of the formalism. This becomes obvious when considering the following definition of *MKNF models*, which is based on the definitions provided by Motik and Rosati in [MR07] and [MR10].

Definition 2.20 (MKNF model [MR07; MR10])**.** An *MKNF interpretation* M is a non-empty set of first-order interpretations. Any MKNF interpretation M is an *MKNF model* for a given closed MKNF formula φ if and only if

(1) $(I, M, M) \models \varphi$ for all $I \in M$ and

(2) $(I', M', M) \not\models \varphi$ for each $M \subset M'$ and some $I' \in M'$.

According to Definition 2.20, the evaluation of modal **not**-atoms is first fixed w.r.t. some MKNF interpretation such that the MKNF structure (I, M, M) satisfies the formula for all I in M, and then it is checked whether the set M can be increased to a set M' such that (I', M', M) still satisfies the formula for all I' in M'. If M is already maximal in the described sense, M is an MKNF model for the formula. In this way, the previous definition imposes a minimality condition on MKNF models because the larger the set M is, the smaller is the set of formulas which are satisfied by (I, M, M) for all I in M. Note that this is very similar to the definition of stable models presented in section 2.3 where the evaluation of default negated program atoms is fixed w.r.t. an interpretation and afterwards it is checked whether the interpretation is a minimal model of the resulting positive program. As in the case of the SMS, in general, there can be zero, one or several MKNF models for an MKNF formula.

The three-valued MKNF semantics published by Knorr et al. extend the semantics of Motik and Rosati by introducing the additional truth value *undefined*, which is assigned whenever there is no evidence for a modal **K**-atom being either *true* or *false*, according to Knorr et al. [KAH11]. In order to accommodate for the new truth value in the evaluation

of MKNF formulas, Knorr et al. extend the definition of MKNF structures to so-called *three-valued MKNF structures*. We recall their definition from [KAH11] here.

Definition 2.21 (Three-valued MKNF structure [KAH11])**.** A *three-valued MKNF structure* $(I, \mathcal{M}, \mathcal{N})$ consists of a first-order interpretation I and two pairs $\mathcal{M} = \langle M, M_1 \rangle$ and $\mathcal{N} = \langle N, N_1 \rangle$ of sets of first-order interpretations where $M_1 \subseteq M$ and $N_1 \subseteq N$. An MKNF structure is called *total* if $\mathcal{M} = \langle M, M \rangle$ and $\mathcal{N} = \langle N, N \rangle$.

In contrast to MKNF structures, three-valued MKNF structures contain two pairs of sets of first-order interpretations instead of just two sets of first-order interpretations as possible world structures for the two modal operators. While those first-order atoms which are *true* in first-order interpretations in the first component of each pair represent those atoms which are known to be *true* in the knowledge base, those which are *true* in all interpretations in the second component are known to be *true* or *undefined*. The subset conditions between the sets M and M_1, as well as between N and N_1, are imposed since no first-order atom should be allowed to be known to be *true* and known to be neither *true* nor *undefined* at the same time as paraconsistency is not considered in the approach by Knorr et al. This role of the sets $\mathcal{M}$ and $\mathcal{N}$ is reflected in the following definition of the truth-evaluation of a closed MKNF formula in a three-valued MKNF structure, which is also identical to the definition in [KAH11].

Definition 2.22 (Evaluation in a three-valued MKNF structure [KAH11])**.** Let $(I, \mathcal{M}, \mathcal{N})$ be a three-valued MKNF structure and $\{\mathbf{t}, \mathbf{u}, \mathbf{f}\}$ the set of truth values with the order $\mathbf{f} < \mathbf{u} < \mathbf{t}$, where the operator $\max$ (resp. $\min$) chooses the greatest (resp. least) element with respect to this order. We define:

- $(I, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \begin{cases} \mathbf{t} & \text{iff } (t_1^I, \ldots, t_n^I) \in P^I \\ \mathbf{f} & \text{iff } (t_1^I, \ldots, t_n^I) \notin P^I \end{cases}$

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

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

- $(I, \mathcal{M}, \mathcal{N})(\varphi_1 \supset \varphi_2) = \mathbf{t}$ iff $(I, \mathcal{M}, \mathcal{N})(\varphi_2) \geq (I, \mathcal{M}, \mathcal{N})(\varphi_1)$ and $\mathbf{f}$ otherwise

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

- $(I, \mathcal{M}, \mathcal{N})(\mathbf{K}\varphi) = \begin{cases} \mathbf{t} & \text{iff } (J, \langle M, M_1 \rangle, \mathcal{N})(\varphi) = \mathbf{t} \text{ for all } J \in M \\ \mathbf{f} & \text{iff } (J, \langle M, M_1 \rangle, \mathcal{N})(\varphi) = \mathbf{f} \text{ for some } J \in M_1 \\ \mathbf{u} & \text{otherwise} \end{cases}$

- $(I, \mathcal{M}, \mathcal{N})(\mathbf{not}\,\varphi) = \begin{cases} \mathbf{t} & \text{iff } (J, \mathcal{M}, \langle N, N_1 \rangle)(\varphi) = \mathbf{f} \text{ for some } J \in N_1 \\ \mathbf{f} & \text{iff } (J, \mathcal{M}, \langle N, N_1 \rangle)(\varphi) = \mathbf{t} \text{ for all } J \in N \\ \mathbf{u} & \text{otherwise} \end{cases}$

Knorr et al. define their semantics such that MKNF formulas not containing any modal operators still obtain a two-valued interpretation. In this way, the authors are able to show the faithfulness of their approach w.r.t. classical ontology semantics whenever a hybrid MKNF knowledge base is given in which the program component is empty. Like in the two-valued approach by Motik and Rosati, the two modal operators **K** and **not** are defined symmetrically such that $\neg$**K** is equivalent to **not** whenever both operators are interpreted w.r.t. the same set of possible worlds [KAH11]. Due to the new truth value $\mathbf{u}$ designating *undefined*, knowledge is minimized in the three-valued semantics w.r.t. to the order $\mathbf{f} < \mathbf{u} < \mathbf{t}$, i.e. a modal **K**-formula is *false* if it is *false* in one possible world, it is *undefined* if it is not *false* in any world but *undefined* in at least one world, and it is *true* if it is *true* in all worlds. On the basis of the previously presented evaluation function, Knorr et al. define *three-valued MKNF models* similar to the definition of MKNF models used by Motik and Rosati. However, knowledge minimization now is performed w.r.t. to the extended order just mentioned. In the following, we will refer to the order used for minimizing two- three- and six-valued MKNF models by the term *knowledge minimization order* (or *minimization order*). This notion will play an important role in the discussion of our own approach in the next chapter. The following definition is based on the corresponding definitions provided by the authors in [KAH08] and [KAH11].

Definition 2.23 (Three-valued MKNF model [KAH08; KAH11])**.** An *MKNF interpretation pair* (M, N) consists of two MKNF interpretations M, N with $\emptyset \subset N \subseteq M$. Any MKNF interpretation pair (M, N) is a *three-valued MKNF model* for a given closed MKNF formula φ if and only if

(1) $(I, \langle M, N \rangle, \langle M, N \rangle)(\varphi) = t$ for all $I \in M$ and

(2) $(I', \langle M', N' \rangle, \langle M, N \rangle)(\varphi) \neq t$ for some $I' \in M'$ and each interpretation pair (M', N') with $M \subseteq M'$ and $N \subseteq N'$ where at least one of the inclusions is proper.

Although there are possibly several three-valued MKNF models of an MKNF formula like in the semantics of Motik and Rosati, at least one three-valued MKNF model always exists and moreover, it is always possible to single out one specific model, the *well-founded MKNF model*, in which as much knowledge as possible is left *undefined*. We will discuss this skeptical form of reasoning in more detail when presenting the counterpart of a well-founded model in our approach. Knorr et al. also show that every total three-valued MKNF model corresponds exactly to one two-valued MKNF model [KAH11]. To conclude this section, we illustrate the difference between the two-valued and the three-valued MKNF semantics as well as their connection in the following example, which is inspired by [KAH11].

Example 2.24 (Two- and Three-valued MKNF semantics)**.** Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$ containing only two MKNF rules.

$$\begin{aligned} \mathbf{K}\,\mathsf{risk(food)} &\leftarrow \mathbf{not}\,\mathsf{safe(food)}. \\ \mathbf{K}\,\mathsf{safe(food)} &\leftarrow \mathbf{not}\,\mathsf{risk(food)}. \end{aligned}$$

The knowledge base expresses that a certain food is either a risk or it is safe, but not both. We abbreviate food by f, risk by r, and safe by s.

Regarding the two-valued MKNF semantics, the MKNF interpretation $M = \{\{\mathsf{r(f)}\}\}$ fulfills the first condition of Definition 2.20. Yet, this MKNF interpretation is not maximal since the MKNF interpretation $M' = \{\{\mathsf{r(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}$, which is a superset of M, violates condition (2) of that definition (where modal **not**-atoms are still interpreted w.r.t. M). However, M' satisfies both conditions of Definition 2.20 and hence, it is a two-valued MKNF model of $\mathcal{K}_G$. The modal **K**-atom $\mathbf{K}\,\mathsf{r(f)}$ is *true* in M', while $\mathbf{K}\,\mathsf{s(f)}$ is *false*. The only other MKNF model of $\mathcal{K}_G$ is the two-valued MKNF interpretation $\{\{\mathsf{s(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}$, in which $\mathbf{K}\,\mathsf{s(f)}$ is *true*, and $\mathbf{K}\,\mathsf{r(f)}$ is *false*.

W.r.t. the three-valued MKNF semantics, there are two total three-valued MKNF models of $\mathcal{K}_G$ that correspond to the two-valued MKNF models, respectively. They are represented by the MKNF interpretation pairs $(\{\{\mathsf{r(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}, \{\{\mathsf{r(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\})$ and $(\{\{\mathsf{s(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}, \{\{\mathsf{s(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\})$, and evaluate the **K**-atoms $\mathbf{K}\,\mathsf{r(f)}$ and $\mathbf{K}\,\mathsf{s(f)}$ to the same truth values as the two-valued MKNF models. However, $\mathcal{K}_G$ provides no evidence for either $\mathbf{K}\,\mathsf{r(f)}$ or $\mathbf{K}\,\mathsf{s(f)}$ being *true*. Now, if both modal **K**-atoms are evaluated to *undefined* by an MKNF interpretation pair, then, after fixing the evaluation of the modal **not**-atoms in condition (2) of Definition 2.23, each rule is satisfied in case the head of the rule is evaluated to *true* or *undefined* according to Definition 2.22. Interpreting both with *undefined* results in a minimal interpretation w.r.t. the order $\mathbf{f} < \mathbf{u} < \mathbf{t}$, such that the MKNF interpretation pair interpreting both **K**-atoms with *undefined*, results in a third three-valued MKNF model of $\mathcal{K}_G$. It is represented by the pair $(\{\emptyset, \{\mathsf{r(f)}\}, \{\mathsf{s(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}, \{\{\mathsf{r(f)}, \mathsf{s(f)}\}\})$. Note that for example the MKNF interpretation pair $(\{\{\mathsf{r(f)}\}, \{\mathsf{s(f)}\}, \{\mathsf{r(f)}, \mathsf{s(f)}\}\}, \{\{\mathsf{r(f)}, \mathsf{s(f)}\}\})$ is not a three-valued MKNF model of $\mathcal{K}_G$. Though it would result in the same truth evaluation, also assigning the truth value *undefined* to $\mathbf{K}\,\mathsf{r(f)}$ and $\mathbf{K}\,\mathsf{s(f)}$, the first component of the interpretation pair is not maximal. The third three-valued MKNF model assumes the least number of **K**-atoms to be either *true* or *false*, and it is in fact the unique well-founded MKNF model of $\mathcal{K}_G$ under the three-valued MKNF semantics. ◊

3

Model-Theoretic Characterization

After having discussed the diverse research areas which our approach is connected to, and having introduced several notions which are central for the work developed in this thesis, we are ready to present our main contributions. In this chapter, we first present the details of the model-based definition of the paraconsistent well-founded hybrid MKNF semantics, before turning to the procedural definition in Chapter 4. Moreover, we provide the first two faithfulness results respectively w.r.t. the three-valued MKNF semantics of Knorr et al. [KAH11], and the paraconsistent ontology semantics of $\mathcal{ALC}4$ as defined by Maier et al. [MMH13]. We will also illustrate the functional principles and the motivations behind the definitions that are presented by a number of intuitive examples.

The paraconsistent semantics developed in this chapter is based on six truth values, where the three-valued MKNF semantics, which is the basis of our approach, is extended by three additional truth values. Alternatively, the six truth values also constitute an extension of the four truth values of Belnap's famous logic $\mathcal{FOUR}$ [BJ77a]. We begin by justifying why six truth values are needed as the foundation for our semantics.

3.1 The Need for Six Truth Values

The first question that has to be asked when developing a many-valued semantics is how many truth values are exactly needed. On the one hand, it has to be possible to capture all information that is required about the "state" of a formula. On the other hand, the introduction of unintuitive or meaningless truth values should be avoided to keep the logic as simple as possible, following the principle of Ockham's razor. For instance, if a user queries a knowledge base for some piece of knowledge and gets the answer that it has the truth value **fcb** (which is part of the nine-valued bilattice shown in Figure 2.2 and

stands for *false with contradictory belief*), it is not easy for the user to understand what this implies.

Since our logic should be able to deal with contradictory information, at least one new truth value *inconsistent*, designated by **b**, has to be introduced in addition to the truth values already present in the approach that we extend [KAH11], namely *true*, *false* and *undefined*. Otherwise, it would not be possible to assign a model to contradictory formulas such that the Principle of Explosion would be implied and inconsistent hybrid MKNF knowledge bases would not be meaningful anymore. In this case, we obtain Belnap's four truth values [BJ77a].

When only taking formulas not containing modal atoms into account, i.e. the first-order formula expressing the ontology component, Knorr et al. only apply the truth values *true* and *false* in order to maintain the classical DL-semantics [KAH11]. In this case, we also want to introduce just as many truth values as needed to obtain the desired behavior. Because the ontology component should be allowed to be inconsistent alone in our framework, i.e. without any interaction with the program component, at least the third truth value **b** also has to be introduced for interpreting MKNF formulas not containing any modal operators. However, as Maier et al. demonstrate [MMH13], the truth value **u** is not required to provide a paraconsistent semantics for the ontology component, and by omitting it even a stronger semantics can be attained.

Now, Knorr et al. identify two different kinds of inconsistencies in their semantics, besides the case where some piece of information is considered *true* and *false* simultaneously it can also be the case that it is *undefined* and *false* at the same time [KAH11], due to the intervention of the Coherence Principle. The latter situation might occur when the body of some MKNF rule is interpreted to be *undefined*, but the head of the rule is derivable to be *false* from the ontology component. Because of the Coherence Principle, the modal **K**-atom in the head of the rule should be *false* in this case. However, this interpretation does not satisfy the rule in the approach by Knorr et al. as a result of the definition of the implication operator used in their semantics [KAH11], which maps a rule with *undefined* body and *false* head to *false* (cf. Definition 2.22). This kind of inconsistency could be dealt with by simply mapping an implication to a designated truth value in the mentioned case. However, for technical reasons, this is not possible in the context of MKNF where knowledge is minimized because *undefinedness* would always be minimized to *falsity*. This issue will be discussed in more detail when we present the interpretation function of the implication operator that we apply in our semantics. As a result, we solve the problem of dealing with the second kind of inconsistency by introducing the further truth value *classically false*, denoted by **cf**, which is assigned in these cases. Although the introduction of the truth value **cf** is mainly technically motivated, it has the desirable side-effect that we are able to distinguish between **K**-atoms which are derivable to be *classically false* by means of the ontology component, and those which are *false* because their positive version is not derivable. Accordingly, our truth value *classically false* expresses a stronger notion of *falsity* than the truth value *false* adopted from the

three-valued MKNF semantics.

Besides the information that some fact could be derived to be *classically false* from the ontology, we also want to provide a user with the information whether some derived fact relies on a contradiction. On the one hand, this is important for a user in order to decide if she can rely on some derived knowledge. On the other hand, propagation of inconsistencies is built into $WFSX_p$ and in order to obtain a formalism that corresponds to $WFSX_p$ in terms of the program semantics, we need to mirror this behavior in some form in the model-theoretic characterization of our semantics. One option would be to simply propagate inconsistencies such that every piece of information that can only be inferred by involving inconsistent knowledge is assigned the truth value *inconsistent* itself. Yet, like in the case of rules with *undefined* body and *false* head, knowledge minimization makes this strategy unfeasible because either all *true* knowledge would be minimized to the truth value *inconsistent* or alternatively, the head of rules with *inconsistent* bodies would be minimized to the truth value *true* such that propagation of inconsistencies fails. This issue will also be elaborated on in Section 3.3. In addition, it is often necessary to differentiate between information that can only be derived by involving inconsistent knowledge and information that is contradictory itself. In order to distinguish between these two kinds of "states" at least one more truth value has to be introduced, which we call *suspiciously true* and is denoted by $\mathbf{st}$[1]. This gives us already the six truth values which form the basis of our paraconsistent well-founded MKNF semantics. As has been discussed, they provide exactly the right granularity for representing the information we want to express in our logic.

Next, we illustrate the usefulness of the six truth values with an example, where we consider which information about the entities in a hybrid MKNF knowledge base a user would like to be provided with by a semantics. After formally introducing the model-theoretic characterization of our semantics, we will come back to this example and demonstrate that the semantics behaves as desired w.r.t. the use-case considered here.

Example 3.1 (The usefulness of having six truth values)**.** Consider the following hybrid MKNF knowledge base.

$$\mathsf{HasCertifiedForwarder} \sqsubseteq \neg\mathsf{IsMonitored}$$

$$\begin{aligned}
\mathbf{K}\,\mathsf{IsMonitored(x)} &\leftarrow \mathbf{K}\,\mathsf{good(x)}, \mathbf{K}\,\mathsf{risk(x)}.\\
\mathbf{K}\,\mathsf{risk(x)} &\leftarrow \mathbf{K}\,\mathsf{good(x)}, \mathbf{not}\,\mathsf{isLabelled(x)}.\\
\mathbf{K}\,\mathsf{isLabelled(x)} &\leftarrow \mathbf{K}\,\mathsf{good(x)}, \mathbf{not}\,\mathsf{risk(x)}.\\
\mathbf{K}\,\mathsf{resolvedRisk(x)} &\leftarrow \mathbf{K}\,\mathsf{good(x)}, \mathbf{K}\,\mathsf{IsMonitored(x)}.\\
\mathbf{K}\,\mathsf{HasCertifiedForwarder(food)} &\leftarrow
\end{aligned}$$

[1]The idea behind the truth value $\mathbf{st}$ is related to the motivation given by Sakama and Inoue for introducing the truth value $\mathbf{st}$ into their six-valued semantics [SI95]. However, we do not introduce its complement $\mathbf{sf}$ here such that $\mathbf{st}$ rather represents a special case of $\mathbf{b}$ in our semantics.

$$\mathbf{K}\,\mathsf{HasCertifiedForwarder(pesticide)} \leftarrow$$
$$\mathbf{K}\,\mathsf{risk(pesticide)} \leftarrow$$
$$\mathbf{K}\,\mathsf{good(food)} \leftarrow$$
$$\mathbf{K}\,\mathsf{good(pesticide)} \leftarrow$$

The second and the third MKNF rule in this knowledge base together formalize that a good either isLabelled or is a risk by means of a recursion through default negation. Further, every good that is a risk IsMonitored, due to the first rule. The fourth rule states that every good that IsMonitored is a resolvedRisk. Moreover, if something is in the class HasCertifiedForwarder, it can be proven by means of the only ontology axiom that it is not the case that it IsMonitored.

Considering a user who wants to query the knowledge base about the truth value of certain **K**-atoms, the truth value *true* should be returned for all program facts since no program fact is contradictory in this knowledge base. Regarding the second and the third rule in the program component, it is either the case that the modal **K**-atom **K** risk(food) is *true* and **K** isLabelled(food) is *false*, or vice versa. Consequently, the truth value *undefined* should be assigned to both modal **K**-atoms to provide the user with the information that it cannot be decided if they are *true* or *false* – at least until further information is available. Consequently, without taking the ontology into account, the **K**-atom **K** IsMonitored(food) should also be *undefined*. However, due to the ontology axiom, ¬IsMonitored(food) can be derived from the ontology component and the user would like to be informed that the fact can be proven not to hold by receiving the answer *classically false* w.r.t. **K** IsMonitored(food). In the case of the good pesticide, **K** risk(pesticide) is contained as a fact in the program component, so that **K** IsMonitored(pesticide) can be derived from the knowledge base. Yet, ¬IsMonitored(pesticide) can also be derived by means of the ontology component, such that the user should be informed that **K** IsMonitored(pesticide) is *inconsistent*. Finally, **K** resolvedRisk(pesticide) can also be derived, but only by involving the *inconsistent* **K**-atom **K** IsMonitored(pesticide). Consequently, a user would like to be informed that **K** resolvedRisk(pesticide) is *suspiciously true*, so that she knows that she has to be careful in using this information. ◊

3.2 The Lattice $\mathcal{SIX}$

We represent the six truth values $\mathbf{b}$ (*inconsistent*), $\mathbf{st}$ (*suspiciously true*), $\mathbf{t}$ (*true*), $\mathbf{f}$ (*false*), $\mathbf{cf}$ (*classically false*) and $\mathbf{u}$ (*undefined*), motivated in the previous section, by means of a lattice, that we call the lattice $\mathcal{SIX}$ and that is shown in Figure 3.1. Among the six truth values, the values $\mathbf{b}$ and $\mathbf{t}$ are the designated truth values of our logic. It is explained below why $\mathbf{st}$ is not defined to be a designated truth value. The lattice $\mathcal{SIX}$ defines two partial orders over the truth values, i.e. the orders $\mathbf{f} < \mathbf{cf} < \mathbf{u} < \mathbf{t}$ and $\mathbf{f} < \mathbf{st} < \mathbf{b} < \mathbf{t}$.

These orders are used in our semantics to define the interpretation functions of the first-order operators. The lattice $\mathcal{SIX}$ can be viewed as an extension of the bilattice $\mathcal{FOUR}$ (cf.

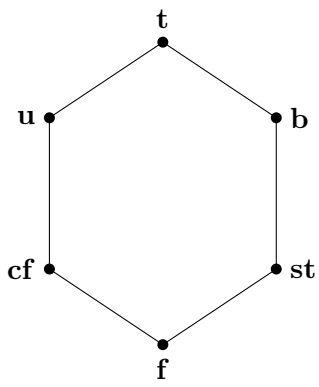

Figure 3.1: The six truth values ordered in the lattice $\mathcal{SIX}$.

Figure 2.1), where just two additional values are introduced below the truth values **u** and **b**, respectively. Note that due to the extension, the resulting algebraic structure does not constitute a bilattice anymore, like it is also the case for the six-valued lattice **VI** used by Sakama and Inoue [SI95] depicted in Figure 2.2. As a result, the notions of a truth- and a knowledge-order are not directly applicable to the lattice $\mathcal{SIX}$. However, the additional truth values **st** and **cf** can also be understood as special cases of the truth values **b** and **f**, respectively, so that the meaning provided by the values **b** and **f** is refined by splitting each into two separate values. They are still closely related since the truth value **st** behaves like **b** in our semantics by being its own complement w.r.t. classical and default negation, and **cf** behaves as if it was **f** apart from the case where it is used to evaluate the head of an MKNF rule with *undefined* body. The increase of the expressive power of our semantics is achieved by slightly adapting the way the two new truth values are treated w.r.t. the implication operator, and by not appointing the value **st** as a designated truth value, in contrast to **b**.

3.3 Interpretation of the Implication Operator

Next, we discuss the implication operator used in our semantics. As described in Section 2.2, in case only the internal implication operator is used, the paraconsistent version of several tractable DLs can be translated into DLs under classical semantics where consequences and tractability can be preserved simultaneously, so that it is possible to use available standard reasoners. Since this property is crucial for potential implementations of our approach, we employ an implication operator (presented in Table 3.1) in our semantics which is defined like internal implication in [MMH13] for the truth values **b**, **t**, **f** and **u**, apart from the assignment $\mathbf{u} \supset \mathbf{f}$[2], which is no longer mapped to a designated

[2]Since we restrict our semantics to three truth values in the case of the ontology component, omitting the truth value *undefined*, the mentioned divergence from the definition of Maier et al. does not interfere with

truth value.

$\supset$	**b**	**st**	**t**	**f**	**cf**	**u**
b	**b**	**t**	**t**	**f**	**f**	**f**
st	**b**	**t**	**t**	**f**	**f**	**f**
t	**b**	**f**	**t**	**f**	**f**	**f**
f	**t**	**t**	**t**	**t**	**t**	**t**
cf	**t**	**t**	**t**	**t**	**t**	**t**
u	**t**	**f**	**t**	**f**	**t**	**t**

Table 3.1: Semantics of the implication operator.

In order to justify the truth evaluation of the assignment $\mathbf{u} \supset \mathbf{f}$, as well as other particular truth value assignments in the interpretation function of the implication operator, we have to take the minimization of models into account, which is a central component of the logic of MKNF. Knowledge minimization in the logic of MKNF enables non-monotonic default reasoning under the CWA by minimizing everything that cannot be derived to *false*. Therefore, the truth value $\mathbf{f}$ has to be the least element of the minimization order used for knowledge minimization in the semantics. As a result, the $\mathbf{K}$-atom in the head of an MKNF rule with *undefined* body would always be minimized to $\mathbf{f}$ if the truth assignment $\mathbf{u} \supset \mathbf{f}$ was mapped to a designated truth value. However, this is not always intended as the following example shows.

Example 3.2. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$ only consisting of a program component.

$$\begin{aligned} \mathbf{K}P(a) &\leftarrow \mathbf{K}Q(a) \\ \mathbf{K}Q(a) &\leftarrow \mathbf{not}\,Q(a) \end{aligned}$$

In this knowledge base, $\mathbf{K}Q(a)$ should be assigned the truth value *undefined* like usual in the WFS due to the recursion through default negation. However, $\mathbf{K}P(a)$ should also be *undefined* as the only MKNF rule in which it occurs in the head has an *undefined* body. $\Diamond$

Consequently, only the truth value $\mathbf{cf}$ can be allowed for the consequent of an implication with *undefined* implicant, and $\mathbf{cf}$ cannot be smaller in the minimization order of our semantics, so that $\mathbf{K}P(a)$ in the previous example is also not minimized to $\mathbf{cf}$.

Furthermore, the minimization of models poses a problem for the propagation of inconsistencies as well. In the minimization order, there cannot be a designated truth value that is smaller than the truth value $\mathbf{t}$ because all facts, and the heads of rules whose body is evaluated to a designated truth value, would be minimized to this value, though they should normally be mapped to *true*. However, in the case of the truth value $\mathbf{st}$, the same strategy we pursue in the case of $\mathbf{cf}$, i.e. defining $\mathbf{st}$ to be greater in the minimization order and forbidding minimization to the truth value $\mathbf{t}$ in case a rule has an *inconsistent* body, does also not work as the following example shows.

the coincidence of our ontology semantics and the semantics defined in [MMH13].

Example 3.3. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$\begin{array}{rcl} \top & \sqsubseteq & \neg P \\ \mathbf{K}\,Q(a) & \leftarrow & \mathbf{K}\,P(a) \\ \mathbf{K}\,R(a) & \leftarrow & \mathbf{K}\,Q(a) \\ \mathbf{K}\,P(a) & \leftarrow & \end{array}$$

In this example, the modal **K**-atom $\mathbf{K}\,P(a)$ is clearly contradictory since the only fact in the program implies that $\mathbf{K}\,P(a)$ holds. At the same time, the classical negation of P is derivable from the ontology component. Consequently, the **K**-atom $\mathbf{K}\,Q(a)$ should be *suspiciously true* in an interpretation satisfying $\mathcal{K}_G$. Moreover, $\mathbf{K}\,R(a)$ should also be mapped to $\mathbf{st}$ as it can only be derived by consulting *inconsistent* information. However, if the truth value $\mathbf{t}$ is smaller than $\mathbf{st}$ in the minimization order, $\mathbf{K}\,Q(a)$ and $\mathbf{K}\,R(a)$ are evaluated to be *true* in every model of $\mathcal{K}_G$ (if this is allowed by the definition of the implication operator). One strategy to solve this problem would be to forbid the assignment of the truth value $\mathbf{t}$ for the **K**-atom in the head of a rule with *inconsistent* or *suspiciously true* body. Yet, assume we add $\mathbf{K}\,Q(a)$ as a second fact to $\mathcal{K}_G$. Then, $\mathbf{K}\,Q(a)$ can be inferred without relying on *inconsistent* knowledge. The same holds for adding the fact $\mathbf{K}\,R(a)$ to the knowledge base. Consequently, the assignments $\mathbf{b} \supset \mathbf{t}$ and $\mathbf{b} \supset \mathbf{st}$ have to be mapped to a designated truth value. Another strategy would be not to define $\mathbf{t}$ to be larger in the minimization order than $\mathbf{st}$. In this case, the correct truth value would be assigned to $\mathbf{K}Q(a)$ and $\mathbf{K}R(a)$ in the original version of $\mathcal{K}_G$ (without the additional facts). Yet, there would still be a problem since all *true* knowledge would also be minimized to *suspiciously true*. ◊

Our solution to the problem described in Example 3.3 consists in defining $\mathbf{st}$ to be smaller in the minimization order than $\mathbf{b}$ and $\mathbf{t}$, but at the same time we do not define $\mathbf{st}$ to be a designated truth value and map implications with the truth assignment $\mathbf{t} \supset \mathbf{st}$ to a non-designated truth value (cf. Table 3.1). However, implications where the implicant is evaluated to $\mathbf{b}$ or $\mathbf{st}$ and the consequent is mapped to $\mathbf{st}$ in an interpretation are still satisfied by this interpretation. In this way, **K**-atoms which are only implied by rules with *inconsistent* or *suspiciously true* bodies are minimized to *suspiciously true*. Though, when the **K**-atom simultaneously occurs as a fact, in the head of a rule with *true* body, or can be derived from the ontology, it is still forced to be *true* (or *inconsistent* if its classical negation is also derivable) in an interpretation satisfying the knowledge base.

As we have discussed in Section 2.3.3, Contradiction Support Detection fails in the case of $WFSX_p$ whenever a program atom that can only be derived from contradictory knowledge is also implied by a rule with *undefined* body. Because we aim to develop a semantics that corresponds to $WFSX_p$ w.r.t. the semantics assigned to the program component alone, we also have to inhibit the propagation of inconsistencies in case a modal **K**-atom appears in the head of a rule whose body is interpreted to be *undefined*

in a model. On the basis of the previous considerations, this can easily be done by also mapping $\mathbf{u} \supset \mathbf{st}$ to a non-designated truth value. As in the case of $\mathbf{t} \supset \mathbf{st}$, this inhibits the minimization to *suspiciously true* since the truth value $\mathbf{st}$ is not allowed for the head in this case (when considering interpretations satisfying these rules). In this way, the desired behavior of the program semantics can be achieved by utilizing the interaction of knowledge minimization and the definition of the implication operator.

3.4 Paraconsistent Truth Evaluation

Next, we have to adapt the definitions of *first-order interpretation* and *MKNF structure* from [KAH11] in order to enable paraconsistent reasoning by extending the semantics with the three new truth values $\mathbf{b}$, $\mathbf{st}$ and $\mathbf{cf}$. Moreover, we have to ensure that knowledge minimization in the model-definition of our semantics is executed w.r.t. a minimization order that satisfies the constraints discussed before. First, we define the notion of a paraconsistent (first-order) interpretation, which extends the two-valued first-order interpretations utilized by Knorr et al. to three-valued interpretations, adding the truth value $\mathbf{b}$. They do not allow for the truth value *undefined* being assigned to a first-order formula and thus, follow the idea that we want to fix inconsistencies in the ontology component where some piece of information is *true* and *false* at the same time, but never where it is neither of them. This corresponds to enforcing the LEM in the paraconsistent DL semantics of Maier et al. [MMH13] and it is considered to be more classical and less paraconsistent in that a stronger entailment relation can be obtained in this way.

Definition 3.4 (Paraconsistent interpretation)**.** Given two first-order interpretations I and I_1, the pair $\mathcal{I} = \langle I, I_1 \rangle$, such that $I_1 \subseteq I$, is called a *paraconsistent interpretation* (or *p-interpretation*).

The idea of the subset relation on interpretation pairs is that, whenever an atom is interpreted to *true* in the first-order interpretation I_1, then it is also *true* in the interpretation I. Intuitively, I indicates what is *true* and *false*, while the additional interpretation I_1 only designates for each (*true*) element in I whether it is actually *inconsistent* (or not). We can intersect such p-interpretations simply component-wise to obtain on which pieces of information they coincide. Given a set M of paraconsistent interpretations $\mathcal{I}_i = \langle I_i, I'_i \rangle$, we define $\bigcap M = \langle \bigcap I_i, \bigcap I'_i \rangle$. It is straightforward to see that the result $\bigcap M$ is indeed again a p-interpretation.

The notion of a three-valued MKNF structure as presented in Definition 2.21 can be adjusted as follows, now using sets of p-interpretations instead of sets of first-order interpretations.

Definition 3.5 (Paraconsistent MKNF structure)**.** A *paraconsistent MKNF structure* (or *p-structure*) $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ consists of a p-interpretation $\mathcal{I}$ and two pairs $\mathcal{M} = \langle M, M_1 \rangle$ and $\mathcal{N} = \langle N, N_1 \rangle$ of sets of p-interpretations. A p-structure is called *total* if $\mathcal{M} = \langle M, M \rangle$ and $\mathcal{N} = \langle N, N \rangle$.

Note that the conditions $M_1 \subseteq M$ and $N_1 \subseteq N$ imposed in Definition 2.21 are not applied anymore in the case of p-structures. Like in the three-valued hybrid MKNF semantics, we define the truth values assigned to modal **K**- and **not**-formulas in terms of the interpretations contained in the sets M_1 and M, and N_1 and N respectively. For example, recall that in the semantics defined by Knorr et al. a modal **K**-formula is *true* if and only if it is *true* in all first-order interpretations in M, *undefined* if and only if it is *true* in all interpretations in M_1 and *false* in some interpretation in M, and it is *false* if and only if it is *false* in some interpretation in M_1 (cf. Definition 2.22). Due to the subset condition in Definition 2.21, the definition does not leave space for a fourth truth value. Moreover, maximizing the sets M and M_1 in Definition 2.20 leads to a minimization in the order $\mathbf{f} < \mathbf{u} < \mathbf{t}$. Since we have to define six different truth values and there are certain restrictions w.r.t. the minimization orders that can be imposed on them, it is necessary to drop the subset conditions here.

We can now define the evaluation of MKNF formulas in p-structures. The operators $\wedge$ and $\vee$ are defined respectively to be the join and meet operation in the lattice $\mathcal{SIX}$, i.e. w.r.t. to the partial orders $\mathbf{f} < \mathbf{cf} < \mathbf{u} < \mathbf{t}$ and $\mathbf{f} < \mathbf{st} < \mathbf{b} < \mathbf{t}$. For the semantics of the implication operator, we rely on the definition provided in Table 3.1.

Definition 3.6 (Six-valued interpretation of MKNF formulas)**.** Let $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ be a p-structure and φ, φ_1, and φ_2 MKNF formulas. We define:

$$(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \begin{cases} \mathbf{b} & \text{iff } (t_1^I, \ldots, t_n^I) \in P^I, (t_1^{I_1}, \ldots, t_n^{I_1}) \in P^{I_1} \\ \mathbf{t} & \text{iff } (t_1^I, \ldots, t_n^I) \in P^I, (t_1^{I_1}, \ldots, t_n^{I_1}) \notin P^{I_1} \\ \mathbf{f} & \text{iff } (t_1^I, \ldots, t_n^I) \notin P^I, (t_1^{I_1}, \ldots, t_n^{I_1}) \notin P^{I_1} \end{cases}$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\top) = \mathbf{t} \qquad (\mathcal{I}, \mathcal{M}, \mathcal{N})(\bot) = \mathbf{f}$$

$$(\mathcal{I}, \mathcal{M}, \mathcal{N})(\neg\varphi) = \begin{cases} \mathbf{b} & \text{iff } (I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{b} \\ \mathbf{st} & \text{iff } (I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{st} \\ \mathbf{t} & \text{iff } (I, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{f}, \mathbf{cf}\} \\ \mathbf{f} & \text{iff } (I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t} \\ \mathbf{u} & \text{iff } (I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{u} \end{cases}$$

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

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

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

$$(\mathcal{I},\mathcal{M},\mathcal{N})(\mathbf{K}\,\varphi) = \begin{cases} \mathbf{b} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{b} \\ \mathbf{st} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in M_1}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{f} \\ \mathbf{t} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in M_1}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)\neq\mathbf{f} \\ \mathbf{f} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{f} \\ & \text{s.t. } \exists\mathcal{J}\in M \text{ with } (\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in M_1}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{f} \\ \mathbf{cf} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{f} \\ & \text{s.t. } \nexists\mathcal{J}\in M \text{ with } (\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{t} \\ \mathbf{u} & \text{iff } (\bigcap_{\mathcal{J}\in M}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{f} \\ & \text{s.t. } \exists\mathcal{J}\in M \text{ with } (\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in M_1}\mathcal{J},\langle M,M_1\rangle,\mathcal{N})(\varphi)\neq\mathbf{f} \end{cases}$$

$$(\mathcal{I},\mathcal{M},\mathcal{N})(\mathbf{not}\,\varphi) = \begin{cases} \mathbf{b} & \text{iff } (\bigcap_{\mathcal{J}\in N}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{b} \\ \mathbf{st} & \text{iff } (\bigcap_{\mathcal{J}\in N}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in N_1}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{f} \\ \mathbf{t} & \text{iff } (\bigcap_{\mathcal{J}\in N}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{f} \\ & [\text{ s.t. } \exists\mathcal{J}\in N \text{ with } (\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in N_1}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{f})] \text{ or} \\ & [\text{ s.t. } \nexists\mathcal{J}\in N \text{ with } (\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{t}] \\ \mathbf{f} & \text{iff } (\bigcap_{\mathcal{J}\in N}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in N_1}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)\neq\mathbf{f} \\ \mathbf{u} & \text{iff } (\bigcap_{\mathcal{J}\in N}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{f} \\ & \text{s.t. } \exists\mathcal{J}\in N \text{ with } (\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)=\mathbf{t} \text{ and} \\ & (\bigcap_{\mathcal{J}\in N_1}\mathcal{J},\mathcal{M},\langle N,N_1\rangle)(\varphi)\neq\mathbf{f} \end{cases}$$

As already hinted, the truth values **st** and **cf** behave like **b** and **f**, respectively, under classical and default negation according to the previous definition. Furthermore, the modal **K**- and **not**-operator are defined symmetrically like in the previous approaches by Motik and Rosati, and Knorr et al., such that $\mathbf{not}\,\varphi$ is equivalent to $\neg\mathbf{K}\,\varphi$, given an MKNF formula φ and provided that $\mathcal{M}=\mathcal{N}$. In the minimization order induced by the definition of the **K**-operator, the truth values **t**, **u** and **f** have the order $\mathbf{f}<\mathbf{u}<\mathbf{t}$ identical to the three-valued semantics by Knorr et al. [KAH11], and **f** is the least element among all truth values. Moreover, we have that $\mathbf{st}<\mathbf{t}$ and $\mathbf{st}<\mathbf{b}$, as well as $\mathbf{f}<\mathbf{cf}$ and $\mathbf{f}<\mathbf{u}$. However, the truth values **t** and **b**, as well as **cf** and **u** are incomparable in the resulting minimization order, effectively making it a partial order over the truth values. For this reason an additional condition has to be introduced when we define *paraconsistent MKNF models* in the next section.

3.5 Paraconsistent Models for Hybrid MKNF Knowledge Bases

Next, we adapt interpretation pairs and paraconsistent satisfaction, on which our definition of paraconsistent MKNF models is based. The following definition is similar to Definition 8 in [KAH11].

Definition 3.7 (Paraconsistent interpretation pair)**.** A *paraconsistent interpretation pair* (or *p-interpretation pair*) (M, N) consists of two non-empty sets M, N of p-interpretations. A p-interpretation pair *paraconsistently satisfies* (or *p-satisfies*) a closed MKNF formula φ, written $(M, N) \models_p \varphi$, if and only if

$$(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\varphi) \in \{\mathbf{b}, \mathbf{t}\}$$

for each $\mathcal{I} \in M \cup N$. If $M = N$, then the p-interpretation pair (M, N) is called *total*. If there exists a p-interpretation pair p-satisfying φ, then φ is *p-satisfiable*.

As discussed in the previous section, a closed MKNF formula is not p-satisfied by a p-interpretation pair if it is mapped to the truth value $\mathbf{st}$ by the p-interpretation pair in order to avoid minimization to this truth value in certain cases. Like in the two- and three-valued MKNF semantics, the modal **K**- and **not**-operator are evaluated w.r.t. the same pair of sets in the case of p-satisfaction. In order to be p-satisfied by a p-interpretation pair (M, N), we require that a formula has to be *inconsistent* or *true* in every p-structure $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)$, where the p-interpretation $\mathcal{I}$ may vary over both sets M and N, in contrast to the definition by Knorr et al. where only interpretations in the former set are checked. This is necessary to ensure that formulas entailed by the knowledge base cannot be interpreted to be *false* by any p-interpretation in N. In the approach by Knorr et al., this is ensured by imposing the subset conditions in the definition of three-valued MKNF structures.

According to [MR06], an MKNF formula is called *subjective* if all first-order atoms only appear within the scope of a modal **K**- or **not**-operator within the formula. The evaluation of a subjective MKNF formula in a p-structure does not depend on the first component of the p-structure, the p-interpretation $\mathcal{I}$. Consequently, we can assume that $\mathcal{I}$ is arbitrary in this case, and we will do so in the remainder of this thesis without referring to each $\mathcal{I} \in M \cup N$ and by replacing $\mathcal{I}$ by the symbol $*$ in the respective p-structure.

Alternatively, if an MKNF formula does not contain any modal atoms (i.e. it is equivalent to a first-order formula), the evaluation of the formula in some p-structure does not depend on the second and third component of the p-structure, the sets of p-interpretations $\mathcal{M}$ and $\mathcal{N}$. Therefore, we can neglect $\mathcal{M}$ and $\mathcal{N}$ in this case and define p-satisfaction of a first-order formula and paraconsistent entailment between two first-order formulas merely in terms of the first component of a p-structure $(\mathcal{I}, \mathcal{M}, \mathcal{N})$. This is done in the following definition of the first-order equivalent of p-satisfaction.

Definition 3.8 (Paraconsistent satisfaction and entailment)**.** Let φ and ψ be closed first-order formulas. A p-interpretation $\mathcal{I} = \langle I, I_1 \rangle$ *paraconsistently satisfies* (or *p-satisfies*) φ, written $\mathcal{I} \models_p \varphi$, if and only if

$$(\langle I, I_1 \rangle, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{b}, \mathbf{t}\}$$

for arbitrary $\mathcal{M}$ and $\mathcal{N}$. If $(\mathcal{I}, \mathcal{M}, \mathcal{N})(\psi) \in \{\mathbf{b}, \mathbf{t}\}$ for all p-interpretations $\mathcal{I}$ such that $(\mathcal{I}, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{b}, \mathbf{t}\}$, then φ *paraconsistently entails* (or *p-entails*) ψ, written $\varphi \models_p \psi$.

Because of the additional condition that has to be introduced in our definition of paraconsistent MKNF models in Definition 3.10 (due to the incomparability of certain truth value in the minimization order), our notion of models is not applicable to arbitrary MKNF formulas, but only to those which exhibit the constrained syntax of hybrid MKNF knowledge bases introduced in Definition 2.13. Since we are first and foremost interested in developing a paraconsistent semantics for hybrid knowledge bases, this does no constitute a major drawback. As a result, we will only consider hybrid MKNF knowledge bases in the rest of this thesis, not considering arbitrary MKNF formulas.

Next, we adopt the following definition from Knorr et al., which eases the formulation of subsequent definitions and propositions. The first part of this definition is adopted from Definition 13 in [KAH11], the second part is identical to the second part of Definition 14 in the same publication.

Definition 3.9 (Objective knowledge of a set of **K**-atoms [KAH11])**.** Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground hybrid MKNF knowledge base. The *set of* **K***-atoms of* $\mathcal{K}_G$, written $\mathsf{KA}(\mathcal{K}_G)$, is the smallest set that contains (i) all ground **K**-atoms occurring in $\mathcal{P}_G$, and (ii) a modal atom $\mathbf{K}\,\xi$ for each ground modal atom $\mathbf{not}\,\xi$ occurring in $\mathcal{P}_G$. For a subset S of $\mathsf{KA}(\mathcal{K}_G)$, the *objective knowledge* of S w.r.t. $\mathcal{K}_G$ is the set of first-order formulas $\mathsf{OB}_{\mathcal{O},S} = \{\pi(\mathcal{O})\} \cup \{\xi \mid \mathbf{K}\xi \in S\}$.

Now, we are ready to adapt the definition of three-valued MKNF models, first introduced in [KAH08], to our approach by defining paraconsistent MKNF models next.

Definition 3.10 (Paraconsistent MKNF model)**.** Let (M, N) be a p-interpretation pair and $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ a hybrid MKNF knowledge base. Any p-interpretation pair (M, N) is a *paraconsistent MKNF model* (or *p-model*) of $\mathcal{K}$ if and only if

(1) $(M, N) \models_p \mathcal{K}$,

(2) for every p-interpretation pair (M', N') with $M \subseteq M'$ and $N \subseteq N'$ where at least one of the inclusions is proper, there is $\mathcal{I}' \in M' \cup N'$ s.t. $(\mathcal{I}', \langle M', N' \rangle, \langle M, N \rangle)(\mathcal{K}) \notin \{\mathbf{b}, \mathbf{t}\}$, and

(3) for every $\mathbf{K}\xi \in \mathsf{KA}(\mathcal{K}_G)$ it holds that $(*, \langle M, N \rangle, \langle M, N \rangle)(\mathbf{K}\xi) \in \{\mathbf{b}, \mathbf{cf}\}$ if and only if $\mathsf{OB}_{\mathcal{O},\{\mathbf{K}\xi' \mid (*,\langle M,N \rangle,\langle M,N \rangle)(\mathbf{K}\xi') \in \{\mathbf{b},\mathbf{st},\mathbf{t}\}\}} \models_p \neg\xi$.

The conditions (1) and (2) of the previous definition resemble the conditions presented in Definition 2.23. In particular, condition (2) ensures that knowledge in p-models is minimized w.r.t. the partial minimization order discussed before.

As mentioned already, due to our definition of the modal **K**-operator the truth values **b** and **t**, as well as **cf** and **u** are still incomparable in the resulting partial minimization order. The purpose of condition (3) in the previous definition is to remove certain undesired p-models of some hybrid MKNF knowledge bases, which are possible due to the incomparability of these truth values w.r.t. the knowledge minimization performed in condition (2). In the first case, e.g. a ground hybrid knowledge base $\mathcal{K}_G$ just consisting of one program fact $\mathbf{K}\,H$ would also have a p-model where this fact is evaluated to *inconsistent* if condition (3) was not added to the definition of p-models. The same is true for the head of MKNF rules where the body is evaluated to *true*. Due to condition (3), the mentioned p-model is not allowed for $\mathcal{K}_G$ anymore because $\neg H$ is not p-entailed by the ontology together with all first-order atoms for which there is a **K**-atom that is evaluated to one of the truth values **b**, **st** and **t** in that p-model.

Furthermore, if we would not add condition (3), then e.g. for the rules

$$\begin{aligned} \mathbf{K}\,P(a) &\leftarrow \mathbf{not}\,P(a). \\ \mathbf{K}\,Q(a) &\leftarrow \mathbf{K}\,P(a). \end{aligned}$$

two models would be admitted: in both $\mathbf{K}\,P(a)$ would be mapped to **u**, but $\mathbf{K}\,Q(a)$ could be either *classically false* or *undefined*. Then only the minimization to the well-founded paraconsistent model (which will be introduced later) would eliminate the p-model where the truth value **cf** is assigned to $\mathbf{K}\,Q(a)$. This problem is also avoided by introducing the third condition in Definition 3.10, i.e. the condition ensures also that unjustified assignments to **cf** are avoided, and does not interfere with rules such as

$$\begin{aligned} \mathbf{K}\,Q(a) &\leftarrow \mathbf{not}\,P(a). \\ \mathbf{K}\,P(a) &\leftarrow \mathbf{not}\,Q(a). \end{aligned}$$

having three p-models as intended.

Next, we lift the property that models of a hybrid MKNF knowledge base coincide with the models of its ground instantiation from the two- and three-valued hybrid MKNF semantics to our six-valued approach. The proof reflects the proof provided by Motik and Rosati [MR06] for showing the corresponding property in case of the two-valued hybrid MKNF semantics.

Proposition 3.11 (Semantical equivalence of ground instances)**.** *Let $\mathcal{K}$ be a DL-safe hybrid MKNF knowledge base, $\mathcal{K}_G$ the ground instantiation of $\mathcal{K}$ and (M, N) a p-interpretation pair. Then (M, N) is a p-model of $\mathcal{K}$ if and only if it is a p-model of $\mathcal{K}_G$.*

Proof. Regarding the first direction, from left to right, assume that (M, N) is a p-model of

a hybrid MKNF knowledge base $\mathcal{K}$. For each ground non-DL-atom A containing a constant not occurring in $\mathcal{K}$ it holds that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{f}$. To see that the previous holds, suppose that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) \neq \mathbf{f}$ and consider a p-interpretation pair (M', N') obtained from (M, N) by adding, for every ground non-DL-atom A containing a constant that does not occur in $\mathcal{K}$, a p-interpretation to M and N respectively that evaluates B to $\mathbf{f}$ and otherwise, is identical to a p-interpretation already contained in M, resp. N. Obviously, it holds that $M \subseteq M'$ and $N \subseteq N'$ where at least one of the inclusions is proper. Consider now each rule $r \in \mathcal{P}$ and its ground instance r_G. If r_G contains only constants that occur in $\mathcal{K}$, we obtain that $(M', N') \models_p r_G$ because the truth values of ground non-DL-atoms containing only constants appearing in $\mathcal{K}$ coincide in M and M', respectively N and N'. Alternatively, r_G contains a constant that does not occur in $\mathcal{K}$ and, due to DL-safety of r_G, its body contains an atom $\mathbf{K}A$ containing a constant that does not occur in $\mathcal{K}$. But then $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) = \mathbf{f}$ by definition of (M', N') and therefore, $(M', N') \models_p r_G$ as the body of r_G is *false* in this case. Thus, $M' \models_p \mathcal{K}$, which contradicts the assumption that (M, N) is a p-model of $\mathcal{K}$ according to condition (2) of Definition 3.10. Consequently, $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{f}$ in case A contains a constant that does not appear in $\mathcal{K}$, from which it follows immediately that (M, N) is a p-model of $\mathcal{K}_G$.

For the other direction, from right to left, suppose that (M, N) is a p-model of $\mathcal{K}_G$. Clearly, we have that $(M, N) \models_p \mathcal{O}$. Thus, we have to prove that $(M, N) \models_p \mathcal{K}$, i.e. that $(M, N) \models_p r$ for each $r \in \mathcal{P}$. Consider a ground instance r_G of r. If r_G contains only constants that occur in $\mathcal{K}$, then $(M, N) \models_p r_G$ holds. Otherwise, since r is DL-safe, each constant that does not appear in $\mathcal{K}$ occurs in some ground non-DL-atom $\mathbf{K}\,A$ in the body of r_G. Assume that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) \neq \mathbf{f}$ and consider again a p-interpretation pair (M', N') obtained from (M, N) by adding, for every ground non-DL-atom A containing a constant that does not occur in $\mathcal{K}$, a p-interpretation to M and N respectively that evaluates B to $\mathbf{f}$ and otherwise is identical to a p-interpretation already contained in M, resp. N. Since $\mathcal{K}_G$ does not contain a constant that does not occur in $\mathcal{K}$, we know that $(M', N') \models_p \mathcal{K}_G$, which contradicts the assumption that (M, N) is a p-model of $\mathcal{K}_G$. Hence, $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{f}$, so that $(M, N) \models_p r_G$ and hence, $(M, N) \models_p \mathcal{K}$. Additionally, suppose that a p-interpretation pair (M'', N'') exists such that $M \subseteq M''$ and $N \subseteq N''$ where at least one of the inclusions is proper, and $(\mathcal{I}'', \langle M'', N''\rangle, \langle M, N\rangle)(\mathcal{K}) \in \{\mathbf{b}, \mathbf{t}\}$ for every $\mathcal{I}'' \in M'' \cup N''$. Clearly, it is the case that $(\mathcal{I}'', \langle M'', N''\rangle, \langle M, N\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for every $\mathcal{I}'' \in M'' \cup N''$, which contradicts the assumption that (M, N) is a p-model of $\mathcal{K}_G$. Hence, (M, N) is a p-model of $\mathcal{K}$. □

As the previous proposition shows that the semantics of hybrid MKNF knowledge bases and their ground instantiations coincide, we will only consider ground hybrid MKNF knowledge bases in the following.

We can now turn back to our running example and demonstrate in detail that the paraconsistent MKNF semantics just defined yields the expected results. The following

example represents a fragment of the ground version of the hybrid MKNF knowledge base presented in Example 3.1. We omit the predicate good here since DL-safety is not an issue in a ground knowledge base.

Example 3.12 (P-models of a ground hybrid MKNF knowledge base)**.** Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ containing one ontology axiom, four MKNF rules and one fact.

$$\mathsf{HasCertifiedForwarder} \sqsubseteq \neg\mathsf{IsMonitored}$$

$$\begin{aligned}
\mathbf{K}\,\mathsf{IsMonitored(food)} &\leftarrow \mathbf{K}\,\mathsf{risk(food)}.\\
\mathbf{K}\,\mathsf{risk(food)} &\leftarrow \mathbf{not}\,\mathsf{isLabelled(food)}.\\
\mathbf{K}\,\mathsf{isLabelled(food)} &\leftarrow \mathbf{not}\,\mathsf{risk(food)}.\\
\mathbf{K}\,\mathsf{resolvedRisk(food)} &\leftarrow \mathbf{K}\,\mathsf{IsMonitored(food)}.\\
\mathbf{K}\,\mathsf{HasCertifiedForwarder(food)} &\leftarrow
\end{aligned}$$

At first, we take only the program component $\mathcal{P}_G$ into account. A p-interpretation pair (M, N) which is a p-model of $(\emptyset, \mathcal{P}_G)$ has to p-satisfy $\mathcal{P}_G$ according to condition (1) of Definition 3.10, and hence has to p-satisfy each rule, according to the definition of the conjunction operator by means of the join in the lattice $\mathcal{SIX}$. So, the only fact in the program, $\mathbf{K}\,\mathsf{HasCertifiedForwarder(food)}$, has to be mapped to a designated truth value by (M, N). However, the designated truth value $\mathbf{b}$ is not allowed for $\mathbf{K}\,\mathsf{HasCertifiedForwarder(food)}$ in a p-model due to condition (3) of the definition of p-models since the classically negated atom $\neg\mathsf{HasCertifiedForwarder(food)}$ is not derivable (even when taking $\mathcal{O}$ into account). Hence, it has to be assigned the truth value $\mathbf{t}$ by (M, N).

In order to p-satisfy the second and the third rule, the modal **K**-atoms $\mathbf{K}\,\mathsf{risk(food)}$ and $\mathbf{K}\,\mathsf{isLabelled(food)}$ could both be evaluated with $\mathbf{t}$ in (M, N). In this case, the modal **not**-atoms $\mathbf{not}\,\mathsf{risk(food)}$ and $\mathbf{not}\,\mathsf{isLabelled(food)}$ would be evaluated to $\mathbf{f}$, according to the definition of the modal **not**-operator, such that both implications would be p-satisfied. However, after fixing the evaluation of the modal **not**-atoms, this interpretation would not be minimal w.r.t. the minimization order since $\mathbf{K}\,\mathsf{risk(food)}$ and $\mathbf{K}\,\mathsf{isLabelled(food)}$ could both be minimized to *false* in this case. Now, there are three different options for assigning a truth value to the **K**-atoms $\mathbf{K}\,\mathsf{risk(food)}$ and $\mathbf{K}\,\mathsf{isLabelled(food)}$ in (M, N) which fulfill all conditions (1)-(3) of Definition 3.10:

1. The truth value $\mathbf{t}$ can be assigned to $\mathbf{K}\,\mathsf{risk(food)}$, and $\mathbf{K}\,\mathsf{isLabelled(food)}$ can be mapped to *false* in (M, N). In this case, $\mathbf{not}\,\mathsf{risk(food)}$ is evaluated to *false* and $\mathbf{K}\,\mathsf{isLabelled(food)}$ is evaluated to *true*. This p-satisfies both rules according to the definition of the implication operator in Table 3.1 as the second rule has a *true* body and a *true* head, while the body and the head of the third rule are both evaluated to the truth value $\mathbf{f}$ then. Furthermore, after fixing the evaluation of

the modal **not**-atoms in these rules, the assignment is minimal. On the one hand **K** isLabelled(food) takes already the least value of the minimization order. On the other hand **K** risk(food) appears in the head of a rule with a *true* body such that it is not allowed to take one of the truth values **st**, **f**, **cf** and **u**. Moreover, **K** risk(food) cannot take the truth value **b** in (M, N) as condition (3) of Definition 3.10 prohibits this.

2. The **K**-atom **K** risk(food) can be mapped to **f**, and **K** isLabelled(food) can be evaluated to the truth value *true* in (M, N). Then, **not** risk(food) is evaluated to *false* and **K** isLabelled(food) is evaluated to *true*. This assignment p-satisfies the second and the third rule and results in a p-model of $(\emptyset, \mathcal{P}_G)$ for analogous reasons as for the first option.

3. Both modal **K**-atoms can be evaluated to *undefined* in (M, N). The second and the third rule are p-satisfied by (M, N) in this case since the corresponding **not**-atoms are mapped to *undefined* as well, resulting in both rules having an *undefined* body and head. After fixing the interpretation of the modal **not**-atoms, the truth value *undefined* is also the smallest value in the minimization order that can be assigned to **K** risk(food) and **K** isLabelled(food) since the value **f** is not allowed for the heads of rules with *undefined* body according to the definition of the implication operator in Table 3.1. However, without condition (3) of Definition 3.10, they could be evaluated to the truth value *classically false* in a p-model because **u** and **cf** are incomparable in the minimization order. Condition (3) prohibits this.

Furthermore, when only taking the program component into account, the truth value of the **K**-atom **K** IsMonitored(food) in the respective p-model is also determined by each of the three options for assigning a truth value to the **K**-atom **not** risk(food) just mentioned. If **not** risk(food) is *false*, **K** IsMonitored(food) also has to take the truth value *false* as it is the least element in the minimization order and since this is allowed by the definition of the implication operator. In case **not** risk(food) is *undefined*, the value *false* is no longer allowed for **K** IsMonitored(food) because of the definition of $\supset$, and the smallest values which are allowed are *undefined* and *classically false*. However, the truth value *classically false* cannot be assigned due to condition (3) of the definition of p-models and hence, **K** IsMonitored(food) has to be mapped to the truth value **u** in this case. Finally, if **not** risk(food) is *true*, the minimal value that can be taken by **K** IsMonitored(food) and which fulfills the definition of $\supset$ as well as condition (3) is **t**. By the same line of reasoning, the truth value of **K** resolvedRisk(food) w.r.t. to each of the three options is identical to the value assigned to **K** IsMonitored(food) in the respective p-model.

Now, taking also the ontology component into account, for the only axiom to be p-satisfied by a p-model (M, N) of $\mathcal{K}_G$, the consequent of the implication must be mapped to one of the truth values *inconsistent* and *true* by every p-interpretation in the sets M and N due to the definition of $\supset$ and because the implicant is also *inconsistent* or *true* in all

of these p-interpretations. Consequently, IsMonitored(food) must be evaluated to one of *inconsistent* and *false* in every p-interpretation in $M \cup N$, according to the definition of the negation operator. As a result, **K** IsMonitored(food) is mapped to either **b** or **cf** by (M, N).

Regarding the first option of assigning truth values to the **K**-atoms **K** risk(food) and **K** isLabelled(food), this means that **K** IsMonitored(food) has to be interpreted with the truth value **b** as it appears in the head of a rule with *true* body. Note that this is now allowed according to condition (3) since the ontology component allows us to derive ¬IsMonitored(food). Consequently, **K** resolvedRisk(food) now is in the head of a rule with *inconsistent* body and due to the definition of the implication operator can be minimized to **st**. This shows that Contradiction Support Detection works as expected in p-models.

Regarding the second option, **K** IsMonitored(food) can take the smaller value **cf** because the corresponding rule body is mapped to the truth value **f**. In this way, the user can detect that the classical negation of **K** IsMonitored(food) is entailed by the ontology. The **K**-atom **K** resolvedRisk(food) is still assigned the truth value *false* by (M, N) in this case.

Finally, in the case of the third option, the **K**-atom **K** IsMonitored(food) in the head of the first rule, which has an *undefined* body in this case, is evaluated to **cf** when the ontology component is taken into account as condition (3) ensures that among the two incomparable (w.r.t. minimization order) truth values **u** and **cf** the latter is chosen. This demonstrates the intervention of the Coherence Principle which "overwrites" the regular propagation of the truth value *undefined* in the program component.

In conclusion, we obtain three different p-models for $\mathcal{K}_G$ corresponding to the three options for assigning truth values to the **K**-atoms **K** risk(food) and **K** isLabelled(food) described above. In the first one, the **K**-atom **K** risk(food) is *true*, **K** isLabelled(food) is *false*, **K** IsMonitored(food) is *inconsistent* and **K** resolvedRisk(food) is *suspiciously true*. In the second p-model, **K** risk(food) and **K** resolvedRisk(food) are *false*, **K** isLabelled(food) is *true*, and **K** IsMonitored(food) is *classically false*. In the third p-model, **K** risk(food) and **K** isLabelled(food) are *undefined*, **K** IsMonitored(food) is *classically false* and the **K**-atom **K** resolvedRisk(food) is *false*. The program fact **K** HasCertifiedForwarder(food) is evaluated to *true* in all of the three p-models as already stated. Here, the third model is special in the sense that it is most "skeptical" about the *true* and *false* knowledge that is derivable from $\mathcal{K}_G$, and in fact it represents the unique *well-founded p-model* of $\mathcal{K}_G$, a notion which will be defined later. ◊

3.6 Propagation of Inconsistencies in the Ontology Component

After having defined under which conditions a p-interpretation pair models a ground hybrid MKNF knowledge base, some particularities of the semantics regarding the propagation of inconsistencies can now be discussed. We will demonstrate by means of two

examples why it is necessary that *inconsistent* interpretations of modal **K**-atoms are defined solely in terms of the first component M of a p-interpretation pair (M, N) in Definition 3.6.

While it is predetermined that inconsistencies arising within the ontology are not propagated (in order to establish the correspondence to $\mathcal{ALC}4$), and dependencies on contradictions have to be detectable from the semantics of the program (for the coincidence with $WFSX_p$), the border cases are not so clear. For instance, we could only assign the truth value **st** to modal **K**-atoms that can be derived from the program component, so that all knowledge which is p-entailed by the ontology is either *true* or *inconsistent*. On the other hand, we can also propagate inconsistencies via the ontology, so that if a **K**-atom can only be derived from the ontology with the help of an inconsistent **K**-atom that occurs in the program component, it is also *suspiciously true*. Here, we opt for the latter approach in order to move the propagation of inconsistencies as far into the ontology as possible. In this way, we are able to detect dependencies on contradictions in as many cases as possible without sacrificing faithfulness w.r.t. $\mathcal{ALC}4$. As the truth value **st** distinguishes those pieces of knowledge which depend on a contradiction from those which are contradictory themselves, there is no downside to this strategy and in general, the semantics provides more information in this way. After introducing a relation between the objective knowledge of a knowledge base w.r.t. certain sets of **K**-atoms and the p-models of the knowledge base in Proposition 3.37, we will be able to characterize the propagation of inconsistencies within the ontology component formally in Corollary 3.38. For now, we just demonstrate how the described behavior can be achieved by our particular definition of the semantics of modal **K**-formulas, by means of the following two examples.

The first example shows that if a **K**-atom has to be *inconsistent* because its classical negation can be derived from the ontology, but its positive version can only be derived from the program component, dependencies on this contradiction are propagated via the ontology.

Example 3.13. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$\begin{array}{rcl} \top & \sqsubseteq & \neg P \\ P & \sqsubseteq & R \\ \mathbf{K}\,P(a) & \leftarrow & \\ \mathbf{K}\,R(a) & \leftarrow & \mathbf{K}\,R(a) \end{array}$$

Due to the first ontology axiom, $P(a)$ cannot be evaluated to *true* by any p-interpretation in the sets M and N of a p-interpretation pair (M, N) that p-satisfies $\mathcal{K}_G$. Because of the fact in the program component, $P(a)$ can also not be *false* in any p-interpretation in M, so that it has to be **b** under all p-interpretations in M. Therefore, $\mathbf{K}\,P(a)$ has to be mapped to **b** by every p-model (M, N) of $\mathcal{K}_G$, and due to minimization by condition (2) of Definition

3.10, there is a p-interpretation in N which maps $P(a)$ to *false*.

Now, $R(a)$ has to be assigned one of the truth values $\mathbf{b}$ and $\mathbf{t}$ by all p-interpretation pairs in M due to the second axiom. At the same time, it can also be mapped to *false* by some p-interpretation pair in N because of some p-interpretation in N that maps $P(a)$ to *false*. Due to minimization, $\mathbf{K}\,R(a)$ is mapped to $\mathbf{st}$ by every p-model of $\mathcal{K}_G$. Consequently, it can be detected that $\mathbf{K}R(a)$ depends on a contradictory **K**-atom in the program component. $\Diamond$

However, when a contradictory first-order atom can be derived from the ontology without depending on contradictory knowledge from the program component, this inconsistency cannot be propagated within the ontology because otherwise faithfulness w.r.t. $\mathcal{ALC}4$ would be lost. We show that this is the case for our semantics by means of the following example.

Example 3.14. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$\begin{array}{rcl} \top & \sqsubseteq & \neg P \\ \top & \sqsubseteq & P \\ P & \sqsubseteq & R \\ \mathbf{K}\,R(a) & \leftarrow & \mathbf{K}\,R(a) \end{array}$$

In contrast to the previous example, $P(a)$ cannot be evaluated to either *true* or *false* by any p-interpretation in the sets M and N of any p-interpretation pair (M, N) that p-satisfies $\mathcal{K}_G$. Consequently, $R(a)$ can be mapped to one of the truth values $\mathbf{b}$ and $\mathbf{t}$ by all p-interpretations in M and N because of the second axiom. Due to minimization, it will be mapped to $\mathbf{t}$ by some p-interpretation in M. As expected, $\mathbf{K}\,R(a)$ is evaluated to *true* by every p-model of $\mathcal{K}_G$ and therefore, the inconsistency is not propagated within the ontology. $\Diamond$

In Definition 3.8, we have already defined the paraconsistent entailment relation for first-order formulas. After having discussed the previous two examples, we can now also justify the following definition of paraconsistent entailment of an MKNF formula by a hybrid MKNF knowledge base on the basis of p-models.

Definition 3.15 (Paraconsistent MKNF entailment)**.** If $\mathcal{K}_G$ is a ground hybrid MKNF knowledge base and ψ a closed MKNF formula, and it holds that $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\psi) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$ for each $\mathcal{I} \in M \cup N$ holds for all p-models (M, N) of $\mathcal{K}_G$, then $\mathcal{K}_G$ *p-MKNF-entails* ψ, written $\mathcal{K}_G \models_p^{MKNF} \psi$.

Although $\mathbf{st}$ is not a designated truth value w.r.t. the condition for p-satisfaction in Definition 3.7, a closed MKNF formula ψ that is not evaluated to any of the truth values $\mathbf{f}$, $\mathbf{cf}$ and $\mathbf{u}$ by any of the p-models of a ground hybrid MKNF knowledge base $\mathcal{K}_G$ should still be p-entailed by $\mathcal{K}_G$. Otherwise, in Example 3.13, $\mathbf{K}\,R(a)$ would not be p-MKNF-entailed by the knowledge base. However, after adding the concept assertion $P(a)$ to the

ABox of the ontology component, $R(a)$ would be p-MKNF-entailed by $\mathcal{K}_G$. Due to the way we define p-MKNF-entailment for closed MKNF formulas in Definition 3.15, $R(a)$ is p-MKNF-entailed by $\mathcal{K}_G$ in both cases.

3.7 Faithfulness w.r.t. the Three-Valued MKNF Semantics

Next, we establish the first faithfulness result, i.e. we show the correspondence between the three-valued and the paraconsistent MKNF semantics in the case of consistent hybrid MKNF knowledge bases. For this purpose, we define the construction of a *paraconsistent counterpart* from a given three-valued MKNF interpretatin pair (as defined in [KAH11]).

Definition 3.16 (P-counterpart and p-extension)**.** Let (M, N) be a three-valued MKNF interpretation pair. Then, the p-interpretation pair $(M', N') = (\{\langle I, I_1\rangle \mid I \in M \text{ and } I_1 \subseteq I\}, \{\langle I, I_1\rangle \mid I \in N \text{ and } I_1 \subseteq I\})$ is the *paraconsistent counterpart* (or *p-counterpart*) of (M, N). Given a first-order interpretation I, the set $\{\langle I, I_1\rangle \mid I_1 \subseteq I\}$ is the *paraconsistent extension* (or *p-extension*) of I.

Our goal is to show that under certain conditions the p-counterpart of an MKNF interpretation pair is a p-model of a given hybrid MKNF knowledge base if and only if the corresponding three-valued MKNF interpretation pair is a three-valued MKNF model of that knowledge base. In the subsequent lemma we first show the relation of the interpretations of modal atoms in a three-valued MKNF interpretation pair and its p-counterpart.

Lemma 3.17. *Let (M, N) be an MKNF interpretation pair, I an arbitrary interpretation in M or N, $(M', N') = (\{\langle I, I_1\rangle \mid I \in M \text{ and } I_1 \subseteq I\}, \{\langle I, I_1\rangle \mid I \in N \text{ and } I_1 \subseteq I\})$ the p-counterpart of (M, N), $\mathbf{K}\,A$ a modal $\mathbf{K}$-atom and $\mathbf{not}\,B$ a modal $\mathbf{not}$-atom. Then*

- *it is not possible that one of $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) \in \{\mathbf{b}, \mathbf{st}\}$ and $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) \in \{\mathbf{b}, \mathbf{st}\}$ holds,*
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{t}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) = \mathbf{t}$,
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{f}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) \in \{\mathbf{f}, \mathbf{cf}\}$,
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,A) = \mathbf{u}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) = \mathbf{u}$,
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{not}\,B) = \mathbf{t}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) = \mathbf{t}$,
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{not}\,B) = \mathbf{f}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) = \mathbf{f}$ *and*
- $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{not}\,B) = \mathbf{u}$ *iff* $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) = \mathbf{u}$.

Proof. First, note that it is neither possible that $(\bigcap_{\mathcal{J} \in M'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{b}$ nor that $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{b}$ holds due to the construction of M' and N'. Consequently, also $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) = \mathbf{b}$ and $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) = \mathbf{b}$ are not possible, according to Definition 3.6. Additionally, recall that in the case of the

three-valued semantics it holds that $N \subseteq M$. Thus, we know that $N' \subseteq M'$ also holds and we derive that $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\, A) = \mathbf{st}$ and $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\, B) = \mathbf{st}$ are not possible either. This proves the first item of this lemma.

Regarding the second item, we have that $(J, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\, A) = \mathbf{t}$ holds iff $(J, \langle M, N\rangle, \langle M, N\rangle)(A) = \mathbf{t}$ for each $J \in M$ iff $(\bigcap_{\mathcal{J} \in M'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{t}$ and $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{t}$ iff $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\, A) = \mathbf{t}$, according to Definition 3.6 and since $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{b}$ is not possible. In the third case, we derive that $(J, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\, A) = \mathbf{f}$ iff $(J, \langle M, N\rangle, \langle M, N\rangle)(A) = \mathbf{f}$ for some $J \in N$ holds iff it is the case that $(\bigcap_{\mathcal{J} \in M'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{f}$ and $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{f}$ iff $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\, A) \in \{\mathbf{f}, \mathbf{cf}\}$, according to Definition 3.6, and since $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{b}$ is not possible and $N' \subseteq M'$. Finally, considering the fourth item of this lemma, we derive that $(J, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\, A) = \mathbf{u}$ holds iff $(J, \langle M, N\rangle, \langle M, N\rangle)(A) = \mathbf{f}$ for some $J \in M$ and $(J, \langle M, N\rangle, \langle M, N\rangle)(A) = \mathbf{t}$ for each $J \in N$ iff $(\bigcap_{\mathcal{J} \in M'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{f}$ and $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{t}$ iff $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\, A) = \mathbf{u}$, by Definition 3.6 and since $(\bigcap_{\mathcal{J} \in N'} \mathcal{J}, \langle M', N'\rangle, \langle M', N'\rangle)(A) = \mathbf{b}$ is not possible. The cases for the modal **not**-atoms can be proven similarly. □

It is obvious from the previous lemma that modal atoms are not interpreted paraconsistently in the p-counterpart (M', N') of some MKNF interpretation pair (M, N) since the truth values $\mathbf{b}$ and $\mathbf{st}$ do not appear in the interpretations of the modal **K**- and **not**-atoms. The following lemma demonstrates that for every first-order formula and every first-order interpretation I, there is a particular p-interpretation contained in the p-extension of I that evaluates φ to the same truth value as in I. Since first-order interpretations are two-valued, it cannot be the case that any first-order formula is evaluated to $\mathbf{b}$ by all p-interpretations in the p-extension of I.

Lemma 3.18. *Let φ be a closed MKNF formula not containing any modal operators and I an arbitrary first-order interpretation. Then $(I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t}$, and $(I, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{f}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{f}$.*

Proof. We prove both claims simultaneously by a structural induction on the structure of φ. For the base case, assume that φ is a first-order atom of the form $P(t_1, \ldots, t_n)$. First, we derive that $(I, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{t}$ iff $(t_1^I, \ldots, t_n^I) \in P^I$ iff it holds that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{t}$. Additionally, we derive that $(I, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{f}$ iff $(t_1^I, \ldots, t_n^I) \notin P^I$ iff it holds that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{f}$.

Now, we show the induction step for the classical negation operator $\neg$. So, assume that the claim holds for some first-order formula φ' and let $\varphi = \neg\varphi'$. In this case, we obtain that $(I, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{t}$ iff $(I, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{f}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{f}$ (by the induction hypothesis) iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{t}$. Moreover, we obtain that $(I, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{f}$ iff $(I, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{t}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{t}$ (by the induction hypothesis) iff it is the case that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{f}$. The induction steps for the other classical operators proceed similarly. □

The following lemma extends the result of Lemma 3.18 by showing that if there is a p-interpretation in the p-extension of some first-order interpretation I which evaluates a first-order formula φ to $\mathbf{t}$, then there cannot be a p-interpretation in that p-extension which evaluates φ to $\mathbf{f}$ (and the other way around). Accordingly, Lemma 3.18 and Lemma 3.19 together reveal the two-valued nature of p-extensions.

Lemma 3.19. *Let φ be a closed MKNF formula not containing any modal operators and I an arbitrary first-order interpretation. Then $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{t}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$, and $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi) = \mathbf{f}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi) \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$.*

Proof. We prove both claims simultaneously by a structural induction on the structure of φ. For the base case, assume that φ is a first-order atom of the form $P(t_1, \ldots, t_n)$. First, we derive that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{t}$ iff it is the case that $(t_1^I, \ldots, t_n^I) \in P^I$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$. Additionally, we derive that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{f}$ iff $(t_1^I, \ldots, t_n^I) \notin P^I$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{f}$ for every $I_1 \subseteq I$ iff it holds that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$. (Note that the latter bi-conditional is vacuously true since $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) = \mathbf{b}$ is actually not possible for any $I_1 \subseteq I$ in case $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(P(t_1, \ldots, t_n)) \neq \mathbf{t}$ for any $I_1 \subseteq I$.)

Next, we show the induction step for the classical negation operator $\neg$. So, assume that the claim holds for some first-order formula φ' and let $\varphi = \neg\varphi'$. In this case, we obtain that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{t}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{f}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi') \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$ (by the induction hypothesis) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$. Moreover, we obtain that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') = \mathbf{f}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi') = \mathbf{t}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi') \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$ (by the induction hypothesis) iff it is the case that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\neg\varphi') \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$.

Now, we perform the induction step for the conjunction operator $\wedge$. So, assume that the claim holds for two arbitrary first-order formulas φ_1 and φ_2, and let $\varphi = \varphi_1 \wedge \varphi_2$. We derive that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_1 \wedge \varphi_2) = \mathbf{t}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_1) = \mathbf{t}$ and $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_2) = \mathbf{t}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_1) \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$ and $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_2) \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$ (by the induction hypothesis) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_1 \wedge \varphi_2) \in \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I$. For the second claim, we derive that $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_1 \wedge \varphi_2) = \mathbf{f}$ iff $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_1) = \mathbf{f}$ or $(\langle I, \emptyset\rangle, \mathcal{M}, \mathcal{N})(\varphi_2) = \mathbf{f}$ iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_1) \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$ or $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_2) \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$ (by the induction hypothesis) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\varphi_1 \wedge \varphi_2) \in \{\mathbf{b}, \mathbf{f}\}$ for every $I_1 \subseteq I$. The induction steps for the other classical operators can be proven similarly. □

By utilizing the previously proven lemmas, we can now show that under certain conditions an MKNF interpretation pair satisfies a hybrid MKNF knowledge base if and only if its p-counterpart p-satisfies the knowledge base. This fact is a direct consequence of the following proposition, which in turn is crucial for the proof of the first faithfulness theorem.

Proposition 3.20 (Correspondence w.r.t. satisfaction)**.** *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a consistent ground hybrid MKNF knowledge base, (M, N) an MKNF interpretation pair, (M', N') its p-counterpart where there is no rule in $\mathcal{P}_G$ for which the body is undefined and the head is classically false in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$, and $\mathcal{I}' = \langle I, I_1\rangle$ a p-interpretation such that $I \in M \cup N$ and $I_1 \subseteq I$. Then it holds that $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathcal{K}_G) = \mathbf{t}$ iff $(\mathcal{I}', \langle M', N'\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for every $\mathcal{I}'$.*

Proof. We know by Definition 2.15 that $\mathcal{K}_G = \pi(\mathcal{O}) \wedge \pi(\mathcal{P}_G)$. Consequently, we can prove this proposition by proving that $(I, \langle M, N\rangle, \langle M, N\rangle)(\pi(\mathcal{O})) = \mathbf{t}$ holds if and only if it is the case that $(\mathcal{I}', \langle M', N'\rangle, \langle M', N'\rangle)(\pi(\mathcal{O})) \in \{\mathbf{b}, \mathbf{t}\}$ for every $\mathcal{I}'$ as well as it is the case that $(I, \langle M, N\rangle, \langle M, N\rangle)(\pi(\mathcal{P}_G)) = \mathbf{t}$ if and only if $(\mathcal{I}', \langle M', N'\rangle, \langle M', N'\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$ holds for every $\mathcal{I}'$. Since the ontology component is represented by a first-order formula not containing any modal operators, the claim w.r.t. $\pi(\mathcal{O})$ follows directly from Lemma 3.18 and Lemma 3.19.

So, we only need to prove that the claim also holds for the program component $\mathcal{P}_G$, i.e. that it is the case that $(I, \langle M, N\rangle, \langle M, N\rangle)(\pi(\mathcal{P}_G)) = \mathbf{t}$ if and only if it holds that $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$. Due to Definition 2.15, we know that $\pi(\mathcal{P}_G)$ consists of a conjunction of implications of the form $\mathbf{K}\,H \subset \mathbf{K}\,A_1 \wedge \ldots \wedge \mathbf{K}\,A_n \wedge \mathbf{not}\,B_1 \wedge \ldots \wedge \mathbf{not}\,B_m$. First, we prove the claim for a single implication i, i.e. we show that $(I, \langle M, N\rangle, \langle M, N\rangle)(i) = \mathbf{t}$ if and only if $(*, \langle M', N'\rangle, \langle M', N'\rangle)(i) \in \{\mathbf{b}, \mathbf{t}\}$. An implication i in $\pi(\mathcal{P}_G)$ is *true* in $(I, \langle M, N\rangle, \langle M, N\rangle)$ iff its consequent is *true*; its implicant is *false*; or both are *undefined*, regarding Definition 7 in [KAH11]. The previous holds iff $\mathbf{K}\,H$ is *true* in $(I, \langle M, N\rangle, \langle M, N\rangle)$; at least one $\mathbf{K}\,A_i$ or $\mathbf{not}\,B_j$ is *false* in $(I, \langle M, N\rangle, \langle M, N\rangle)$; or $\mathbf{K}\,H$ is *undefined*, all $\mathbf{K}\,A_i$ and $\mathbf{not}\,B_j$ are *true* or *undefined* and at least one $\mathbf{K}\,A_i$ or $\mathbf{not}\,B_j$ is *undefined* in $(I, \langle M, N\rangle, \langle M, N\rangle)$. According to Lemma 3.17, this is the case iff $\mathbf{K}\,H$ is evaluated to $\mathbf{t}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$; some $\mathbf{K}\,A_i$ is evaluated to one of $\mathbf{f}$ and $\mathbf{cf}$ or some $\mathbf{not}\,B_j$ is evaluated to $\mathbf{f}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$; or $\mathbf{K}\,H$ is evaluated to $\mathbf{u}$, all $\mathbf{K}\,A_i$ and $\mathbf{not}\,B_j$ are evaluated to one of $\mathbf{t}$ and $\mathbf{u}$, and at least one $\mathbf{K}\,A_i$ or $\mathbf{not}\,B_j$ is evaluated to $\mathbf{u}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$. Finally, the aforementioned holds iff the consequent of i is evaluated to $\mathbf{t}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$; the implicant of i is evaluated to one of $\mathbf{f}$ and $\mathbf{cf}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$; or both the implicant and the consequent of i are evaluated to $\mathbf{u}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$. According to Table 3.1, i is mapped to $\mathbf{t}$ by $(*, \langle M', N'\rangle, \langle M', N'\rangle)$ in all of these cases. This proves that if $(I, \langle M, N\rangle, \langle M, N\rangle)(i) = \mathbf{t}$, then $(*, \langle M', N'\rangle, \langle M', N'\rangle)(i) \in \{\mathbf{b}, \mathbf{t}\}$.

In order to prove the other direction, we just have to show that these are all possible cases where $(*, \langle M', N'\rangle, \langle M', N'\rangle)(i) \in \{\mathbf{b}, \mathbf{t}\}$. According to Lemma 3.17, there cannot be a modal $\mathbf{K}$-atom $\mathbf{K}\,A$ in the implicant or the consequent of i such that it holds that $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,A) \in \{\mathbf{b}, \mathbf{st}\}$, and also there cannot be a modal $\mathbf{not}$-atom in the implicant of i such that $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{not}\,B) \in \{\mathbf{b}, \mathbf{st}\}$ holds. As a result, neither the implicant of i nor the consequent of i can be evaluated to one of $\mathbf{b}$ and $\mathbf{st}$ in (M', N'). The only remaining case in which an implication is mapped to one of $\mathbf{b}$ and $\mathbf{t}$, regarding

Table 3.1, occurs when the implicant is evaluated to $\mathbf{u}$ and the consequent is evaluated to $\mathbf{cf}$. However, this case is already excluded by the definition of (M', N'). Hence, both directions of the claim hold for a single implication in $\pi(\mathcal{P}_G)$. Since $\pi(\mathcal{P}_G)$ constitutes a conjunction of implications, the claim w.r.t. the program component follows directly by the definition of the conjunction in Definition 7 of [KAH11] and Definition 3.6. The same holds for the conjunction of $\pi(\mathcal{O})$ and $\pi(\mathcal{P}_G)$ and hence, this suffices to prove the proposition for $\mathcal{K}_G$. □

As the program component may be empty, the previous proposition obviously holds for first-order formulas as well. As a result, we obtain the following corollary, where the operator $\models$ denotes the classical entailment relation of first-order logic.

Corollary 3.21. *Let φ and ψ be two consistent closed first-order formulas. Then $\varphi \models \psi$ if and only if $\varphi \models_p \psi$.*

Proof. This is a direct consequence of Proposition 3.20 and the definition of the paraconsistent entailment relation in Definition 3.8. □

Before we present the main faithfulness result w.r.t. the three-valued MKNF semantics, we demonstrate that even a consistent hybrid MKNF knowledge base can have p-models which do not correspond to any three-valued MKNF model. This fact motivates the introduction of the additional condition on the p-interpretation pair (M', N') in Proposition 3.20. By restricting the p-interpretations in Proposition 3.20 to those in which no rule has an *undefined* body and a *classically false* head, we obtain the one-to-one correspondence between three-valued MKNF models and p-models of hybrid MKNF knowledge bases.

Example 3.22. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$\begin{array}{rcl} P & \sqsubseteq & \neg S \\ \mathbf{K}\,P(a) & \leftarrow & \mathbf{not}\,Q(a) \\ \mathbf{K}\,Q(a) & \leftarrow & \mathbf{not}\,P(a) \\ \mathbf{K}\,R(a) & \leftarrow & \mathbf{not}\,R(a) \\ \mathbf{K}\,S(a) & \leftarrow & \mathbf{K}\,R(a) \end{array}$$

When only considering the program component, $\mathbf{K}\,R(a)$ has to be *undefined* in every three-valued MKNF model (M, N) as well as in every p-model (M', N') of $\mathcal{K}_G$. For the first two rules alone, there are three different models under both the three-valued and the paraconsistent MKNF semantics. In particular, $\mathbf{K}\,P(a)$ takes each of the truth values $\mathbf{t}$, $\mathbf{u}$ and $\mathbf{f}$ in some model. Moreover, since $\mathbf{K}\,R(a)$ is *undefined* in every model of the program component, $\mathbf{K}\,S(a)$ is also evaluated to $\mathbf{u}$. Taking the ontology axiom into account, $S(a)$ has to be *false* or *inconsistent* in every p-interpretation in M', as well as *false* in every interpretation in N. Hence, $\mathbf{K}S(a)$ cannot be *undefined* in the respective model under each

of the two semantics anymore. Now, in the case of the paraconsistent MKNF semantics, $\mathbf{K}\,S(a)$ can simply be evaluated to $\mathbf{cf}$, resulting in a p-model. However, the mentioned p-model would be the p-counterpart of an MKNF interpretation pair that evaluates the head of the last rule to $\mathbf{f}$ and its body to $\mathbf{u}$. This assignment does not yield a three-valued MKNF model. As a consequence, $\mathcal{K}_G$ only has two models under the three-valued semantics such that there is one p-model that does not correspond to any three-valued model, even though the knowledge base is consistent (in the sense that it has at least one three-valued MKNF model). $\Diamond$

For the proof of the following theorem, we adapt the proof of Proposition 1 in [KAH11] to our approach. Here, we may assume that $\pi(\mathcal{K}_G)$ still equals $\pi(\mathcal{O}) \wedge \pi(\mathcal{P}_G)$ when considering three-valued MKNF models of $\mathcal{K}_G$ and that the satisfaction relation in this case is defined w.r.t. each I in $M \cup N$, like in the case of p-satisfaction as defined in Definition 3.7. The previous is possible because the resulting three-valued semantics coincides with the original semantics from [KAH11] when only considering ground hybrid MKNF knowledge bases - as a result of the definition of the modal $\mathbf{K}$-operator in Definition 7 of [KAH11] and due to the fact that $N \subseteq M$ in the definition of the satisfaction relation as formulated in Definition 8 of [KAH11].

Theorem 3.23 (Faithfulness w.r.t. the three-valued MKNF semantics)**.** *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a consistent ground hybrid MKNF knowledge base, (M, N) an MKNF interpretation pair and (M', N') its p-counterpart where there is no rule in $\mathcal{P}_G$ for which the body is evaluated to $\mathbf{u}$ and the head is evaluated to $\mathbf{cf}$ in $(*, \langle M', N'\rangle, \langle M', N'\rangle)$. Then (M, N) is a three-valued MKNF model of $\mathcal{K}_G$ if and only if (M', N') is a p-model of $\mathcal{K}_G$.*

Proof. Assume that (M', N') is a p-model of $\mathcal{K}_G$, i.e. that (M', N') fulfills the three conditions of Definition 3.10. We prove that (M, N) is a three-valued MKNF model of $\mathcal{K}_G$. It follows from the first condition of Definition 3.10 that $(\mathcal{I}', \langle M', N'\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$ and thus, by Proposition 3.20, that $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathcal{K}_G) = \mathbf{t}$ for each $I \in M \cup N$. The second condition states that for each p-interpretation pair (M'', N'') with $M' \subseteq M''$ and $N' \subseteq N''$ where at least one of the inclusions is proper, we have $(\langle I'', I_1\rangle, \langle M'', N''\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \notin \{\mathbf{b}, \mathbf{t}\}$ for some $\langle I'', I_1\rangle \in M'' \cup N''$. Due to Proposition 3.20, we conclude that $(I'', \langle M'', N''\rangle, \langle M, N\rangle)(\mathcal{K}_G) \neq \mathbf{t}$. Hence, we obtain that also for any MKNF interpretation pair (M'', N'') with $M \subseteq M''$ and $N \subseteq N''$ where at least one of the inclusions is proper it holds that $(I'', \langle M'', N''\rangle, \langle M, N\rangle)(\mathcal{K}_G) \neq \mathbf{t}$ for some $I'' \in M'' \cup N''$. Consequently, (M, N) fulfills the two conditions of Definition 9 in [KAH11] and therefore, is a three-valued MKNF model of $\mathcal{K}_G$.

Now, assume that (M, N) is a three-valued MKNF model of $\mathcal{K}_G$. We prove that (M', N') is a p-model of $\mathcal{K}_G$. We know that $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathcal{K}_G) = \mathbf{t}$ for each $I \in M \cup N$ by condition (1) of Definition 9 in [KAH11]. Consequently, we obtain that it holds that $(\mathcal{I}', \langle M', N'\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$ according to Proposition 3.20, so that the first condition of Definition 3.10 is fulfilled. Moreover, as (M, N) is a three-valued MKNF model of $\mathcal{K}_G$, we know that for all (M'', N'') with $M \subseteq M''$ and $N \subseteq N''$

where at least one of the inclusions is proper, we have that $(I'', \langle M'', N''\rangle, \langle M, N\rangle)(\mathcal{K}_G) \neq \mathbf{t}$ for some $I'' \in M'' \cup N''$. We obtain that $(\langle I'', I_1\rangle, \langle M'', N''\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \notin \{\mathbf{b}, \mathbf{t}\}$ for every $I_1 \subseteq I''$, again by Proposition 3.20. Hence, we infer that for any p-interpretation pair (M'', N'') with $M' \subset M''$ and $N' \subset N''$ where at least one of the inclusions is proper we have that $(\mathcal{I}'', \langle M'', N''\rangle, \langle M', N'\rangle)(\mathcal{K}_G) \notin \{\mathbf{b}, \mathbf{t}\}$ for some $\mathcal{I}'' \in M'' \cup N''$.

Regarding condition (3) of Definition 3.10, we just have to prove that for every $\mathbf{K}\,\xi \in \mathsf{KA}(\mathcal{K}_G)$ it holds that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi) = \mathbf{cf}$ if and only if $\{\pi(\mathcal{O})\} \cup \{\xi' \mid \mathbf{K}\,\xi' \in \mathsf{KA}(\mathcal{K}_G)$ and $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi') = \mathbf{t}\} \models_p \neg\xi$ because of the first item of Lemma 3.17. Let $\mathbf{K}\,\xi$ be an arbitrary $\mathbf{K}$-atom in $\mathsf{KA}(\mathcal{K}_G)$. Now, due to Definition 3.6, it holds that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi) = \mathbf{cf}$ iff $(I, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \mathbf{f}$ for all $I \in M$ because otherwise there would be a p-interpretation in M' in which ξ is evaluated to $\mathbf{t}$ according to the definition of (M', N'). This in turn holds iff $\{\pi(\mathcal{O})\} \cup \{\xi' \mid \mathbf{K}\,\xi' \in \mathsf{KA}(\mathcal{K}_G)$ and $(I, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi') = \mathbf{t}$ for each $I \in M \cup N\} \models \neg\xi$ since otherwise there would also be an interpretation in M that evaluates ξ to $\mathbf{t}$ according to Proposition 3 in [KAH11]. By means of Corollary 3.21 and Lemma 3.17, we derive that $\{\pi(\mathcal{O})\} \cup \{\xi' \mid \mathbf{K}\xi' \in \mathsf{KA}(\mathcal{K}_G)$ and $(*, \langle M', N'\rangle, \langle M', N'\rangle)(\mathbf{K}\,\xi') = \mathbf{t}\} \models_p \neg\xi$. This proves that (M', N') also fulfills condition (3) of Definition 3.10 and hence, that it is a p-model of $\mathcal{K}_G$. □

The following corollary to Theorem 3.23 lifts the faithfulness of total three-valued MKNF models w.r.t. two-valued MKNF models to our paraconsistent semantics.

Corollary 3.24. *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a consistent ground hybrid MKNF knowledge base, M an MKNF interpretation and $(M', M') = (\{\langle I, I_1\rangle \mid I \in M$ and $I_1 \subseteq I\}, \{\langle I, I_1\rangle \mid I \in M$ and $I_1 \subseteq I\})$ a p-interpretation pair such that there is no rule in $\mathcal{P}_G$ for which the body is evaluated to $\mathbf{u}$ and the head is evaluated to $\mathbf{cf}$ in $(*, \langle M', M'\rangle, \langle M', M'\rangle)$. Then M is a two-valued MKNF model of $\mathcal{K}_G$ if and only if (M', M') is a p-model of $\mathcal{K}_G$.*

Proof. This corollary is a direct consequence of Proposition 1 in [KAH11] and Theorem 3.23. □

3.8 Faithfulness of the Ontology Semantics w.r.t. $\mathcal{ALC}4$

Next, we consider the paraconsistent semantics assigned to a hybrid MKNF knowledge base $\mathcal{K}$ with $\mathcal{K} = (\mathcal{O}, \emptyset)$, i.e. a hybrid knowledge base where the program component is empty such that our semantics effectively boils down to a paraconsistent ontology semantics. As stated in Section 2.2, for the sake of a concise presentation, we only treat ontologies expressed in the syntax of the expressive and widely used DL $\mathcal{ALC}$ here. Nevertheless, our framework can be extended to more expressive DLs such as $\mathcal{SROIQ}4$ as well. The results established in this section are twofold. Firstly, we show that every p-interpretation $\langle I, I_1\rangle$ in the sets M and N of a p-model (M, N) of a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \emptyset)$[3] corresponds to a 4-model of $\mathcal{O} \cup LEM(\mathcal{O})$ and thus, that our

[3]We assume that $\mathcal{O}$ is expressed in the syntax of $\mathcal{ALC}$.

approach is faithful w.r.t. the semantics published in [MMH13]. Secondly, we obtain that the consequences yielded by the p-entailment relation $\models_p$ w.r.t. $\pi(\mathcal{O})$, which has been defined in Definition 3.8 and will be used in the definition of the operator $D_{\mathcal{K}_G}$ in Definition 4.1, correspond to the consequences of the entailment relation $\models_{\mathcal{ALC}4}$ w.r.t. $\mathcal{O} \cup LEM(\mathcal{O})$ presented in Section 2.2.

In Section 3.7, the p-counterpart of a three-valued MKNF interpretation pair has been defined in order to derive the first faithfulness result. To show faithfulness in case of the semantics of the ontology component, we adopt the same strategy and define the p-counterpart of a 4-interpretation in the following definition. The notion of p-counterpart is overloaded in this case, but it is clear from the context what it refers to. Furthermore, just as in the case of the two first-order interpretations I and I_1 forming a p-interpretation $\mathcal{I} = \langle I, I_1 \rangle$, we assume that 4-interpretations here also adhere to the Standard Name Assumption introduced in Section 2.4.2. This is a reasonable assumption since Maier et al. describe an embedding of $\mathcal{ALC}4$ into $\mathcal{ALC}$ [MMH13] and as mentioned already, the notions of satisfiability w.r.t. interpretations complying with the Standard Name Assumption and satisfiability w.r.t. unconstrained interpretations are equivalent in the case of $\mathcal{ALC}$.

Definition 3.25 (P-counterpart of a 4-interpretation)**.** Let $\mathcal{I}$ be a 4-interpretation over the signature (N_I, N_C, N_R). Then, the p-interpretation $\langle I, I_1 \rangle$, where $I_1 \subseteq I$, is the *p-counterpart* of $\mathcal{I}$ if and only if the following conditions are fulfilled:

- For all $a \in N_I$ and all $A \in N_C$ it holds that:
  - $a^I \notin A^I$ if and only if $a^{\mathcal{I}} \notin p^+(A^{\mathcal{I}})$ and $a^{\mathcal{I}} \in p^-(A^{\mathcal{I}})$,
  - $a^I \in A^I$ if and only if $a^{\mathcal{I}} \in p^+(A^{\mathcal{I}})$, and
  - $a^{I_1} \in A^{I_1}$ if and only if $a^{\mathcal{I}} \in p^+(A^{\mathcal{I}})$ and $a^{\mathcal{I}} \in p^-(A^{\mathcal{I}})$.
- For all $a_1, a_2 \in N_I$ and all $r \in N_R$ it holds that:
  - $(a_1^I, a_2^I) \in r^I$ if and only if $(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in p^+(r^{\mathcal{I}})$, and
  - $(a_1^{I_1}, a_2^{I_1}) \in r^{I_1}$ if and only if $(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in p^+(r^{\mathcal{I}})$ and $(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in p^-(r^{\mathcal{I}})$.

Due to Definition 3.25, obviously not every 4-interpretation has a p-counterpart. For instance, if an atomic concept assertion $A(a)$ is evaluated in a 4-interpretation such that $a^{\mathcal{I}} \notin p^+A^{\mathcal{I}}$ and $a^{\mathcal{I}} \notin p^-A^{\mathcal{I}}$, then there cannot exist a p-counterpart of $\mathcal{I}$ since neither $a^I \notin A^I$ nor $a^I \in A^I$ is allowed in this case due to Definition 3.25. However, provided a 4-interpretation $\mathcal{I}$ over a signature (N_I, N_C, N_R) 4-models the set of axioms $LEM(\mathcal{O})$ w.r.t. an ontology $\mathcal{O}$ over the same signature, it is ensured that $\mathcal{I}$ has a p-counterpart because then an atomic concept assertion cannot be evaluated to *undefined* in $\mathcal{I}$.

In addition, note that according to Definition 3.25, every p-interpretation is the p-counterpart of one or more 4-interpretations. Hence, the mapping of 4-interpretations satisfying $LEM(\mathcal{O})$ to their p-counterpart is surjective, but not injective. The reason

for the latter fact is that role assertions can still be *undefined* in a 4-interpretation that satisfies the set of axioms $LEM(\mathcal{O})$ and a similar set of axioms regarding roles cannot be formulated in the syntax of $\mathcal{ALC}$ (and even not in the one of $\mathcal{SROIQ}$[4]) according to Maier et al. On the basis of the previous definition we can formulate and prove the main faithfulness result w.r.t. the ontology component, from which the two results mentioned above follow directly as corollaries. In order to prove the faithfulness theorem, we recall the following proposition established in the paper by Maier et al. [MMH13].

Proposition 3.26 (Characterization of $LEM(\mathcal{O})$ [MMH13])**.** *Let $\mathcal{O}$ be an ontology. A 4-interpretation $\mathcal{I}$ is a 4-model of $LEM(\mathcal{O})$ if and only if for each concept C of $\mathcal{O}$ it holds that $p^+(C^\mathcal{I}) \cup p^-(C^\mathcal{I}) = \Delta^\mathcal{I}$.*

Proof. The original proof can be found in the paper by Maier et al. [MMH13]. □

The following proposition shows that the truth assignments of 4-interpretations and their p-counterpart correspond w.r.t. concept and role assertions.

Proposition 3.27 (Correspondence w.r.t. concept and role assertions)**.** *Let $\mathcal{O}$ be an ontology expressed in the syntax of the DL $\mathcal{ALC}$, $C(a)$ a complex concept assertion and $r(a_1, a_2)$ a role assertion in $\mathcal{O}$, $\mathcal{I}$ a 4-interpretation such that it holds for each concept C of $\mathcal{O}$ that $p^+(C^\mathcal{I}) \cup p^-(C^\mathcal{I}) = \Delta^\mathcal{I}$ and $\langle I, I_1\rangle$ the p-counterpart of $\mathcal{I}$. Then it holds that*

- *$a^\mathcal{I} \in p^+(C^\mathcal{I})$ if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(a))) \in \{\mathbf{b}, \mathbf{t}\}$,*
- *$a^\mathcal{I} \in p^-(C^\mathcal{I})$ if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(a))) \in \{\mathbf{b}, \mathbf{f}\}$, and*
- *$(a_1^\mathcal{I}, a_2^\mathcal{I}) \in p^+(r^\mathcal{I})$ if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(r(a_1, a_2))) \in \{\mathbf{b}, \mathbf{t}\}$.*

Proof. We begin by proving the claim regarding a role assertion $r(a_1, a_2)$. It holds that $(a_1^\mathcal{I}, a_2^\mathcal{I}) \in p^+(r^\mathcal{I})$ iff $(a_1^I, a_2^I) \in r^I$ (due to Definition 3.25) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(r(a_1, a_2))) \in \{\mathbf{b}, \mathbf{t}\}$ is the case (according to Definition 3.6).

The proof of the first two claims is done simultaneously by a structural induction on the structure of C. Let a be an individual in N_I. We consider three base cases, namely that the concept assertion is of the form $\bot(a)$, $\top(a)$ or $A(a)$, where A is an atomic concept. In the case of $\bot(a)$ and $\top(a)$ the claim follows directly because $\bot^\mathcal{I} = \langle \emptyset, \Delta^\mathcal{I}\rangle$ and $\pi(\bot(a))$ is evaluated to $\mathbf{f}$ in every 4-interpretation, and $\top^\mathcal{I} = \langle \Delta^\mathcal{I}, \emptyset\rangle$ and $\pi(\top(a))$ is evaluated to $\mathbf{t}$ in every 4-interpretation. Given an atomic concept assertion A, $a^\mathcal{I} \in p^+(A^\mathcal{I})$ holds iff $a^I \in A^I$ (due to Definition 3.25) iff it is the case that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(A(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by Definition 3.6). Additionally, it holds that $a^\mathcal{I} \in p^-(A^\mathcal{I})$ iff $a^I \notin A^I$ or $a^{I_1} \in A^{I_1}$ (due to Definition 3.25) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(A(a))) \in \{\mathbf{b}, \mathbf{f}\}$ is the case (by Definition 3.6).

Now, assume that the claim holds for the concept description C and consider the concept description $\neg C$. Then $a^\mathcal{I} \in p^+((\neg C)^\mathcal{I})$ holds iff $a^\mathcal{I} \in p^-(C^\mathcal{I})$ (according to Table 2.1) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(a))) \in \{\mathbf{b}, \mathbf{f}\}$ (by the induction hypothesis) iff it is the case

[4]Maier et al. note that there is an extension of $\mathcal{SROIQ}$ that could possibly express such axioms for roles, introduced in [RKH08].

that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(\neg C(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by Definition 3.6). In addition, $a^{\mathcal{I}} \in p^-((\neg C)^{\mathcal{I}})$ holds iff $a^{\mathcal{I}} \in p^+(C^{\mathcal{I}})$ (according to Table 2.1) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by the induction hypothesis) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(\neg C(a))) \in \{\mathbf{b}, \mathbf{f}\}$ is the case (by Definition 3.6).

Next, we perform the induction step for the DL-operator $\sqcap$. So, assume that the claim holds for the concept descriptions C_1 and C_2 and consider the concept description $C_1 \sqcap C_2$. Then $a^{\mathcal{I}} \in p^+((C_1 \sqcap C_2(a))^{\mathcal{I}})$ holds iff $a^{\mathcal{I}} \in p^+(C_1^{\mathcal{I}})$ and $a^{\mathcal{I}} \in p^+(C_2^{\mathcal{I}})$ (according to Table 2.1) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1(a))) \in \{\mathbf{b}, \mathbf{t}\}$ and $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by the induction hypothesis) iff it is the case that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1 \sqcap C_2(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by Definition 3.6). In addition, it holds that $a^{\mathcal{I}} \in p^-((C_1 \sqcap C_2(a))^{\mathcal{I}})$ iff $a^{\mathcal{I}} \in p^-(C_1^{\mathcal{I}})$ or $a^{\mathcal{I}} \in p^-(C_2^{\mathcal{I}})$ (according to Table 2.1) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1(a))) \in \{\mathbf{b}, \mathbf{f}\}$ or $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1(a))) \in \{\mathbf{b}, \mathbf{f}\}$ (by the induction hypothesis) iff it is the case that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1 \sqcap C_2(a))) \in \{\mathbf{b}, \mathbf{f}\}$ (by Definition 3.6).

Now, assume that r is a role name and that the claim holds for the concept description C. Consider the concept description $\exists r.C$. In this case, $a^{\mathcal{I}} \in p^+((\exists r.C(a))^{\mathcal{I}})$ holds iff $\exists y.[(a, y) \in p^+(r^{\mathcal{I}}) \wedge y \in p^+(C^{\mathcal{I}})]$ (according to Table 2.1) iff there is a $y \in \Delta^{\mathcal{I}}$ s.t. $(a, y) \in p^+(r^{\mathcal{I}})$ and $y \in p^+(C^{\mathcal{I}})$ iff there is a $y \in \Delta$ s.t. $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(r(a, y))) \in \{\mathbf{b}, \mathbf{t}\}$ and $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(y))) \in \{\mathbf{b}, \mathbf{t}\}$ (by the induction hypothesis) iff it is the case that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(\exists r.C(a))) \in \{\mathbf{b}, \mathbf{t}\}$ (by Definition 3.6). Moreover, it holds that $a^{\mathcal{I}} \in p^-((\exists r.C(a))^{\mathcal{I}})$ iff $\forall y.[(a, y) \in p^+(r^{\mathcal{I}}) \rightarrow y \in p^-(C^{\mathcal{I}})]$ (according to Table 2.1) iff for all $y \in \Delta^{\mathcal{I}}$ it holds that $(a, y) \in p^+(r^{\mathcal{I}})$ implies that $y \in p^-(C^{\mathcal{I}})$ iff for all $y \in \Delta$ it holds that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(r(a, y))) \in \{\mathbf{b}, \mathbf{t}\}$ implies that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(y))) \in \{\mathbf{b}, \mathbf{f}\}$ (by the induction hypothesis) iff $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(\exists r.C(a))) \in \{\mathbf{b}, \mathbf{f}\}$ (by Definition 3.6).

The induction steps for the other constructors contained in the syntax of $\mathcal{ALC}$ can be shown similarly. □

On the basis of the previous proposition, the main faithfulness result of this section can be shown next.

Theorem 3.28 (Faithfulness w.r.t. $\mathcal{ALC}4$)**.** *Let $\mathcal{O}$ be an ontology expressed in the syntax of $\mathcal{ALC}$ and $\mathcal{I}$ a 4-interpretation such that it holds for each concept C of $\mathcal{O}$ that $p^+(C^{\mathcal{I}}) \cup p^-(C^{\mathcal{I}}) = \Delta^{\mathcal{I}}$. Then $\mathcal{I}$ is a 4-model of $\mathcal{O}$ if and only if its p-counterpart $\langle I, I_1\rangle$ p-satisfies $\pi(\mathcal{O})$.*

Proof. A 4-interpretation $\mathcal{I}$ is a 4-model of $\mathcal{O}$ if and only if it is a 4-model of the ABox $\mathcal{A}$ and the TBox $\mathcal{T}$ of $\mathcal{O}$. It is a 4-model of $\mathcal{A}$ if and only if it is a 4-model of all role and concept assertions in $\mathcal{A}$. The previous is the case if and only if $a^{\mathcal{I}} \in p^+(C^{\mathcal{I}})$ holds for all concept assertions $C(a) \in \mathcal{A}$ and $(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in p^+(r^{\mathcal{I}})$ holds for all role assertions $r(a_1, a_2) \in \mathcal{A}$. According to Proposition 3.27, this holds if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C(a))) \in \{\mathbf{b}, \mathbf{t}\}$ and $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(r(a_1, a_2))) \in \{\mathbf{b}, \mathbf{t}\}$ holds for all $C(a), r(a_1, a_2) \in \mathcal{A}$. As $\pi(\mathcal{A})$ is a conjunction of first-order formulas corresponding to the role and concept assertions in $\mathcal{A}$, we obtain that the following is the case if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(\mathcal{A})) \in \{\mathbf{b}, \mathbf{t}\}$ holds and hence, if and only if $\langle I, I_1\rangle$ p-satisfies the first-order translation of the ABox of $\mathcal{O}$, due to Definition 3.8.

Now, we consider the TBox of $\mathcal{O}$. The 4-interpretation $\mathcal{I}$ is a 4-model of $\mathcal{T}$ if and only if it is a 4-model of each GCI in $\mathcal{T}$. We know due to Table 2.1 that it is a 4-model of a GCI of the form $C_1 \sqsubseteq C_2$ if and only if it is the case that $p^+(C_1^{\mathcal{I}}) \subseteq p^+(C_2^{\mathcal{I}})$. The previous holds if and only if for every individual $a \in N_I$ we have that $a^{\mathcal{I}} \in p^+(C_1^{\mathcal{I}})$ implies that $a^{\mathcal{I}} \in p^+(C_2^{\mathcal{I}})$. Due to Proposition 3.27, this is the case if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1(a))) \in \{\mathbf{b}, \mathbf{t}\}$ implies that $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_2(a))) \in \{\mathbf{b}, \mathbf{t}\}$. The previous is the case if and only if $(\langle I, I_1\rangle, \mathcal{M}, \mathcal{N})(\pi(C_1 \sqsubseteq C_2)) \in \{\mathbf{b}, \mathbf{t}\}$ holds for every GCI $C_1 \sqsubseteq C_2$ in $\mathcal{T}$ and thus, if and only if $\langle I, I_1\rangle$ also p-satisfies $\pi(\mathcal{T})$, according to Definition 3.8. Since $\pi(\mathcal{O}) = \pi(\mathcal{A}) \wedge \pi(\mathcal{T})$, we can conclude that $\mathcal{I}$ is a 4-model of $\mathcal{O}$ if and only if $\langle I, I_1\rangle$ p-satisfies $\pi(\mathcal{O})$. □

The two results mentioned in the beginning of this section now follow directly as corollaries to Theorem 3.28. Firstly, it follows that the p-interpretations in the two sets of a p-model correspond to 4-models in $\mathcal{ALC}4$.

Corollary 3.29. *Let $\mathcal{K} = (\mathcal{O}, \emptyset)$ be a hybrid MKNF knowledge base, where $\mathcal{O}$ is expressed in the syntax of $\mathcal{ALC}$, and (M, N) a p-model of $\mathcal{K}$. Then every p-interpretation $\langle I, I_1\rangle \in M \cup N$ is the p-counterpart of a 4-interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a 4-model of $\mathcal{O} \cup LEM(\mathcal{O})$.*

Proof. Since every p-interpretation $\langle I, I_1\rangle \in M \cup N$ p-satisfies $\pi(\mathcal{O})$ due to Definition 3.10, the corollary follows directly from Theorem 3.28. □

Secondly, the notions of p-entailment and entailment in $\mathcal{ALC}4$ coincide as well.

Corollary 3.30 (Correspondence of p-entailment and entailment in $\mathcal{ALC}4$)**.** *Let $\mathcal{O}$ be an ontology expressed in the syntax of $\mathcal{ALC}$ and A an $\mathcal{ALC}$-axiom. Then $\mathcal{O} \cup LEM(\mathcal{O}) \models_{\mathcal{ALC}4} A$ if and only if $\pi(\mathcal{O}) \models_p \pi(A)$.*

Proof. It holds that $\mathcal{O} \cup LEM(\mathcal{O}) \models_{\mathcal{ALC}4} A$ if and only if every 4-model of $\mathcal{O} \cup LEM(\mathcal{O})$ is also a 4-model of $\mathcal{O} \cup LEM(\mathcal{O}) \cup \{A\}$. According to Theorem 3.28, a 4-interpretation $\mathcal{I}$ is a 4-model of $\mathcal{O} \cup LEM(\mathcal{O})$ if and only if its p-counterpart $\langle I, I_1\rangle$ p-satisfies $\pi(\mathcal{O})$, and it is a 4-model of $\mathcal{O} \cup LEM(\mathcal{O}) \cup \{A\}$ if and only if $\langle I, I_1\rangle$ p-satisfies $\pi(\mathcal{O} \cup \{A\})$. Consequently, $\pi(\mathcal{O}) \models_p \pi(A)$ holds if and only if the previous is the case. □

Consequently, algorithms developed for $\mathcal{ALC}4$ can be used to derive consequences from the ontology in the alternating fixpoint construction presented in the next Chapter.

3.9 The Well-Founded Paraconsistent Model

It is obvious from Example 3.12 that there can be several p-models for a given ground hybrid MKNF knowledge base. As has already been hinted, we aim at singling out one specific p-model for every hybrid MKNF knowledge base which is the most "skeptical" p-model in that it makes the least assumptions about knowledge contained in a knowledge base being either "true" or not "true or undefined". This notion of "skeptical" reasoning is realized by a p-model that assumes the least possible number of modal **K**-atoms to

be either *true* (resp. *inconsistent* or *suspiciously true*) or *false* (resp. *inconsistent*, *suspiciously true*, or *classically false*) by leaving everything *undefined* that is not evaluated either to one of the formerly mentioned truth values, or to one of the latter values, in all p-models of a knowledge base.

Next, we provide a formal definition of a p-model which has the mentioned property. For this purpose, we adapt the notion of a *well-founded MKNF model* from [KAH11]. The following two definitions are taken directly from [KAH11] and adapted to our framework. As Knorr et al. note, MKNF interpretation pairs can be compared by defining an order which is similar to the knowledge order used in logic programming [KAH11]. The same can be done in the case of p-interpretation pairs.

Definition 3.31 (Knowledge order of p-interpretation pairs)**.** Let (M_1, N_1) and (M_2, N_2) be p-interpretation pairs. We have that $(M_1, N_1) \succeq_k (M_2, N_2)$ iff $M_1 \subseteq M_2$ and $N_1 \supseteq N_2$.

Regarding the truth evaluation w.r.t. a p-interpretation pair, the truth value *undefined* is the least element of the order induced by the previous definition. For instance, a p-interpretation pair (M_2, N_2) interpreting an MKNF formula only consisting of a single **K**-atom which is evaluated to **u** by (M_2, N_2) is smaller in the order than any p-interpretation pair (M_1, N_1) that evaluates the atom to another truth value. The order defined in Definition 3.31 can now be applied for comparing p-models.

Definition 3.32 (Well-founded paraconsistent model)**.** Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base and (M, N) a p-model of $\mathcal{K}_G$ such that $(M_1, N_1) \succeq_k (M, N)$ for all p-models (M_1, N_1) of $\mathcal{K}_G$. Then (M, N) is a *well-founded paraconsistent model (or well-founded p-model)* of $\mathcal{K}_G$.

Consequently, the well-founded p-model of a hybrid MKNF knowledge base is the one that leaves as much as possible *undefined*. At the same time, a modal **K**-atom is *inconsistent*, *suspiciously true* or *true* in the well-founded p-model of a hybrid MKNF knowledge base $\mathcal{K}_G$ if and only if it is *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$; and it is *false*, *inconsistent*, *suspiciously true*, or *classically false* in the well-founded p-model $\mathcal{K}_G$ if and only if it is *false*, *inconsistent*, *suspiciously true*, or *classically false* in every p-model of $\mathcal{K}_G$. This property will be proven in Chapter 4.

A well-founded p-model of a hybrid MKNF knowledge base always exists and it is unique, as the following theorem states. The same theorem w.r.t. three-valued well-founded MKNF models occurs nearly identical in the publication by Knorr et al. [KAH11].

Theorem 3.33 (Existence and uniqueness of p-models)**.** *If $\mathcal{K}$ is a DL-safe hybrid MKNF knowledge base where the DL $\mathcal{ALC}$ is used for formalizing the ontology component, then a well-founded p-model exists, and it is unique.*

The restriction on the syntax of the DL used for formalizing the ontology component in the previous theorem is imposed in order to ensure that the ontology is always p-satisfied by some p-interpretation. According to Maier et al. [MMH13], it can be guaranteed that a 4-model of an $\mathcal{ALC}4$-ontology exists. Nevertheless, this result can also be

obtained if the syntax of the DL used in the ontology component is extended by certain additional constructs, such as *number restrictions* and *role composition* [MMH13]. In the remainder of this thesis, we assume that the DL used for expressing the ontology component of a hybrid MKNF knowledge base always admits a 4-model.

The proof of the theorem is postponed until Chapter 4, where we show how the well-founded p-model of a knowledge base can be obtained by means of an alternating fixpoint construction. There, we will show the correctness and completeness of this procedural characterization w.r.t. the model-theoretic definition of well-founded p-models presented here, such that Theorem 3.33 will be a direct consequence of this proof.

It is now possible to check that one of the three p-models of the knowledge base $\mathcal{K}_G$ in Example 3.12 is in fact the well-founded p-model of $\mathcal{K}_G$.

Example 3.34. Consider again the ground hybrid MKNF knowledge base $\mathcal{K}_G$ discussed in Example 3.12. As we have seen, this knowledge base has three p-models resulting from a recursion through default negation in the second and the third MKNF rule. We recall these two rules here.

$$\begin{aligned} \mathbf{K}\,\mathsf{risk(food)} &\leftarrow \mathbf{not}\,\mathsf{isLabelled(food)}. \\ \mathbf{K}\,\mathsf{isLabelled(food)} &\leftarrow \mathbf{not}\,\mathsf{risk(food)}. \end{aligned}$$

Let (M_1, N_1), (M_2, N_2) and (M, N) be the three p-models of the two MKNF rules alone, corresponding to the p-models obtained by the first, second and third option presented in Example 3.12, respectively. Since the **K**-atom $\mathbf{K}\,\mathsf{risk(food)}$ is *true* in (M_1, N_1), risk(food) is not *false* in any p-interpretation in M_1 or N_1 and it is *true* in at least one p-interpretation in M_1, according to the definition of the **K**-operator in Definition 3.6. On the other hand, isLabelled(food) is *false* in at least one p-interpretation in M_1 and N_1, and it is *true* in at least one p-interpretation in M_1 because $\mathbf{K}\,\mathsf{isLabelled(food)}$ is *false* in (M_1, N_1). Regarding the p-model (M, N), both **K**-atoms $\mathbf{K}\,\mathsf{risk(food)}$ and $\mathbf{K}\,\mathsf{isLabelled(food)}$ are *undefined*. Hence, risk(food) and isLabelled(food) are *false* as well as *true* in at least one p-interpretation in M, and they are both not *false* in any p-interpretation in N, by Definition 3.6. Consequently, it holds that $M_1 \subseteq M$ and $N_1 \supseteq N$ since M, N_1, M_1 and N are the maximal sets having the mentioned properties (due to knowledge minimization by condition (2) of Definition 3.10). As a result, we obtain that $(M_1, N_1) \succeq_k (M, N)$. By the same line of reasoning, $(M_2, N_2) \succeq_k (M, N)$ holds as well. According to Definition 3.32, (M, N) is in fact the well-founded p-model of $\mathcal{K}_G$, at least when only considering the second and the third rule. It can be shown that the same holds when considering the whole knowledge base $\mathcal{K}_G$. $\Diamond$

3.10 Finite Representations of Paraconsistent MKNF Models

As described by Knorr et al. [KAH11] and originally proposed by Motik and Rosati [MR10], we also aim at having a finite model-representation in the case of paraconsistent MKNF models, which is not given if the domain is countably infinite [KAH11]. For this purpose, we represent a p-model by means of two finite first-order formulas obtained via the objective knowledge defined in Definition 3.9, so that the sets of p-interpretations that satisfy them correspond to the given p-model. For this, we extend Definition 15 from [KAH11] and speak of a pair of subsets induced by a paraconsistent MKNF interpretation pair, instead of a partition, as **K**-atoms can be in T and F at the same time now according to the following definition.

Definition 3.35 (The induced pair of a p-model)**.** Given a set S of ground **K**-atoms and a p-interpretation pair (M, N), the pair (T, F) of subsets of S is *induced by* (M, N) as follows:

(1) $\mathbf{K}\xi \in T$ if and only if $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$, and

(2) $\mathbf{K}\xi \in F$ if and only if $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) \in \{\mathbf{b}, \mathbf{st}, \mathbf{f}, \mathbf{cf}\}$.

The statement expressed by the following lemma corresponds to the Coherence Principle and thus, postulates that classical negation implies negation by failure in the model-theoretic definition of our semantics.

Lemma 3.36 (The Coherence Principle w.r.t. model-theoretic definition)**.** *Let (M, N) be a p-model of a ground hybrid MKNF knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, (T, F) the pair of subsets of $\mathsf{KA}(\mathcal{K}_G)$ induced by (M, N) and $\mathbf{K}\xi$ a **K**-atom in $\mathsf{KA}(\mathcal{K}_G)$. Then $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$ implies that $\mathbf{K}\xi \in F$.*

Proof. Condition (3) of Definition 3.10 states that $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$ holds if and only if it is the case that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) \in \{\mathbf{b}, \mathbf{cf}\}$. From case (2) of Definition 3.35, we can directly infer that $\mathbf{K}\xi$ has to be in F. □

Obviously, we also know that the contrapositive of Lemma 3.36 is true, namely that if some **K**-atom $\mathbf{K}\xi$ is not in F, then we also have that $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$. We will use this property in the proof of Proposition 3.37, which provides a means to obtain a p-model from the pair induced by that model. This result will also be used to show that the p-model we compute by means of the fixpoint calculation introduced in the next chapter is in fact the well-founded p-model.

Knorr et al. show that the finite representations induced by three-valued MKNF models correspond exactly to the given three-valued MKNF model. We adapt their result to the notion of the induced pair of a p-model defined in Definition 3.35. The following proposition and its proof are similar to Proposition 3 and the corresponding proof in [KAH11].

Proposition 3.37 (Correspondence of p-models and the induced pair)**.** *Let (M, N) be a p-model of a ground hybrid MKNF knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, and (T, F) the pair of subsets of $\mathsf{KA}(\mathcal{K}_G)$ induced by (M, N). Then $(M, N) = (\{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},T}\}, \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F}\})$.*

Proof. Given a ground hybrid MKNF knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$, let (M, N) be a p-model of $\mathcal{K}_G$, (T, F) the pair of subsets of $\mathsf{KA}(\mathcal{K}_G)$ induced by (M, N), and $(M', N') = (\{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},T}\}, \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F}\})$. We prove that $(M, N) = (M', N')$.

First, we prove $M \subseteq M'$. Accordingly, we prove that for each p-interpretation $\mathcal{I} \in M$ it holds that $\mathcal{I} \in M' = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},T}\}$, i.e. that $\mathcal{I} \models_p \{\pi(\mathcal{O})\}$ and $\mathcal{I} \models_p \{\xi \mid \mathbf{K}\,\xi \in T\}$. Since (M, N) is a p-model of $\mathcal{K}_G$, we know that $(M, N) \models_p \pi(\mathcal{O})$. Thus, we have that $\mathcal{I} \models_p \pi(\mathcal{O})$ for each $\mathcal{I} \in M$. Now, consider each $\mathbf{K}\,\xi \in T$. Since (M, N) induces the pair (T, F), we know that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$. Thus, we obtain $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in M$ by Definition 3.6, and therefore, $\mathcal{I} \models_p \xi$ for each $\mathcal{I} \in M$. This proves that $\mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},T}$ for each $\mathcal{I} \in M$. Hence, for each $\mathcal{I} \in M$ it holds that $\mathcal{I} \in M'$ and consequently, $M \subseteq M'$.

Next, we prove $N \subseteq N'$. So, we have to prove that for each $\mathcal{I} \in N$ it holds that $\mathcal{I} \in N' = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F}\}$, i.e. that $\mathcal{I} \models_p \{\pi(\mathcal{O})\}$ and $\mathcal{I} \models_p \{\xi \mid \mathbf{K}\xi \in \mathsf{KA}(\mathcal{K}_G) \setminus F\}$. Again, since $(M, N) \models_p \pi(\mathcal{O})$, we know that $\mathcal{I} \models_p \pi(\mathcal{O})$ for each $\mathcal{I} \in N$. Consider each $\mathbf{K}\xi \in \mathsf{KA}(\mathcal{K}_G)$ with $\mathbf{K}\xi \notin F$. By Definition 3.35, we know that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) \notin \{\mathbf{b}, \mathbf{st}, \mathbf{f}, \mathbf{cf}\}$ and hence, that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi) \in \{\mathbf{t}, \mathbf{u}\}$. Thus, by Definition 3.6, we obtain that $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in N$ and therefore, also that $\mathcal{I} \models_p \xi$ for each $\mathcal{I} \in N$. This suffices to prove that $\mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F}$ for each $\mathcal{I} \in N$, i.e. that $\mathcal{I} \in N'$ for each $\mathcal{I} \in N$, from which $N \subseteq N'$ follows.

We now prove that in fact $(M, N) = (M', N')$, i.e. $M = M'$ and $N = N'$. So, assume the contrary, namely that (M', N') is a p-interpretation pair with $M \subseteq M'$ and $N \subseteq N'$ where at least one of the inclusions is proper. We prove that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. In this way, we derive a contradiction to (M, N) being a p-model of $\mathcal{K}_G$. For the former, it is enough to prove that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\pi(\mathcal{O}) \wedge \pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. By definition of M' and N' we know that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\pi(\mathcal{O})) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. We only have to prove the same for $\pi(\mathcal{P}_G)$. We achieve this by proving that for each possible combination of the cases in Definition 3.35, the modal atoms appearing in $\pi(\mathcal{P}_G)$ are evaluated to identical truth values in the p-structures $(*, \langle M, N\rangle, \langle M, N\rangle)$ and $(*, \langle M', N'\rangle, \langle M, N\rangle)$. Since (M, N) being a p-model of $\mathcal{K}_G$ ensures that $(*, \langle M, N\rangle, \langle M, N\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$, this suffices to prove that $(*, \langle M', N'\rangle, \langle M, N\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$. So, consider an arbitrary $\mathbf{K}$-atom $\mathbf{K}\xi$ in $\mathsf{KA}(\mathcal{K}_G)$.

- Suppose that $\mathbf{K}\,\xi \in T$ and $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$. From Definition 3.35 and condition (3) of Definition 3.10, it follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,\xi) = \mathbf{b}$. By definition of M', we obtain $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M'$. Because we have that $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$, we conclude $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ is not possible for any

$\mathcal{I}' \in M'$. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{b}$ as well.

- Suppose that $\mathbf{K}\xi \in T$, $\mathbf{K}\xi \in F$ and $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$. From Definition 3.35 and condition (3) of Definition 3.10, it follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{st}$. We obtain $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M'$ by definition of M' as in the previous case. Since we have that $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$, we derive $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ for some $\mathcal{I}' \in M'$. We know that $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \mathbf{f}$ for some $\mathcal{I} \in N$ by Definition 3.6 and since N is a subset of N', the same holds for some $\mathcal{I} \in N'$. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{st}$ as well.

- Suppose that $\mathbf{K}\xi \in T$ and $\mathbf{K}\xi \notin F$. It follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{t}$ by Definition 3.35. We obtain $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$ by definition of M' and N'. From the fact that $\mathbf{K}\xi$ is not in F, we can infer $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$ due to Lemma 3.36. Thus, we derive that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ for some $\mathcal{I}' \in M'$. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{t}$ as well.

- Suppose that $\mathbf{K}\xi \notin T$, $\mathbf{K}\xi \in F$ and $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$. From Definition 3.35 and condition (3) of Definition 3.10, it follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{f}$. We know that $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \mathbf{f}$ for some $\mathcal{I} \in M$ and for some $\mathcal{I} \in N$ by Definition 3.6 and since M is a subset of M' and N is a subset of N', the same holds for some $\mathcal{I}' \in M'$ and for some $\mathcal{I}' \in N'$. As we also have that $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$, we conclude $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ for some $\mathcal{I}' \in M'$. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{f}$ as well.

- Suppose that $\mathbf{K}\xi \notin T$ and $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$. From Definition 3.35 and condition (3) of Definition 3.10, it follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{cf}$. As before, we can derive $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{f}$ for some $\mathcal{I}' \in M'$. Since we have that $\mathsf{OB}_{\mathcal{O},T} \models_p \neg\xi$, we conclude $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ is not possible for any $\mathcal{I}' \in M'$. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{cf}$ as well.

- Suppose that $\mathbf{K}\xi \notin T$ and $\mathbf{K}\xi \notin F$. It follows that $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{u}$, by Definition 3.35. We know that $(\mathcal{I}, \langle M, N\rangle, \langle M, N\rangle)(\xi) = \mathbf{f}$ for some $\mathcal{I} \in M$ by Definition 3.6 and since M is a subset of M', the same holds for some $\mathcal{I}' \in M'$. Because $\mathbf{K}\xi$ is not in F, we can infer that $\mathsf{OB}_{\mathcal{O},T} \not\models_p \neg\xi$ by Lemma 3.36. We conclude $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) = \mathbf{t}$ for some $\mathcal{I}' \in M'$. Additionally, we know that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\xi) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in N'$ by definition of N'. Hence, by Definition 3.6, we obtain $(*, \langle M', N'\rangle, \langle M, N\rangle)(\mathbf{K}\xi) = \mathbf{u}$ as well.

- For modal **not**-atoms appearing in $\pi(\mathcal{P}_G)$ we directly obtain the identical evaluations because they are interpreted w.r.t. (M, N) in each case.

Consequently, we obtain a contradiction to (M, N) being a p-model of $\mathcal{K}_G$ and conclude that $(M, N) = (M', N')$. □

As discussed in Section 3.6, dependencies on contradictions that occur in the program component are propagated via the ontology, while dependencies on contradictions that can be derived without involving *inconsistent* knowledge from the program component are not propagated in our framework. It has been illustrated in Example 3.13 and Example 3.14 that this behavior can be achieved by deliberately relaxing the conditions for a modal **K**-formula being interpreted with the truth value $\mathbf{b}$ in Definition 3.6. As a result, a **K**-formula is evaluated to $\mathbf{b}$ in a p-interpretation pair (M, N) if and only if $(\bigcap_{\mathcal{J} \in M} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) = \mathbf{b}$, and either $(\bigcap_{\mathcal{J} \in N} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) \neq \mathbf{f}$ or $(\bigcap_{\mathcal{J} \in N} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) = \mathbf{f}$ holds. Hereby, inconsistencies interpreted in the former way are propagated via the ontology, while those which are defined in the latter way are not. The following corollary to Proposition 3.37 demonstrates that those **K**-atoms which are interpreted with $\mathbf{b}$ such that $(\bigcap_{\mathcal{J} \in N} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) \neq \mathbf{f}$ are indeed those modal **K**-atoms which can be derived from the ontology without involving any contradictory knowledge from the program component (i.e. from the ontology and *true* as well as *undefined* **K**-atoms alone). It might seem unintuitive that we have to consider *undefined* **K**-atoms here as well. However, this results from the fact that Contradiction Support Detection is not complete and fails whenever a contradictory **K**-atom can also be derived from *undefined* knowledge, as described in Section 2.3.3. Corollary 3.38 is used in the proof of Proposition 4.18 in Chapter 5 to distinguish between inconsistencies that are propagated within the ontology and those which are not.

Corollary 3.38. *Let (M, N) be a p-model of a ground hybrid MKNF knowledge base $\mathcal{K}_G$ and $\mathbf{K}\,H$ a **K**-atom in $\mathsf{KA}(\mathcal{K}_G)$. Then $(\bigcap_{\mathcal{J} \in M} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) = \mathbf{b}$ and $(\bigcap_{\mathcal{J} \in N} \mathcal{J}, \langle M, N \rangle, \mathcal{N})(H) \neq \mathbf{f}$ if and only if $\mathsf{OB}_{\mathcal{O},T} \models_p H$, $\mathsf{OB}_{\mathcal{O},T} \models_p \neg H$ and $\mathsf{OB}_{\mathcal{O},\{\mathbf{K}\,H' \mid (*,\langle M,N \rangle,\langle M,N \rangle)(\mathbf{K}\,H') \in \{\mathbf{t},\mathbf{u}\}\}} \models_p H$.*

Proof. The corollary follows directly from Definition 3.35 and Proposition 3.37. □

4

Alternating Fixpoint Characterization

In this chapter, we show how the well-founded p-model of a hybrid MKNF knowledge base introduced in the last chapter can be obtained procedurally by means of an alternating fixpoint computation. Since the ontology component is considered as an oracle in this computation, the procedural definition of well-founded p-models reflects the alternating fixpoint computation of the paraconsistent well-founded model of an extended logic program in $WFSX_p$, which has been presented in Section 2.3.2. In addition, the algorithm for obtaining the paraconsistent well-founded model of $WFSX_p$ works basically identical to the algorithm for obtaining the consistent well-founded model of $WFSX$ (the precursor of $WFSX_p$ not accounting for paraconsistent reasoning) in that only one condition has to be relaxed [Dam96]. Knorr et al. [KAH08] adapt the latter construction in their approach by following the principle of computing a fixpoint by alternately applying two anti-monotonic Γ-operators (defined for hybrid MKNF knowledge bases in this case) and by using a transformation of a knowledge base resembling the semi-normal version for LPs in order to enforce the Coherence Principle (at least in the version of the approach presented in [KAH08]). Consequently, our definitions underlying the alternating fixpoint construction of a well-founded p-model closely resemble the definitions provided by Knorr et al. in [KAH11], and can partly be adopted directly. Additionally, several results concerning the alternating fixpoint construction carry over directly from [KAH11], so that we will just refer to the proofs given by Knorr et al. in these cases. Accordingly, it also holds for the alternating fixpoint construction employed here that a least fixpoint can always be obtained provided the DL used for formalizing the ontology component adheres to the restrictions imposed in Theorem 3.33.

At the same time, the details of the proofs showing the correspondence between the procedural and the declarative characterization of our semantics become more intricate

due to the fact that six different truth values have to be accounted for. After the adaptation of the alternating fixpoint construction from [KAH11] to our approach, we will first show that the well-founded pair of a hybrid MKNF knowledge base obtained by this construction corresponds to a p-model of the knowledge base. Afterwards, we will prove that it exactly corresponds to the well-founded p-model of a knowledge base, i.e. that the construction is sound and complete w.r.t. the model-based definition of well-founded p-models. Since a least fixpoint of the construction always exists, the latter result implies that a unique well-founded p-model exists for every hybrid MKNF knowledge base in case the ontology component of the knowledge base is always p-satisfied by some p-interpretation. On the basis of the formalization of our paraconsistent well-founded hybrid MKNF semantics, the faithfulness result w.r.t. the semantics of $WFSX_p$ for LPs follows straightforwardly due to the close relation between the different formalizations of the alternating fixpoint constructions applied in the two approaches. Furthermore, we will discuss the complexity of computing the well-founded p-model by means of the alternating fixpoint construction in the end of this chapter.

4.1 Derivation of Immediate Consequences

We begin by defining how the set of all modal **K**-atoms that can be derived from a positive ground hybrid MKNF knowledge base, i.e. a hybrid MKNF knowledge base where all MKNF rules in the program component are positive according to Definition 2.13, can be obtained. The following definition is adapted from Definition 16 in [KAH11] to our framework.

Definition 4.1 (Immediate consequence operators)**.** Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a positive, ground hybrid MKNF knowledge base. The operators $R_{\mathcal{K}_G}$, $D_{\mathcal{K}_G}$, $T_{\mathcal{K}_G}$ and $T'_{\mathcal{K}_G,\mathcal{C}}$ are defined on subsets of $\mathsf{KA}(\mathcal{K}_G)$ as follows.

$$\begin{aligned}
R_{\mathcal{K}_G}(S) &= \{\mathbf{K}H \mid \mathcal{P}_G \text{ contains a rule of the form } \mathbf{K}H \leftarrow \mathbf{K}A_1, \ldots, \mathbf{K}A_n \\
&\quad \text{such that, for all } i,\ 1 \leq i \leq n,\ \mathbf{K}A_i \in S\} \\
D_{\mathcal{K}_G}(S) &= \{\mathbf{K}\xi \mid \mathbf{K}\xi \in \mathsf{KA}(\mathcal{K}_G) \text{ and } \mathsf{OB}_{\mathcal{O},S} \models_p \xi\} \\
T_{\mathcal{K}_G}(S) &= R_{\mathcal{K}_G}(S) \cup D_{\mathcal{K}_G}(S) \\
T'_{\mathcal{K}_G,\mathcal{C}}(S) &= (R_{\mathcal{K}_G}(S) \cup D_{\mathcal{K}_G}(S)) \setminus \{\mathbf{K}\xi \mid \mathbf{K}\xi \in \mathsf{KA}(\mathcal{K}_G) \text{ and } \mathsf{OB}_{\mathcal{O},\mathcal{C}} \models_p \neg\xi\}
\end{aligned}$$

In contrast to the T-operator presented in Section 2.3, which is applied repeatedly w.r.t. a positive logic program to derive all of its consequences and thus, to compute its minimal model, the operators $T_{\mathcal{K}_G}$ and $T'_{\mathcal{K}_G,\mathcal{C}}$ introduced in the previous definition integrate the results of two "sub-operators", $R_{\mathcal{K}_G}$ and $D_{\mathcal{K}_G}$. While the operator $R_{\mathcal{K}_G}$ derives consequences from the program component just like the T-operator in the case of LPs, the operator $D_{\mathcal{K}_G}$ derives all modal **K**-atoms which are p-entailed by the ontology

component (after adding those first-order atoms that correspond to a **K**-atom that has already been derived). The definition here differs from the one in [KAH11] in that $T'_{\mathcal{K}_G,\mathcal{C}}$ is introduced as a second immediate consequence operator in order to implement the Coherence Principle in our approach. How this strategy relates to the one pursued by Knorr et al. will be discussed in detail in the next section, where we will also demonstrate the need for two immediate consequence operators with an example. Moreover, another difference consists in the fact that we use the p-entailment operator $\models_p$ to derive **K**-atoms from the ontology component, and not the classical first-order entailment operator $\models$. As already mentioned, for certain tractable DLs such as $\mathcal{EL}^{++}$ and $DL\text{-}Lite$, standard reasoners can be applied to derive these paraconsistent consequences in the computation of immediate consequences (cf. [MMH13]).

Knorr et al. show that the $T_{\mathcal{K}_G}$-operator in their framework is monotonic (cf. Proposition 4 in [KAH11]). Now, the definition of the $T_{\mathcal{K}_G}$-operator in Definition 4.1 only differs in terms of the entailment operator applied. Further, the entailment operator used in the paraconsistent approach by Maier et al. is monotonic [MMH13] and thus, the p-entailment operator $\models_p$ is monotonic as well due to Corollary 3.30. As a result, the proof of monotonicity of the operator $T_{\mathcal{K}_G}$ can be directly transferred to the operator $T_{\mathcal{K}_G}$ used in our approach, i.e. $T_{\mathcal{K}_G}$ as defined above is monotonic. Moreover, the operator $T'_{\mathcal{K}_G,\mathcal{C}}$ is monotonic as well because only a fixed set of modal **K**-atoms is removed from the results of the operators $R_{\mathcal{K}_G}$ and $D_{\mathcal{K}_G}$ in every application of the operator. As a result, the operators $T_{\mathcal{K}_G}$ and $T'_{\mathcal{K}_G,\mathcal{C}}$ have a least fixpoint by the *Knaster-Tarski Theorem*, as noted by Knorr et al. [KAH11]. As in [KAH11], we denote the least fixpoint of the two operators by $T_{\mathcal{K}_G} \uparrow \omega$ and $T'_{\mathcal{K}_G,\mathcal{C}} \uparrow \omega$, respectively. Here, ω denotes the limit ordinal of natural numbers and the computation of the respective fixpoint is reached after finitely many iterations, according to Knorr et al. [KAH11]. The least fixpoint of $T_{\mathcal{K}_G}$ is obtained as follows (corresponding to the formulation in [KAH11]):

$$\begin{aligned} T_{\mathcal{K}_G} \uparrow 0 &= \emptyset \\ T_{\mathcal{K}_G} \uparrow (n+1) &= T_{\mathcal{K}_G}(T_{\mathcal{K}_G} \uparrow n) \\ T_{\mathcal{K}_G} \uparrow \omega &= \bigcup_{i \geq 0} T_{\mathcal{K}_G} \uparrow i \end{aligned}$$

Further, the least fixpoint of $T'_{\mathcal{K}_G,\mathcal{C}}$ is obtained in the same way. Intuitively, the least fixpoint $T_{\mathcal{K}_G} \uparrow \omega$ contains all modal **K**-atoms that can be derived from the hybrid MKNF knowledge base $\mathcal{K}_G$ and contains nothing else and thus, can be viewed as the counterpart to the minimal model of an LP. Similarly, $T'_{\mathcal{K}_G,\mathcal{C}} \uparrow \omega$ contains all **K**-atoms that can be derived without involving any **K**-atom whose classical negation is p-entailed by the objective knowledge of $\mathcal{C}$ w.r.t. $\mathcal{K}_G$.

As the immediate consequence operators discussed before are only applicable to positive knowledge bases, we need a means to obtain a positive version of a ground hybrid MKNF knowledge base. This can be done by defining a transformation of ground hybrid

MKNF knowledge bases into a positive version, which resembles the GL-transformation discussed in Section 2.3. While Knorr et al. have to define two different kinds of transformation in their approach to enforce the Coherence Principle by following a strategy similar to the idea of using the semi-normal version of a program for certain derivations in $WFSX_p$, we only have to define a single transformation since we move the implementation of the Coherence Principle into the computation of immediate consequences. The following definition of the *MKNF transform* is identical to the definition of the first transformation provided by Knorr et al. in [KAH11].

Definition 4.2 (MKNF transform [KAH11])**.** Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground hybrid MKNF knowledge base and $S \subseteq \mathsf{KA}(\mathcal{K}_G)$. The *MKNF transform* $\mathcal{K}_G/S$ is defined as $\mathcal{K}_G/S = (\mathcal{O}, \mathcal{P}_G/S)$, where $\mathcal{P}_G/S$ contains all rules

$$\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots, \mathbf{K}\,A_n$$

for which there exists a rule

$$\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots, \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$$

in $\mathcal{P}_G$ with $\mathbf{K}\,B_j \notin S$ for each $1 \leq j \leq m$.

The following example illustrates the two notions of the MKNF transform and the direct consequences of a ground hybrid MKNF knowledge base.

Example 4.3. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$, which corresponds to the hybrid knowledge base shown in Example 1.1 and is only grounded with the constant pesticide (omitting the constant food and the predicate good). Furthermore, we compute the MKNF transform $\mathcal{K}_G/S$ where S is equal to $\{\mathbf{K}\,\mathsf{isLabelled(pesticide)}, \mathbf{K}\,\mathsf{HasCertifiedForwarder(pesticide)}\}$, which results in the following positive ground hybrid MKNF knowledge base:

$$\begin{aligned}
\mathsf{ToxicChemical} &\sqsubseteq \exists\mathsf{Contains.ToxicSubstance}\\
\exists\mathsf{Contains.ToxicSubstance} &\sqsubseteq \mathsf{ProvenRisk}\\
\mathsf{ProvenRisk} \sqcup \mathsf{PotentialRisk} &\sqsubseteq \mathsf{Risk}\\
\mathsf{HasCertifiedForwarder} &\sqsubseteq \neg\mathsf{IsMonitored}\\
\mathsf{ToxicChemical(pesticide)}
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}\,\mathsf{IsMonitored(pesticide)} &\leftarrow \mathbf{K}\,\mathsf{Risk(pesticide)}.\\
\cancel{\mathbf{K}\,\mathsf{PotentialRisk(pesticide)} \leftarrow \mathbf{not}\,\mathbf{K}\,\mathsf{isLabelled(pesticide)}.}\\
\mathbf{K}\,\mathsf{resolvedRisk(pesticide)} &\leftarrow \mathbf{K}\,\mathsf{IsMonitored(pesticide)}.\\
\mathbf{K}\,\mathsf{isLabelled(pesticide)} &\leftarrow
\end{aligned}$$

Note that the second MKNF rule is not contained in the MKNF transform $\mathcal{K}_G/S$, indicated by canceling out this rule. We now show the computation of the least fixpoint of the operator $T'_{\mathcal{K}_G/S,\mathcal{C}}$ according to Definition 4.1, where we assume that $\mathcal{C} = S$. In the computation, we abbreviate pesticide, isLabelled, isMonitored, resolvedRisk, ToxicChemical, ProvenRisk and Risk by p, l, m, resR, TC, PrR and R, respectively.

$$
\begin{aligned}
T'_{\mathcal{K}_G/S,\mathcal{C}} \uparrow 0 &= \emptyset \\
T'_{\mathcal{K}_G/S,\mathcal{C}} \uparrow 1 &= (\{\mathbf{K}\,\mathsf{l(p)}\} \cup \{\mathbf{K}\,\mathsf{TC(p)}\}) \setminus \{\mathbf{K}\,\mathsf{m(p)}\} \\
T'_{\mathcal{K}_G/S,\mathcal{C}} \uparrow 2 &= (\{\mathbf{K}\,\mathsf{l(p)}\} \cup \{\mathbf{K}\,\mathsf{TC(p)}, \mathbf{K}\,\mathsf{PrR(p)}, \mathbf{K}\,\mathsf{R(p)}\}) \setminus \{\mathbf{K}\,\mathsf{m(p)}\} \\
T'_{\mathcal{K}_G/S,\mathcal{C}} \uparrow 3 &= (\{\mathbf{K}\,\mathsf{l(p)}, \mathbf{K}\,\mathsf{m(p)}\} \cup \{\mathbf{K}\,\mathsf{TC(p)}, \mathbf{K}\,\mathsf{PrR(p)}, \mathbf{K}\,\mathsf{R(p)}\}) \setminus \{\mathbf{K}\,\mathsf{m(p)}\} \\
T'_{\mathcal{K}_G/S,\mathcal{C}} \uparrow \omega &= \{\mathbf{K}\,\mathsf{l(p)}, \mathbf{K}\,\mathsf{TC(p)}, \mathbf{K}\,\mathsf{PrR(p)}, \mathbf{K}\,\mathsf{R(p)}\}
\end{aligned}
$$

First, note that the result of the immediate consequence operator is in fact monotonically increasing in every iteration and that the ontology and the program component interact in the derivation of consequences. Because the **K**-atom **K** HasCertifiedForwarder(pesticide) is contained in the set $\mathcal{C}$, ¬isMonitored(pesticide) is p-entailed by the objective knowledge of $\mathcal{C}$ w.r.t. $\mathcal{K}_G/S$. Consequently, **K** isMonitored(pesticide) is not contained in the least fixpoint of $T'_{\mathcal{K}_G/S,\mathcal{C}}$, even though it can be derived by means of the program component (cf. iteration 3). This illustrates that the Coherence Principle is implemented in the operator $T'_{\mathcal{K}_G,\mathcal{C}}$. Additionally, the deletion of modal **K**-atoms whose classical negation can be derived from the ontology also affects other **K**-atoms even if their classical negation is not derivable. For example, the **K**-atom **K** resolvedRisk(pesticide) cannot be derived because **K** isMonitored(pesticide) is deleted from the results of the operators $R_{\mathcal{K}_G}$ and $D_{\mathcal{K}_G}$. As a result, inconsistencies are propagated in the alternating fixpoint construction presented in the next section. In the least fixpoint of the operator $T_{\mathcal{K}_G/S}$, **K** isMonitored(pesticide) is not deleted and thus, we obtain $T_{\mathcal{K}_G/S} \uparrow \omega = \{\mathbf{K}\,\mathsf{l(p)}, \mathbf{K}\,\mathsf{TC(p)}, \mathbf{K}\,\mathsf{PrR(p)}, \mathbf{K}\,\mathsf{R(p)}, \mathbf{K}\,\mathsf{m(p)}, \mathbf{K}\,\mathsf{resR(p)}\}$. ◊

Just like in the case of the SMS and the WFS for LPs, we can combine the transformation of a ground hybrid MKNF knowledge base by means of the MKNF transform defined above and the derivation of all immediate consequences of the resulting positive knowledge base by means of the operator $T_{\mathcal{K}_G}$ (resp. $T'_{\mathcal{K}_G,\mathcal{C}}$) within a single Γ-operator. In the following definition, similar to Knorr et al. [KAH11], we define two distinct Γ-operators, one for each of our two immediate consequence operators introduced in Definition 4.1.

Definition 4.4 (The operators $\Gamma_{\mathcal{K}_G}$ and $\Gamma'_{\mathcal{K}_G}$)**.** Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground hybrid MKNF knowledge base and $S \subseteq \mathsf{KA}(\mathcal{K}_G)$. We define the two operators $\Gamma_{\mathcal{K}_G}(S) = T_{\mathcal{K}_G/S} \uparrow \omega$, and $\Gamma'_{\mathcal{K}_G}(S) = T'_{\mathcal{K}_G/S,S} \uparrow \omega$.

Note that the $\Gamma'_{\mathcal{K}_G}$-operator is defined in terms of the operator $T'_{\mathcal{K}_G/S,\mathcal{C}}$ where $\mathcal{C} = S$. Consequently, checking for the derivability of the classical and the default negation of

atoms is done w.r.t. the same set, which corresponds to the approach by Knorr et al. [KAH11]. In the fixpoint computation of the well-founded p-model introduced in the next section, S will contain all **K**-atoms that could already be proven to be *true* (resp. *inconsistent* or *suspiciously true*). Before presenting the definition of the alternating fixpoint construction of our semantics, we just have to transfer one result proven in [KAH11] to our approach. The following Lemma states that both the $\Gamma_{\mathcal{K}_G}$- and $\Gamma'_{\mathcal{K}_G}$-operator are anti-monotonic.

Lemma 4.5 (Anti-monotonicity of the operators $\Gamma_{\mathcal{K}_G}$ and $\Gamma'_{\mathcal{K}_G}$)**.** *If $\mathcal{K}_G$ is a ground hybrid MKNF knowledge base and $S \subseteq S' \subseteq \mathsf{KA}(\mathcal{K}_G)$, then $\Gamma_{\mathcal{K}_G}(S') \subseteq \Gamma_{\mathcal{K}_G}(S)$ and $\Gamma'_{\mathcal{K}_G}(S') \subseteq \Gamma'_{\mathcal{K}_G}(S)$.*

Proof. Considering that the p-entailment operator $\models_p$ is monotonic, the proof equals the proof of Lemma 3 in [KAH11]. So, we simply refer to the proof given there. □

4.2 The Alternating Fixpoint Construction

On the basis of the two previously defined anti-monotonic operators $\Gamma_{\mathcal{K}_G}$ and $\Gamma'_{\mathcal{K}_G}$, it is now possible to define two monotonic, alternating sequences which compute the sets of modal **K**-atoms which are "true", i.e. those which are *inconsistent*, *suspiciously true* or *true*; and those *K*-atoms which are not "false", i.e. the ones which are *true* or *undefined*, respectively. Note that this is very similar to the fixpoint computation of the paraconsistent well-founded model in $WFSX_p$ where those objective literals which are obtained after an application of the operator $\Gamma\Gamma_s$ are "true" while those which are not obtained after an application of the operator Γ_s are "false". The following definition of the alternating fixpoint construction of the well-founded p-model of a ground hybrid MKNF knowledge base makes the two sequences that are implicitly present in the fixpoint computation defined by Alferes et al. for $WFSX_p$ [ADP95] explicit. The definition corresponds to the definition given by Knorr et al. [KAH11].

Definition 4.6 (Alternating fixpoint construction [KAH11])**.** Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base. We define two sequences $\mathbf{P}_i$ and $\mathbf{N}_i$ as follows.

$$\begin{aligned} \mathbf{P}_0 &= \emptyset & \mathbf{N}_0 &= \mathsf{KA}(\mathcal{K}_G) \\ \mathbf{P}_{n+1} &= \Gamma_{\mathcal{K}_G}(\mathbf{N}_n) & \mathbf{N}_{n+1} &= \Gamma'_{\mathcal{K}_G}(\mathbf{P}_n) \\ \mathbf{P}_\omega &= \bigcup \mathbf{P}_i & \mathbf{N}_\omega &= \bigcap \mathbf{N}_i \end{aligned}$$

It will always be clear from the context which is the underlying knowledge base of the two sequences, so that we can omit a reference to the respective knowledge base in the denotation of the two sequences. The sequence of $\mathbf{P}_i$ is a monotonically increasing sequence, while the sequence of $\mathbf{N}_i$ is a monotonically decreasing sequence, as stated by the following lemma adopted from [KAH11]. Consequently, the former sequence maximizes the set of modal **K**-atoms that are "true", while the latter minimizes the set of **K**-atoms which are not "false".

Lemma 4.7 (Monotonicity of the sequences of $\mathbf{P}_i$ and $\mathbf{N}_i$ [KAH11])**.** *Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base. Then $\mathbf{P}_\alpha \subseteq \mathbf{P}_\beta$ and $\mathbf{N}_\beta \subseteq \mathbf{N}_\alpha$ for all ordinals α, β with $\alpha \leq \beta \leq \omega$.*

Proof. The proof of this lemma is identical to the proof of Lemma 4 in [KAH11] and thus, omitted here. □

Moreover, by drawing on the Knaster-Tarski Theorem, Knorr et al. show that a least fixpoint is reached in case of the sequence of $\mathbf{P}_i$, and that a greatest fixpoint is reached in case of the sequence of $\mathbf{N}_i$, after $n < \omega$ iterations. Consequently, both sequences are finite. We recall the following result from [KAH11].

Proposition 4.8 (Least and greatest fixpoint [KAH11])**.** *Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base. Then $\mathbf{P}_\omega$ is the least fixpoint of the sequence of $\mathbf{P}_i$ and $\mathbf{N}_\omega$ is the greatest fixpoint of the sequence of $\mathbf{N}_i$.*

Proof. Again, the proof is identical to the proof given by Knorr et al. (cf. Proposition 5 in [KAH11]). □

Consequently, the two sets $\mathbf{P}_\omega$ and $\mathbf{N}_\omega$ exist for every ground hybrid MKNF knowledge base and are unique. Next, we define the well-founded pair of a ground hybrid MKNF knowledge base in terms of $\mathbf{P}_\omega$ and $\mathbf{N}_\omega$, which we will use to show the correspondence between the result obtained by the alternating fixpoint construction and the well-founded p-model of a ground hybrid MKNF knowledge base in Section 4.5. While Knorr et al. define a well-founded partition of a consistent hybrid MKNF knowledge base [KAH11], in our case the knowledge base does not have to be consistent and accordingly, **K**-atoms can be in the sets T_W and F_W simultaneously.

Definition 4.9 (The well-founded pair)**.** The *well-founded pair* of a ground hybrid MKNF knowledge base $\mathcal{K}_G$ is defined by:

$$(T_W, F_W) = (\mathbf{P}_\omega, \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_\omega)$$

In Section 4.5, we will show that the set T_W of the well-founded pair of a ground hybrid MKNF knoweldge base $\mathcal{K}_G$ contains all modal **K**-atoms which are *inconsistent*, *suspiciously true* or *true* in the well-founded p-model of $\mathcal{K}_G$, while the set F_W contains all **K**-atoms which are *inconsistent*, *suspiciously true*, *false*, or *classically false*. All atoms which are neither in T_W nor in F_W will be shown to be *undefined* in the well-founded p-model of $\mathcal{K}_G$.

On the basis of Definition 4.9, we can now state the procedural counterpart of Lemma 3.36, which has been expressed in terms of the model-theoretic definition of our semantics and characterizes the implementation of the Coherence Principle. Therefore, the following lemma reveals the enforcement of the Coherence Principle w.r.t. the well-founded

pair of a hybrid MKNF knowledge base. How the implementation of the Coherence Principle is achieved in the alternating fixpoint construction just presented is the topic of the next section.

Lemma 4.10 (The Coherence Principle w.r.t. the procedural characterization)**.** *Let* $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ *be a ground hybrid MKNF knowledge base and* $\mathbf{K}\,H$ *a* $\mathbf{K}$*-atom in* $\mathsf{KA}(\mathcal{K}_G)$*. Then* $\mathsf{OB}_{\mathcal{O},T_W} \models_p \neg H$ *implies that* $\mathbf{K}\,H \in F_W$*.*

Proof. The lemma follows directly from the construction of $\mathbf{N}_\omega$ by means of the operator $\Gamma'_{\mathcal{K}_G}$ and the definition of $T'_{\mathcal{K}_G,\mathcal{C}}$ in Definition 4.1, where those $\mathbf{K}$-atoms are removed from $T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega}(S)$ which are derivable from the objective knowledge of $\mathbf{P}_\omega$. □

4.3 Implementation of the Coherence Principle

In the three-valued version of the well-founded MKNF semantics introduced in [KAH11], the Coherence Principle is implemented by introducing a second variant of the MKNF transform, the so-called *MKNF-coherent transform*. Due to an additional condition in the definition of the MKNF-coherent transform, not only rules where a modal **not**-atom is *false* are removed from $\mathcal{P}_G$ in the transform, but also those rules for which the classical negation of the head can be derived from the objective knowledge of $\mathbf{P}_i$. As a result, in [KAH11], the corresponding $\mathbf{K}$-atoms cannot be derived by the operator $R_{\mathcal{K}_G}$ from the MKNF-coherent transform in the sequence of $\mathbf{N}_i$ and thus, the default negation of these atoms is implied by the derivation of the corresponding classically negated first-order atom (as postulated by the Coherence Principle).

In our case, we have to extend the implementation of the Coherence Principle to the ontology component because our goal is to also model inconsistent hybrid MKNF knowledge bases. As a consequence, it is possible that some first-order atom as well as its classical negation can be derived from the ontology at the same time. Therefore, it is necessary to not only cover the cases where some $\mathbf{K}$-atom may not be derived from the program by means of $R_{\mathcal{K}_G}$, but also those where a $\mathbf{K}$-atom is introduced in the sequence of $\mathbf{N}_i$ by means of the operator $D_{\mathcal{K}_G}$.

We achieve the desired behavior of the alternating fixpoint construction by moving the deletion of *classically false* $\mathbf{K}$-atoms into the operator $T'_{\mathcal{K}_G,\mathcal{C}}$, which represents a modification of the simpler operator $T_{\mathcal{K}_G}$. In this operator, after forming the union of the results of the operator $R_{\mathcal{K}_G}$ (yielding $\mathbf{K}$-atoms which are a consequence of the program) and of the operator $D_{\mathcal{K}_G}$ (yielding $\mathbf{K}$-atoms which are a consequence of the ontology), all $\mathbf{K}$-atoms $\mathbf{K}\,\xi$ where $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \models_p \neg\xi$ holds are removed from the result of $T'_{\mathcal{K}_G,\mathbf{P}_i}$. In this way, the MKNF-coherent transform can be omitted in the presentation of our approach, so that we obtain a unified implementation of the Coherence Principle not only for $\mathbf{K}$-atoms which are in the head of an MKNF rule, but also for those where the first-order atom under the $\mathbf{K}$-operator is a consequence of the ontology.

In the alternating fixpoint construction, the $T'_{\mathcal{K}_G,\mathcal{C}}$-operator is only used to compute the result of the operator $\Gamma'_{\mathcal{K}_G}$, which in turn is only used in the sequence of $\mathbf{N}_i$. As a result, if the classical negation of a first-order atom follows from the objective knowledge of $\mathbf{P}_i$, then the corresponding **K**-atom is contained in F_W as shown by Lemma 4.10. Note that this fact about the procedural formulation of our semantics in terms of the alternating fixpoint construction reflects the manifestation of the Coherence Principle in the model-based presentation of our semantics as revealed by Lemma 3.36 and can be viewed as its counterpart.

As the Coherence Principle states that classical negation must imply default negation, the extension of the Coherence Principle to the ontology is crucial for the evaluation of modal **not**-atoms, which is demonstrated in the subsequent example.

Example 4.11. Consider the following ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$\begin{array}{rcl} \top & \sqsubseteq & \neg Q \\ \top & \sqsubseteq & Q \\ \mathbf{K}\,P(a) & \leftarrow & \mathbf{not}\,Q(a) \end{array}$$

We start with $\mathbf{P}_0 = \emptyset$ and compute $\mathbf{N}_1 = \Gamma_{\mathcal{K}_G}(\mathbf{P}_0)$, $\mathbf{P}_2 = \Gamma_{\mathcal{K}_G}(\mathbf{N}_1)$, $\mathbf{N}_3 = \Gamma_{\mathcal{K}_G}(\mathbf{P}_1)$ and $\mathbf{P}_4 = \Gamma_{\mathcal{K}_G}(\mathbf{N}_3)$, i.e. construct the two sequences by using $\Gamma_{\mathcal{K}_G}$ in place of $\Gamma'_{\mathcal{K}_G}$, so that those **K**-atoms are not removed from $\mathbf{N}_1$ and $\mathbf{N}_3$ of which the classical negation can be derived from the objective knowledge of $\mathsf{OB}_{\mathcal{O},\mathbf{P}_0}$ and $\mathsf{OB}_{\mathcal{O},\mathbf{P}_2}$, respectively:

$$\begin{array}{rcl} \mathbf{N}_1 & = & \{\mathbf{K}\,Q(a), \mathbf{K}\,P(a)\} \\ \mathbf{P}_2 & = & \{\mathbf{K}\,Q(a)\} \\ \mathbf{N}_3 & = & \{\mathbf{K}\,Q(a)\} \\ \mathbf{P}_4 & = & \{\mathbf{K}\,Q(a)\} \end{array}$$

$\mathbf{P}_4$ is the least fixpoint of the sequence of $\mathbf{P}_i$, $\mathbf{N}_3$ is the greatest fixpoint of the sequence of $\mathbf{N}_i$ and the corresponding well-founded pair is $(\{\mathbf{K}\,Q(a)\}, \{\mathbf{K}\,P(a)\})$. Consequently, $\mathbf{K}\,Q(a)$ does not appear in the set F_W of the well-founded pair, even though the classical negation of $Q(a)$ is p-entailed by the ontology. This constitutes a violation of the Coherence Principle. Additionally, $\mathbf{K}\,P(a)$ is *false* in the p-model which can be obtained from this well-founded pair (as will be shown later).

Next, we compute the two sequences as defined in Definition 4.6, i.e. remove *classically false* **K**-atoms from the sequence of $\mathbf{N}_i$, and start again with $\mathbf{P}_0 = \emptyset$. Accordingly, we compute $\mathbf{N}_1 = \Gamma'_{\mathcal{K}_G}(\mathbf{P}_0)$, $\mathbf{P}_2 = \Gamma_{\mathcal{K}_G}(\mathbf{N}_1)$, $\mathbf{N}_3 = \Gamma'_{\mathcal{K}_G}(\mathbf{P}_2)$ and $\mathbf{P}_4 = \Gamma_{\mathcal{K}_G}(\mathbf{N}_3)$ now:

$$\begin{array}{rcl} \mathbf{N}_1 & = & \{\mathbf{K}\,P(a)\} \\ \mathbf{P}_2 & = & \{\mathbf{K}\,Q(a), \mathbf{K}\,P(a)\} \\ \mathbf{N}_3 & = & \{\} \end{array}$$

$$\mathbf{P}_4 \quad = \quad \{\mathbf{K}\,Q(a), \mathbf{K}\,P(a)\}$$

Again, $\mathbf{P}_4$ and $\mathbf{N}_3$ are the least and greatest fixpoints of the respective sequence. The corresponding well-founded pair is $(\{\mathbf{K}\,Q(a), \mathbf{K}\,P(a)\}, \{\mathbf{K}\,Q(a), \mathbf{K}\,P(a)\})$. Due to the deletion of those **K**-atoms $\mathbf{K}\,H$ where $\neg H$ follows from $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i}$ in the sequence of $\mathbf{N}_i$, $\mathbf{K}\,Q(a)$ is also in the set F_W of the well-founded pair now. In addition, as it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \not\models_p \neg P(a)$ for all $i \geq 0$, $\mathbf{K}\,P(a)$ is *suspiciously true* in the p-model which can be obtained from the well-founded pair, as intended. $\Diamond$

4.4 Soundness Result w.r.t. P-Models

After having introduced the procedural definition of our paraconsistent well-founded hybrid MKNF semantics in Section 4.2, we can now develop the first soundness result of the well-founded pair computed by the alternating fixpoint construction w.r.t. p-models as defined in Chapter 3. That is, we show that the well-founded pair of a hybrid MKNF knowledge base $\mathcal{K}_G$ induces a p-model of $\mathcal{K}_G$. As a first step, we show that the p-interpretation pair induced by the well-founded pair of a hybrid MKNF knowledge base p-satisfies the knowledge base by proving the following theorem. Accordingly, the following result implies that the p-interpretation pair induced by the well-founded pair fulfills condition (1) of Definition 3.10 for being a p-model of $\mathcal{K}_G$. The proof of the theorem follows the same ideas as the corresponding proof of Theorem 3 in [KAH11].

Theorem 4.12 (Correspondence to a p-interpretation pair p-satisfying $\mathcal{K}_G$)**.** *Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base and $(T_W, F_W) = (\mathbf{P}_\omega, \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_\omega)$ the well-founded pair of $\mathcal{K}_G$. Then $(I_P, I_N) \models_p \mathcal{K}_G$ where $I_P = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega}\}$ and $I_N = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega}\}$.*

Proof. By Definition 2.15, we know that $\mathcal{K}_G = \pi(\mathcal{O}) \wedge \pi(\mathcal{P}_G)$. Since $\pi(\mathcal{O})$ occurs in both $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega}$ and $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega}$, we have that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\pi(\mathcal{O})) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_P \cup I_N$. Therefore, we only have to consider the evaluation of $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\pi(\mathcal{P}_G))$. We begin by evaluating all modal atoms occurring in $\pi(\mathcal{P}_G)$. So, assume $\mathbf{K}\,H$ occurs in $\pi(\mathcal{P}_G)$.

At first, suppose that $\mathbf{K}\,H \in T_W$ and $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg H$. From $\mathbf{K}\,H \in T_W$, we know that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_P$ and since $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg H$, we obtain $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ is not possible for any $\mathcal{I} \in I_P$. Accordingly, we have that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{b}$ by Definition 3.6.

Next, suppose that $\mathbf{K}\,H \in T_W$, $\mathbf{K}\,H \in F_W$ and $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$. Because $\mathbf{K}\,H \in T_W$, we know that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_P$ and since it holds that $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$, we obtain $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for some $\mathcal{I} \in I_P$. Assume that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p H$. In this case, we have that $\mathbf{K}\,H \in \mathbf{N}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$ and since $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$. Hence, we conclude $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \not\models_p H$ since we have that $\mathbf{K}\,H \in F_W$. Thus, we derive that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{f}$ for some $\mathcal{I} \in I_N$. We conclude that

$(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{st}$ by Definition 3.6.

Now, suppose that $\mathbf{K}\,H \in T_W$ and $\mathbf{K}\,H \notin F_W$. Because $\mathbf{K}\,H \notin F_W$, we know that $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$ by Lemma 4.10. We obtain $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for some $\mathcal{I} \in I_P$ as before. Since we know that $\mathbf{K}\,H \notin F_W$, we have that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p H$ and consequently, $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_N$. From this, we conclude $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{t}$ by Definition 3.6.

Now, suppose that $\mathbf{K}\,H \notin T_W$, $\mathbf{K}\,H \in F_W$ and $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$, and assume that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p H$. In this case, $\mathbf{K}\,H \in \mathbf{P}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$. This is a contradiction to $\mathbf{K}H \notin T_W$ and we conclude $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p H$ since we have that $\mathbf{K}H \in F_W$. Thus, we derive $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{f}$ for some $\mathcal{I} \in I_P$. We have that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for some $\mathcal{I} \in I_P$ because it holds that $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$. Next, assume that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p H$. Then, we have that $\mathbf{K}\,H \in \mathbf{N}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$ and since $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$. Thus, we conclude $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \not\models_p H$. From this, we derive $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{f}$ for some $\mathcal{I} \in I_N$. Consequently, we know that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{f}$ by Definition 3.6.

Now, suppose that $\mathbf{K}\,H \notin T_W$ and $\mathsf{OB}_{\mathcal{O},T_W} \models_p \neg H$. As before, we can derive $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{f}$ for some $\mathcal{I} \in I_P$. However, in this case we do not have $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for any $\mathcal{I} \in I_P$ because it holds that $\mathsf{OB}_{\mathcal{O},T_W} \models_p \neg H$. Accordingly, we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{cf}$ by Definition 3.6.

Finally, suppose that $\mathbf{K}H \notin T_W$ and $\mathbf{K}H \notin F_W$. Like in the previous two cases, we obtain that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{f}$ for some $\mathcal{I} \in I_P$. Because $\mathbf{K}H \notin F_W$, we also have that $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$ by Lemma 4.10. Hence, we derive $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for some $\mathcal{I} \in I_P$. Because $\mathbf{K}\,H \notin F_W$, we know that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p H$. Consequently, we obtain that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_N$, and we conclude $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) = \mathbf{u}$ by Definition 3.6.

The cases for $\mathbf{not}\,H \in \pi(\mathcal{P})$ proceed similarly. Accordingly, in case $\mathbf{K}\,H \in T_W$ and $\mathsf{OB}_{\mathcal{O},T_W} \models_p \neg H$ hold, we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,H) = \mathbf{b}$; in case $\mathbf{K}\,H \in T_W$, $\mathbf{K}\,H \in F_W$ and $\mathsf{OB}_{\mathcal{O},T_W} \not\models_p \neg H$ hold, we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,H) = \mathbf{st}$; in case $\mathbf{K}\,H \in T_W$ and $\mathbf{K}\,H \notin F_W$ hold, we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,H) = \mathbf{f}$; in case $\mathbf{K}\,H \notin T_W$ and $\mathbf{K}\,H \in F_W$ hold, we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,H) = \mathbf{t}$; and otherwise we obtain $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,H) = \mathbf{u}$.

Now, consider $\pi(\mathcal{P}_G)$, which consists of a conjunction of implications, each representing a rule in $\mathcal{P}_G$. In order to prove that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$, we just have to make sure that none of the cases that map an implication to a non-designated truth value (i.e. $\mathbf{st}$, $\mathbf{f}$, $\mathbf{cf}$ or $\mathbf{u}$) occurs. According to Table 3.1, these cases arise when either the body of a rule is assigned a truth value from the set $\{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$ and the rule-head is evaluated to one of the truth values from the set $\{\mathbf{f}, \mathbf{cf}, \mathbf{u}\}$, or the body of a rule is *true* or *undefined* and the head is *suspiciously true* or *false*.

We start by proving that the firstly mentioned cases are not possible. If the body of a rule with head $\mathbf{K}H$ is assigned one of the truth values $\mathbf{b}$, $\mathbf{st}$ and $\mathbf{t}$ by the p-interpretation pair (I_P, I_N), this means that for all modal **K**-atoms $\mathbf{K}\,A_i$ occurring in the body of the rule we have that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,A_i) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$, and for all modal **not**-atoms

$\mathbf{not}\, B_j$ which occur in the body we have that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\, B_j) \in \{\mathbf{b}, \mathbf{st}, \mathbf{f}, \mathbf{cf}\}$. Consequently, we obtain that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p A_i$ and $\mathbf{K}\, B_j \notin \mathbf{N}_\omega$ for all i and j. (Note, that the latter is true because, if some $\mathbf{K}\, B_j$ is *inconsistent* or *classically false*, then $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg B_j$ and hence, it is not in $\mathbf{N}_\omega$ according to Lemma 4.10.) Thus, the rule $\mathbf{K}\, H \leftarrow \mathbf{K}\, A_1, \ldots, \mathbf{K}\, A_n$ occurs in the MKNF transform $\mathcal{K}_G/\mathbf{N}_\omega$ due to Definition 4.2. As all A_i are in $\mathbf{P}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$, $\mathbf{K}\, H$ is in $\mathbf{P}_\omega$ as well, by means of $\mathcal{R}_{\mathcal{K}_G}$. But, this means that the head of the rule cannot be assigned any of the truth values $\mathbf{f}$, $\mathbf{cf}$ and $\mathbf{u}$ by (I_P, I_N) since we can infer that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I} \in I_P$. Hence, these cases cannot occur.

Now, assume that the body of a rule with head $\mathbf{K}\, H$ is either *true* or *undefined* in the p-interpretation pair (I_P, I_N). In this case, for all modal $\mathbf{K}$-atoms $\mathbf{K}\, A_i$ occurring in the rule-body, we have that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\, A_i) \in \{\mathbf{t}, \mathbf{u}\}$, and for all modal $\mathbf{not}$-atoms $\mathbf{not}\, B_j$ occurring in the rule-body we have that $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\, B_j) \in \{\mathbf{f}, \mathbf{cf}, \mathbf{u}\}$. So, we know that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p A_i$ for all i, and $\mathbf{K}\, B_j \notin \mathbf{P}_\omega$ for all j. From the latter, we derive that the rule $\mathbf{K}\, H \leftarrow \mathbf{K}\, A_1, \ldots, \mathbf{K}\, A_n$ occurs in the MKNF transform $\mathcal{K}_G/\mathbf{P}_\omega$ due to Definition 4.2. If it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg H$, then only $\mathbf{b}$ and $\mathbf{cf}$ are possible for $\mathbf{K}\, H$. So, assume $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg H$. Like before, since all A_i are in $\mathbf{N}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$, $\mathbf{K}H$ is in $\mathbf{N}_\omega$ as well, by means of $\mathcal{R}_{\mathcal{K}_G}$ and since $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg H$. So, we have that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p H$ and hence, that $(\mathcal{I}, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(H) = \mathbf{t}$ for each $\mathcal{I} \in I_N$. Consequently, the head of the rule can neither be assigned the truth value $\mathbf{st}$ nor the truth value $\mathbf{f}$ in (I_P, I_N) and the latter cases cannot occur either. Thus, $(*, \langle I_P, I_N\rangle, \langle I_P, I_N\rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$ holds, which proves that the p-interpretation pair (I_P, I_N) in fact p-satisfies the knowledge base $\mathcal{K}_G$. □

By proving the next theorem we show that the p-interpretation pair induced by the well-founded pair of a hybrid MKNF knowledge base is in fact a p-model of the knowledge base that fulfills all three conditions of Definition 3.10.

Theorem 4.13 (Soundness w.r.t. p-models)**.** *Let $\mathcal{K}_G$ be an MKNF-consistent ground hybrid MKNF knowledge base and $(T_W, F_W) = (\mathbf{P}_\omega, \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_\omega)$ the well-founded pair of $\mathcal{K}_G$. Then (I_P, I_N) is a p-model of $\mathcal{K}_G$, where $I_P = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega}\}$ and $I_N = \{\mathcal{I} \mid \mathcal{I} \models_p \mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega}\}$.*

Proof. We know from Theorem 4.12 that (I_P, I_N) p-satisfies $\mathcal{K}_G$. Hence, (I_P, I_N) fulfills condition (1) for being a p-model of $\mathcal{K}_G$ as specified in Definition 3.10. Therefore, in order to prove that (I_P, I_N) is a p-model of $\mathcal{K}_G$, we only have to prove that it also fulfills conditions (2) and (3) of that definition.

Regarding condition (2) of Definition 3.10, we have to prove that there is no p-interpretation pair (M', N') with $I_P \subseteq M'$ and $I_N \subseteq N'$ where at least one of the inclusions is proper and $(\mathcal{I}', \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. First, assume that there is a p-interpretation pair (M', N') where $I_P \subset M'$ and $I_N \subseteq N'$, i.e. at least the first inclusion is proper, and $(\mathcal{I}', \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. We show that if some p-interpretation $\mathcal{I}'$ is in M', then it also has to be in I_P and thus, derive a contradiction to our assumption that $I_P \subset M'$. First, we consider the case where the program component $\mathcal{P}_G$ is empty, i.e. where $\mathcal{K}_G = \pi(\mathcal{O})$. In

this case, we obtain that $(\mathcal{I}', \mathcal{M}, \mathcal{N})(\pi(\mathcal{O})) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M'$ by our assumption that $(\mathcal{I}', \langle M', N' \rangle, \langle I_P, I_N \rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M' \cup N'$. Hence, it holds that $\mathcal{I}' \models_p \pi(\mathcal{O})$. However, then $\mathcal{I}'$ must also be in I_P because $I_P = \{\mathcal{I} \mid \mathcal{I} \models_p \pi(\mathcal{O})\}$ in this case. This is a contradiction to our assumption that $I_P \subset M'$.

Now, we consider a ground hybrid MKNF knowledge base $\mathcal{K}_G = \pi(\mathcal{O}) \wedge \pi(\mathcal{P}_G)$ with $\mathcal{P}_G \neq \emptyset$. Since $I_P \subseteq M'$, we can define a set of modal **K**-atoms R_M such that $M' = \{\mathcal{I}' \mid \mathcal{I}' \models_p \mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega \setminus R_M}\}$ where $R_M \subseteq \mathbf{P}_\omega$ and there is no ξ in $\{\xi \mid \mathbf{K}\,\xi \in R_M\}$ such that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega \setminus R_M} \models_p \xi$. The latter restriction is necessary since all ξ in $\{\xi \mid \mathbf{K}\,\xi \in R_M\}$ might actually be derivable from the objective knowledge of $\mathbf{P}_\omega \setminus R_M$ in which case we would directly obtain that $M' = I_P$. We prove that R_M is empty. In this way, we can show again that if some p-interpretation $\mathcal{I}'$ is in M', then it has to be in I_P as well, and therefore, also in this case we are able to derive a contradiction to our assumption that $I_P \subset M'$.

As every modal **K**-atom $\mathbf{K}\,H$ which is in R_M is also in $\mathbf{P}_\omega$, we know that $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow j$ for some j. We show by an induction on j that if some $\mathbf{K}\,H$ is in R_M and we have that $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow j$, then we can derive a contradiction and it follows that $\mathbf{K}\,H$ cannot be in R_M. Thus, R_M has to be empty.

The base case holds straightforwardly, since $T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow 0$ is empty. Now, assume that the claim holds for all $j \leq m$, and consider $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow m+1$. If $\mathbf{K}\,H$ already occurs in $T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow m$, then the claim holds directly by the induction hypothesis. Otherwise, there are two cases to consider. Either there is a positive rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots \mathbf{K}\,A_n$ in $\mathcal{K}_G/\mathbf{N}_\omega$ with $\mathbf{K}\,A_i \in T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow m$, or $\mathbf{K}\,H$ is the consequence of $D_{\mathcal{K}_G/\mathbf{N}_\omega}(T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow m)$.

In the first case, by the induction hypothesis, if some $\mathbf{K}\,A_i$ is in R_M, then it follows that $\mathbf{K}\,H$ cannot be in R_M. So, assume that $\mathbf{K}\,A_i \notin R_M$ for all $\mathbf{K}\,A_i$. This means that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega \setminus R_M} \models_p A_i$ for all $\mathbf{K}\,A_i$ and hence, $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\,A_i) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$ for all $\mathbf{K}\,A_i$. Additionally, there is a rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$ in $\mathcal{K}_G$, and since the positive version of this rule occurs in $\mathcal{K}_G/\mathbf{N}_\omega$, no $\mathbf{K}\,B_j$ occurs in $\mathbf{N}_\omega$. If $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg B_j$ for some $\mathbf{not}\,B_j$, then $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\mathbf{not}\,B_j) \in \{\mathbf{b}, \mathbf{t}\}$ due to Definition 3.6. (Note that all $\mathbf{not}\,B_j$ are still interpreted w.r.t. (I_P, I_N).) So, consider all $\mathbf{not}\,B_k$ such that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg B_k$ and assume that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \models_p B_k$ for some $\mathbf{not}\,B_k$. In this case, we have that $\mathbf{K}\,B_k \in \mathbf{N}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$ and since $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg B_k$. Hence, we conclude that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega} \not\models_p B_k$ for all $\mathbf{not}\,B_k$. Consequently, $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\mathbf{not}\,B_j) \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}$ for all $\mathbf{not}\,B_j$. Now, if $\mathbf{K}\,H$ is in R_M, then $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega \setminus R_M} \not\models_p H$ and hence, H is evaluated to $\mathbf{f}$ by some p-interpretation in M'. By Definition 3.6, we infer that $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\,H) \in \{\mathbf{f}, \mathbf{cf}, \mathbf{u}\}$. This means that we have a rule with *inconsistent*, *suspiciously true* or *true* body and *false*, *classically false* or *undefined* head in this case, and we derive that $(\mathcal{I}', \langle M', N' \rangle, \langle I_P, I_N \rangle)(\pi(\mathcal{P}_G)) \notin \{\mathbf{b}, \mathbf{t}\}$ according to the definition of the implication operator in Table 3.1. So, we derive a contradiction to our assumption that $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$ and thus, $\mathbf{K}\,H$ actually cannot be in R_M.

In the second case, $\mathsf{OB}_{\mathcal{O},S} \models H$ with $S = T_{\mathcal{K}_G/\mathbf{N}_\omega} \uparrow m$ holds. Since we have that $(\mathcal{I}', \langle M', N' \rangle, \langle I_P, I_N \rangle)(\pi(\mathcal{O})) \in \{\mathbf{b}, \mathbf{t}\}$ for each $\mathcal{I}' \in M'$, we also know that $\mathcal{I}' \models_p \pi(\mathcal{O})$ for each $\mathcal{I}' \in M'$. We assume that $\mathbf{K}\,A_i \notin R_M$ for each $\mathbf{K}\,A_i$ in S since, if some $\mathbf{K}\,A_i$ is in R_M,

then $(*, \langle M', N'\rangle, \langle I_P, I_N\rangle)(\pi(\mathcal{P}_G)) \notin \{\mathbf{b}, \mathbf{t}\}$ follows again by the induction hypothesis. So, we also know that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega\setminus R_M} \models_p H$ since $A \subseteq \mathbf{P}_\omega$. According to the restriction of the set R_M, we again derive a contradiction to $\mathbf{K}\,H$ being in R_M. This proves that R_M can only be empty. Consequently, if some p-interpretation $\mathcal{I}'$ is in M', then it has to be in I_P as well, which is a contradiction to our assumption that $I_P \subset M'$.

Now, assume that there is a p-interpretation pair (M', N') where $I_P \subseteq M'$ and $I_N \subset N'$, i.e. at least the second inclusion is proper, and that $(\mathcal{I}', \langle M', N'\rangle, \langle M, N\rangle)(\mathcal{K}_G) \in \{\mathbf{b}, \mathbf{t}\}$ holds for each $\mathcal{I}' \in M' \cup N'$. Similar to the first part of the proof, we show that if some p-interpretation $\mathcal{I}'$ is in N', then it also has to be in I_N and thus, derive a contradiction to our assumption that $I_N \subset N'$. The proof for the case where the program component is empty proceeds analogously to the proof above. So, assume that the program component is not empty. In this case, we can define a set of modal $\mathbf{K}$-atoms R_N such that $N' = \{\mathcal{I}' \mid \mathcal{I}' \models_p \mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega\setminus R_N}\}$ where $R_N \subseteq \mathbf{N}_\omega$ and there is no ξ in $\{\xi \mid \mathbf{K}\,\xi \in R_N\}$ such that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega\setminus R_N} \models_p \xi$. As before, we also have that $M' = \{\mathcal{I}' \mid \mathcal{I}' \models_p \mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega\setminus R_M}\}$ where $R_M \subseteq \mathbf{P}_\omega$. R_M may be empty in this case since $M' = I_P$ is allowed to hold. Again, we show that R_N can only be empty, from which the contradiction follows.

We know that if $\mathbf{K}H \in R_N$, then $\mathbf{K}H \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow j$ for some j holds. As before, we show by an induction on j that if some $\mathbf{K}H$ is in R_N and we have that $\mathbf{K}H \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow j$ for some j, then we can derive a contradiction and it follows that $\mathbf{K}H$ cannot be in R_N. Therefore, R_N has to be empty.

The base case holds as before, because $T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow 0$ is empty. So, assume that the claim holds for all $j \leq m$, and consider $\mathbf{K}H \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m+1$. If $\mathbf{K}H$ already occurs in $T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m$, then the claim holds directly by the induction hypothesis. Otherwise, there are again two possible cases. Either there is a positive rule $\mathbf{K}H \leftarrow \mathbf{K}A_1, \ldots \mathbf{K}A_n$ in $\mathcal{K}_G/\mathbf{P}_\omega$ with $\mathbf{K}A_i \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m$, or $\mathbf{K}H$ is the consequence of $D_{\mathcal{K}_G/\mathbf{P}_\omega}(T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m)$.

In the first case, by the induction hypothesis, if some $\mathbf{K}A_i$ is in R_N, then it follows that $\mathbf{K}H$ cannot be in R_N. So, assume that no $\mathbf{K}A_i$ is in R_N. Consequently, $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega\setminus R_N} \models_p A_i$ holds for all $\mathbf{K}\,A_i$. Since $\mathbf{K}\,A_i \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m$ holds for all $\mathbf{K}\,A_i$, we also know that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg H$ due to Definition 4.1. So, we obtain that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega\setminus R_M} \not\models_p \neg H$ as well. Therefore, we know that for each A_i there is a p-interpretation in M' such that A_i is evaluated to *true* in that p-interpretation. We derive by Definition 3.6 that $(*, \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,A_i) \in \{\mathbf{t}, \mathbf{u}\}$ holds for all $\mathbf{K}\,A_i$. Besides, there is a rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$ in $\mathcal{K}_G$, and since the positive version of this rule occurs in $\mathcal{K}_G/\mathbf{P}_\omega$, no $\mathbf{K}B_j$ occurs in $\mathbf{P}_\omega$. Now, assume that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p B_k$ for some $\mathbf{not}\,B_k$. In this case, we have that $\mathbf{K}\,B_k \in \mathbf{P}_\omega$ by means of $\mathcal{D}_{\mathcal{K}_G}$. Hence, we can conclude that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p B_k$ for all $\mathbf{not}\,B_k$. As a result, we obtain that $(\mathcal{I}', \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathbf{not}\,B_j) \in \{\mathbf{t}, \mathbf{u}\}$ for all $\mathbf{not}\,B_j$ by Definition 3.6. Now, if $\mathbf{K}\,H$ is in R_N, then $\mathsf{OB}_{\mathcal{O},\mathbf{N}_\omega\setminus R_N} \not\models_p H$ and hence, H is evaluated to $\mathbf{f}$ by some p-interpretation in N'. By Definition 3.6, we infer that $(\mathcal{I}', \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) \in \{\mathbf{st}, \mathbf{f}\}$. Note that since we have that $\mathbf{K}\,H \in T'_{\mathcal{K}_G/\mathbf{P}_\omega,\mathbf{P}_\omega} \uparrow m+1$ we know that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg H$ and thus, $(\mathcal{I}', \langle M', N'\rangle, \langle I_P, I_N\rangle)(\mathbf{K}\,H) \in \{\mathbf{b}, \mathbf{cf}\}$ is

not possible. This means that we have a rule with *true* or *undefined* body and *suspiciously true* or *false* head in this case, and we derive that $(\mathcal{I}', \langle M', N' \rangle, \langle I_P, I_N \rangle)(\pi(\mathcal{P}_G)) \notin \{\mathbf{b}, \mathbf{t}\}$ according to the definition of the implication operator in Table 3.1. So, we derive a contradiction to our assumption that $(*, \langle M', N' \rangle, \langle I_P, I_N \rangle)(\pi(\mathcal{P}_G)) \in \{\mathbf{b}, \mathbf{t}\}$ and thus, $\mathbf{K}\, H$ actually cannot be in R_N.

The second case proceeds analogously to the second case of the previous induction. This proves that R_N can only be empty. Consequently, if some p-interpretation $\mathcal{I}'$ is in N', then it has to be in I_N as well, which is a contradiction to our assumption that $I_N \subset N'$.

Finally, consider condition (3) of Definition 3.10. In this case, we have to prove that it holds for all $\mathbf{K}\, \xi \in \mathsf{KA}(\mathcal{K}_G)$ that $(*, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\, \xi) \in \{\mathbf{b}, \mathbf{cf}\}$ if and only if $\{\pi(\mathcal{O})\} \cup \{\xi' \mid \mathbf{K}\, \xi' \in \mathsf{KA}(\mathcal{K}_G)$ and $(*, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\, \xi') \in \{\mathbf{b}, \mathbf{st}, \mathbf{t}\}\} \models_p \neg\xi$, i.e. if and only if $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg\xi$. First, assume that $(*, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\, \xi) \in \{\mathbf{b}, \mathbf{cf}\}$, but $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg\xi$. Since $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \not\models_p \neg\xi$, we know there is a p-interpretation $\mathcal{I}$ in I_P s.t. $(\mathcal{I}, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\xi) = \mathbf{t}$. Hence, due to Definition 3.6, $(*, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\mathbf{K}\, \xi) \notin \{\mathbf{b}, \mathbf{cf}\}$, which is a contradiction to our assumption. Consequently the first direction of condition (3) from left to right holds for (I_P, I_N). Now, assume $\mathsf{OB}_{\mathcal{O},\mathbf{P}_\omega} \models_p \neg\xi$. It follows that there cannot be a p-interpretation $\mathcal{I}$ in I_P s.t. $(\mathcal{I}, \langle I_P, I_N \rangle, \langle I_P, I_N \rangle)(\xi) = \mathbf{t}$. Therefore, only the truth values $\mathbf{b}$ and $\mathbf{cf}$ are possible for $\mathbf{K}\, \xi$ under the p-interpretation pair (I_P, I_N) according to Definition 3.6. Thus, also the second direction from right to left of the biconditional is fulfilled by (I_P, I_N). Accordingly, the p-interpretation pair (I_P, I_N) fulfills condition (3) of Definition 3.10. This proves that (I_P, I_N) is in fact a p-model of $\mathcal{K}_G$. □

4.5 Soundness and Completeness Result

In this section, we develop the main soundness and completeness result for the well-founded pair of a hybrid MKNF knowledge base $\mathcal{K}_G$ w.r.t. the well-founded p-model of $\mathcal{K}_G$. By showing that the unique result of the alternating fixpoint construction introduced in Section 4.2 corresponds exactly to the well-founded p-model of a knowledge base in case the DL used for expressing the ontology component admits a 4-model, we obtain that the well-founded p-model always exists and that it is unique.

Before we can prove the main result of this section, we have to establish an intermediate result in Proposition 4.18 and introduce the definitions needed to prove this proposition next. Knorr et al. define the following notion of a **K**-atom depending on a set of **K**-atom via the objective knowledge to be able to better characterize derivations performed by the operator $D_{\mathcal{K}_G}$ in the immediate consequence operator [KAH11]. We first present our adaptation of the definition provided by Knorr et al. and extend it subsequently in order to also capture derivations performed by the operator $R_{\mathcal{K}_G}$.

Definition 4.14 (Dependence via the objective knowledge)**.** Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base, $\mathbf{K}\, H$ a modal **K**-atom with $\mathbf{K}\, H \in \mathsf{KA}(\mathcal{K}_G)$, and S a (possibly

empty) set of modal **K**-atoms with $S \subseteq \mathsf{KA}(\mathcal{K}_G)$. We say that $\mathbf{K}\,H$ *depends* on S if and only if

(i) $\mathsf{OB}_{\mathcal{O},S} \models_p H$ and

(ii) there is no S' with $S' \subset S$ such that $\mathsf{OB}_{\mathcal{O},S'} \models_p H$.

The notion of a **K**-atom depending on a set of **K**-atoms introduced in Definition 4.14 is used in the following definition of *hybrid dependence*, which applies to a positive ground hybrid MKNF knowledge base and extends the concept of dependence by also taking the program component into account.

Definition 4.15 (Hybrid dependence)**.** Let $\mathcal{K}_G$ be a positive ground hybrid MKNF knowledge base, $\mathbf{K}\,H$ a modal **K**-atom with $\mathbf{K}\,H \in \mathsf{KA}(\mathcal{K}_G)$, and S a (possibly empty) set of modal **K**-atoms with $S \subseteq \mathsf{KA}(\mathcal{K}_G)$. We say that $\mathbf{K}\,H$ *hybrid-depends* on a set S if and only if

(i) $\mathbf{K}\,H$ is in S,

(ii) for every $\mathbf{K}\,H' \in S$

- there is a positive rule $\mathbf{K}\,H' \leftarrow \mathbf{K}\,A_1, ..., \mathbf{K}\,A_n$ in P_G such that all $\mathbf{K}\,A_i$ are in S, or
- there is a set $\{A_1, ..., A_n\}$ on which $\mathbf{K}\,H'$ depends (not containing H' itself)[1] and all $\mathbf{K}\,A_i$ are in S, and

(iii) there is no S' with $S' \subset S$ such that S' fulfills the conditions (i) and (ii).

We call S a *hybrid dependence set* of $\mathbf{K}\,H$.

The following example illustrates the notion of hybrid dependence.

Example 4.16 (Hybrid dependence sets)**.** Consider the following positive ground hybrid MKNF knowledge base $\mathcal{K}_G$.

$$
\begin{array}{rcl}
Q & \sqsubseteq & P \\
\mathbf{K}\,Q(a) & \leftarrow & \\
\mathbf{K}\,R(a) & \leftarrow & \\
\mathbf{K}\,S(a) & \leftarrow & \\
\mathbf{K}\,P(a) & \leftarrow & \mathbf{K}\,R(a), \mathbf{K}\,Q(a) \\
\mathbf{K}\,P(a) & \leftarrow & \mathbf{K}\,S(a) \\
\mathbf{K}\,P(a) & \leftarrow & \mathbf{K}\,T(a)
\end{array}
$$

Since $\mathbf{K}\,Q(a)$, $\mathbf{K}\,R(a)$ and $\mathbf{K}\,S(a)$ are contained as facts in the program, they only hybrid-depend on the set containing themselves, respectively. The **K**-atom $\mathbf{K}\,P(a)$ depends on

[1] We exclude this case since every **K**-atom vacuously depends on itself.

the set $\{\mathbf{K}\,Q(a)\}$ due to the ontology axiom. As $\mathbf{K}\,Q(a)$ only hybrid-depends on the set containing itself, $\{\mathbf{K}\,P(a), \mathbf{K}\,Q(a)\}$ is a hybrid dependence set of $\mathbf{K}\,P(a)$. In addition, there are three rules with head $\mathbf{K}\,P(a)$ in the program component. However, no hybrid dependence set can be obtained from the first rule because the elements of the hybrid dependence set $\{\mathbf{K}\,P(a), \mathbf{K}\,Q(a)\}$ are contained in the set $\{\mathbf{K}\,P(a), \mathbf{K}\,Q(a), \mathbf{K}\,R(a)\}$, so that the minimality condition (iii) of Definition 4.15 would be violated. Regarding the second rule with head $\mathbf{K}\,P(a)$, another hybrid dependence set $\{\mathbf{K}\,P(a), \mathbf{K}\,S(a)\}$ can be obtained since $\mathbf{K}\,S(a)$ hybrid-depends on the set $\{\mathbf{K}\,S(a)\}$ and there is no set on which $\mathbf{K}\,P(a)$ hybrid-depends that is a subset of the set $\{\mathbf{K}\,P(a), \mathbf{K}\,S(a)\}$. Finally, the last rule does not yield another hybrid dependence set of $\mathbf{K}\,P(a)$ because there is neither a rule with head $\mathbf{K}\,T(a)$ in the program nor a set of **K**-atoms on which $\mathbf{K}\,T(a)$ depends according to Definition 4.14. Note that if we would not exclude those dependence sets in condition (ii) of Definition 4.15 which contain the atom itself, there would be another hybrid dependence set of $\mathbf{K}\,P(a)$, namely $\{\mathbf{K}\,P(a), \mathbf{K}\,T(a)\}$. ◊

Intuitively, a modal **K**-atom is derivable from a hybrid MKNF knowledge base if and only if all **K**-atoms in at least one of its hybrid dependence sets are derivable at the same time. This intuition is formalized in the following lemma.

Lemma 4.17. *Let $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ be a ground hybrid MKNF knowledge base, $\mathbf{K}\,H$ a **K**-atom in $\mathsf{KA}(\mathcal{K}_G)$ and S a subset of $\mathsf{KA}(\mathcal{K}_G)$. Then it holds that $\mathbf{K}\,H \notin \Gamma'_{\mathcal{K}_G}(S)$ if and only if for every set $S' \subseteq \mathsf{KA}(\mathcal{K}_G)$ on which $\mathbf{K}\,H$ hybrid-depends there is some $\mathbf{K}\,H' \in S'$ such that $\mathbf{K}\,H' \notin \Gamma'_{\mathcal{K}_G}(S)$.*

Proof. The first direction of the statement, from left to right, can easily be proven by showing the contrapositive. Assume that there is a set $S' \subseteq \mathsf{KA}(\mathcal{K}_G)$ on which $\mathbf{K}\,H$ hybrid-depends such that for all $\mathbf{K}\,H' \in S'$ it is the case that $\mathbf{K}\,H' \in \Gamma'_{\mathcal{K}_G}(S)$. By Definition 4.15 and the definition of the operator Γ' it follows that $\mathbf{K}\,H$ has to be in $\Gamma'_{\mathcal{K}_G}(S)$ as well. The other direction, from right to left, also follows directly by the definition of the operator Γ' since $\mathbf{K}\,H$ can only be derived by $T'_{\mathcal{K}_G/S,S} \uparrow i$ for some i if all elements in some set S' are derived by $T'_{\mathcal{K}_G/S,S} \uparrow m$ for some $m < i$. The previous results from the definition of the operator $T'_{\mathcal{K}_G,\mathcal{C}}$ in Definition 4.1. On the one hand, the operator $D_{\mathcal{K}_G}$ in this definition derives all modal **K**-atoms for which there is a rule in $\mathcal{P}_G$ such that all **K**-atoms in the body have already been derived, corresponding to the first item of condition (ii) in the definition of hybrid dependence sets. On the other hand, the operator $D_{\mathcal{K}_G}$ yields all those modal **K**-atoms $\mathbf{K}\,H$ such that there is a set S on which $\mathbf{K}\,H$ depends and all elements of S have been derived before, corresponding to the second item of condition (ii) in Definition 4.15. □

When we introduced well-founded p-models in Section 3.9, we mentioned that a modal **K**-atom is *inconsistent*, *suspiciously true* or *true* in the well-founded p-model of a hybrid MKNF knowledge base $\mathcal{K}_G$ if and only if this holds for every p-model of $\mathcal{K}_G$; and that a **K**-atom is *inconsistent*, *suspiciously true*, *false* or *classically false* in the well-founded p-model of $\mathcal{K}_G$ if and only if this is the case for all p-models of $\mathcal{K}_G$ as well. The next result

transfers the mentioned property to the result of the alternating fixpoint computation and will be needed for showing that the well-founded pair we compute not only corresponds to a p-model, but to the unique well-founded p-model of the respective hybrid MKNF knowledge base. The idea behind the proof of the following definition is identical to the one of the proof for Proposition 7 in [KAH11], and its structure is reflected here. However, due to the three additional truth values we employ, the details of the proof are more intricate.

Proposition 4.18 (Characterization of the well-founded pair)**.** *Given a ground hybrid MKNF knowledge base $\mathcal{K}_G$ and the pair $(T, F) = (\mathbf{P}_\omega, \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_\omega)$,*

- $\mathbf{K}\,H \in T$ *implies that* $\mathbf{K}\,H$ *is inconsistent, suspiciously true or true in every p-model of* $\mathcal{K}_G$*, and*
- $\mathbf{K}\,H \in F$ *implies that* $\mathbf{K}\,H$ *is inconsistent, suspiciously true, false or classically false in every p-model of* $\mathcal{K}_G$*.*

Proof. Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base and let (T, F) be the pair $(\mathbf{P}_\omega, \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_\omega)$. According to Proposition 4.8, we can prove this proposition by showing that, for all i,

(1) $\mathbf{K}\,H \in \mathbf{P}_i$ implies that $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$, and

(2) $\mathbf{K}\,H \notin \mathbf{N}_i$ implies that $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$.

We prove the cases (1) and (2) simultaneously by an induction on i. For both cases, the base case $i = 0$ holds trivially since $\mathbf{P}_0$ is empty and $\mathbf{N}_0$ is equal to $\mathsf{KA}(\mathcal{K}_G)$.

(i) Assume that the cases (1) and (2) hold for all $i \leq n$. We consider $i = n + 1$ for both cases, namely $\mathbf{K}\,H \in \mathbf{P}_{n+1}$ and $\mathbf{K}\,H \notin \mathbf{N}_{n+1}$.

Case (1): First, let $\mathbf{K}\,H \in \mathbf{P}_{n+1}$. If $\mathbf{K}\,H$ already occurs in $\mathbf{P}_n$, then $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true* or *true* in every p-model (M, N) of $\mathcal{K}_G$, by the induction hypothesis (i). Otherwise, $\mathbf{K}\,H \in \Gamma_{\mathcal{K}_G}(\mathbf{N}_n)$, i.e. $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow \omega$ but $\mathbf{K}\,H \notin \mathbf{P}_n$. Since $\mathbf{K}\,H$ is introduced by $T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow \omega$, we know that $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow j$ for some j, and we prove by induction on j that $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true* or *true* in every p-model (M, N) of $\mathcal{K}_G$.

The base case obviously holds, since $T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow 0$ is empty.

(ii) Assume that the claim holds for all $j \leq m$, and consider $\mathbf{K}\,H \in T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow m + 1$.

If $\mathbf{K}\,H$ already occurs in $T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow m$, then the claim holds automatically by the induction hypothesis (ii). Otherwise, there are two cases to consider. Either there is a positive rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots \mathbf{K}\,A_n$ in $\mathcal{K}_G/\mathbf{N}_n$ with $\mathbf{K}\,A_i \in T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow m$, or $\mathbf{K}\,H$ is the consequence of $D_{\mathcal{K}_G/\mathbf{N}_n}(T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow m)$. In the first case, by the induction hypothesis (ii), all $\mathbf{K}\,A_i$ are *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$. Additionally,

there is a rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$ in $\mathcal{K}_G$, and since the positive version of this rule occurs in $\mathcal{K}_G/\mathbf{N}_n$, no $\mathbf{K}\,B_j$ occurs in $\mathbf{N}_n$, and thus (by the induction hypothesis (i)), all $\mathbf{K}\,B_j$ are *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$. Thus, all $\mathbf{not}\,B_j$ are *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$. Consequently, $\mathbf{K}\,H$ has to be *inconsistent*, *suspiciously true* or *true* in every p-model (M, N) of $\mathcal{K}_G$.

In the second case, $\mathsf{OB}_{\mathcal{O},S} \models_p H$ holds, where $S = T_{\mathcal{K}_G/\mathbf{N}_n} \uparrow m$. Because we have that $(M, N) \models_p \pi(\mathcal{O})$, we know that $\mathcal{I} \models_p \pi(\mathcal{O})$ for each $\mathcal{I} \in M$ of every p-model (M, N) of $\mathcal{K}_G$. Since we also know that all modal **K**-atoms $\mathbf{K}\,A_i \in S$ are *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$ (by the induction hypothesis (ii)), we can conclude that $\mathcal{I} \models_p A_i$ for each $\mathcal{I} \in M$ of every p-model (M, N) of $\mathcal{K}_G$, by the definition of the modal **K**-operator in Definition 3.6. From this, we derive that also $\mathcal{I} \models_p H$ holds for each $\mathcal{I} \in M$ and every p-model (M, N) of $\mathcal{K}_G$ and hence, $\mathbf{K}\,H$ has to be *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$, again by Definition 3.6.

Case (2): Next, consider all $\mathbf{K}\,H \in \mathsf{KA}(\mathcal{K}_G)$ s.t. $\mathbf{K}\,H \notin \mathbf{N}_{n+1}$. Let U be the set of all such $\mathbf{K}\,H$, i.e. the set of all $\mathbf{K}\,H \in \mathsf{KA}(\mathcal{K}_G)$ s.t. $\mathbf{K}\,H \notin \Gamma'_{\mathcal{K}_G}(\mathbf{P}_n)$. Now, from the definition of $\Gamma'_{\mathcal{K}_G}(\mathbf{P}_n)$ in Definition 4.4 and 4.1, we can conclude that for each modal **K**-atom $\mathbf{K}\,H$ in U, it is the case that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \models_p \neg H$ holds or the following conditions[2] are fulfilled:

(Ui) For each rule $\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots, \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$ in $\mathcal{P}_G$ at least one of the following holds:

  (Uia) Some modal **K**-atom $\mathbf{K}\,A_i$ appears in $U \cup \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_n$ or

  (Uib) some modal **K**-atom $\mathbf{K}\,B_i$ appears in $\mathbf{P}_n$, and

(Uii) it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_n \setminus U} \not\models H$.

We consider an arbitrary $\mathbf{K}\,H$ in U. If $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \models_p \neg H$, then we can directly infer from condition (3) of Definition 3.10 that $\mathbf{K}\,H$ is *inconsistent* or *classically false* in every p-model of $\mathcal{K}_G$ because we know by the induction hypothesis (i) that all $\mathbf{K}\,H \in \mathbf{P}_n$ are *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$. Alternatively, we assume that $\neg H$ is not derivable, i.e. that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \not\models_p \neg H$. Then, in order to prove the induction step for case (2), it suffices to prove that if the conditions (Ui)-(Uii) are fulfilled for $\mathbf{K}\,H$, then it is *suspiciously true* or *false* in every p-model of $\mathcal{K}_G$. We begin by proving that the previous holds if the conditions (Ui)-(Uii) are fulfilled for $\mathbf{K}\,H$ without any reference to U. For this, we show that if the conditions are fulfilled for $\mathbf{K}\,H$ without any reference to U, then the body of each rule with head $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$ and additionally, that the ontology component does not force $\mathbf{K}\,H$ to be either *true* or *undefined* in every p-model of $\mathcal{K}_G$. As p-models minimize derivable knowledge according to Definition 3.10, we can then infer directly that $\mathbf{K}\,H$ is *suspiciously*

[2] As mentioned by Knorr et al. in [KAH11], these conditions are similar to the conditions of the definition of an unfounded set in [GRS91b].

true or *false* in every p-model of $\mathcal{K}_G$ if it fulfills the conditions without any reference to U. Finally, we will show that the same holds for all $\mathbf{K}\,H \in U$.

Regarding condition (Uia), we know by the induction hypothesis (i) that every $\mathbf{K}\,A_i \in \mathsf{KA}(\mathcal{K}_G) \setminus \mathbf{N}_n$ is *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$. Thus, the body of the rule is also *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$ according to Definition 3.6.

In the case of condition (Uib), we know by the induction hypothesis (i) that every $\mathbf{K}\,B \in \mathbf{P}_n$ is *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$. Therefore, there is a modal **not**-atom $\mathbf{not}\,B_i$ in the body of the rule which is evaluated to one of the truth values *inconsistent*, *suspiciously true* and *false* in every p-model of $\mathcal{K}_G$. Thus, in case of (Uib), the body of the rule is *inconsistent*, *suspiciously true* or *false* in every p-model of $\mathcal{K}_G$ according to Definition 3.6.

Finally, condition (Uii) states that H can only be derived from the objective knowledge of a set containing at least one modal **K**-atom which is not in $\mathbf{N}_{n+1}$. As before, we assume that the condition is already fulfilled for $\mathbf{K}\,H$ when only considering $\mathbf{N}_n$, where $\mathbf{N}_n \subseteq \mathbf{N}_{n+1}$ according to Lemma 4.7. We know by the induction hypothesis (i) that all **K**-atoms $\mathbf{K}\,A_i$ which are not in $\mathbf{N}_n$ are *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$. Besides the case where a modal **K**-atom $\mathbf{K}\,A_i$ is *inconsistent* in a p-model (M, N) of $\mathcal{K}_G$ such that $(\bigcap_{\mathcal{J}\in N} \mathcal{J}, \langle M, N\rangle, \mathcal{N})(A_i) \neq \mathbf{f}$, we know that for all $\mathbf{K}\,A_i$ there is a p-interpretation $\mathcal{I}$ in the set N of every p-model (M, N) of $\mathcal{K}_G$ such that $(\mathcal{I}, \mathcal{M}, \mathcal{N})(A_i) = \mathbf{f}$.

Regarding those $\mathbf{K}\,A_i$ which are *inconsistent* in some p-model (M, N) of $\mathcal{K}_G$ such that $(\bigcap_{\mathcal{J}\in N} \mathcal{J}, \langle M, N\rangle, \mathcal{N})(A_i) \neq \mathbf{f}$, we know according to Corollary 3.38 that it holds that $\mathsf{OB}_{\mathcal{O},\{\mathbf{K}\,H' \mid (*,\langle M,N\rangle,\langle M,N\rangle)(\mathbf{K}\,H')\in\{\mathbf{t},\mathbf{u}\}\}} \models_p A_i$. Consequently, A_i can also be derived from $\mathsf{OB}_{\mathcal{O},\mathbf{N}_n}$ itself as $\mathbf{N}_n$ contains all modal **K**-atoms $\mathbf{K}\,H'$ where $(*, \langle M, N\rangle, \langle M, N\rangle)(\mathbf{K}\,H') \in \{\mathbf{t}, \mathbf{u}\}$ for some p-model (M, N) of $\mathcal{K}_G$, according to the induction hypothesis (i). Hence, we can neglect those $\mathbf{K}\,A_i \notin \mathbf{N}_n$ which are *inconsistent* in some p-model (M, N) of $\mathcal{K}_G$ s.t. $(\bigcap_{\mathcal{J}\in N} \mathcal{J}, \langle M, N\rangle, \mathcal{N})(A_i) \neq \mathbf{f}$ as they are derivable from $\mathbf{N}_n$ and thus, do not satisfy condition (Uii) without any reference to U. We conclude that the only reason for such **K**-atoms $\mathbf{K}\,A_i$ not being in $\mathbf{N}_{n+1}$ is that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \models_p \neg A_i$ holds.

So, for every A_i which cannot be derived from $\mathsf{OB}_{\mathcal{O},\mathbf{N}_n}$ we have that there is a p-interpretation $\mathcal{I}$ in the set N of every p-model (M, N) of $\mathcal{K}_G$ such that $(\mathcal{I}, \mathcal{M}, \mathcal{N})(A_i) = \mathbf{f}$. We also know that the derivation of H from the ontology depends on at least one $\mathbf{K}\,A_i \notin \mathbf{N}_n$ such that $\mathsf{OB}_{\mathcal{O},\mathbf{N}_n} \not\models_p A_i$. As a result, considering the ontology component, H does not have to be $\mathbf{t}$ or $\mathbf{b}$ in every p-interpretation in N of some p-model (M, N) of $\mathcal{K}_G$ and thus, $\mathbf{K}\,H$ is not forced by the ontology component to be either *true* or *undefined* in every p-model of $\mathcal{K}_G$.

Now, consider the conditions (Ui)-(Uii) together. If the body of each rule with head $\mathbf{K}\,H$ is either *false* or *classically false* in some p-model (M, N) of $\mathcal{K}_G$ and $\mathbf{K}\,H$ is not forced by the ontology to be *suspiciously true* in the same p-model (M, N), then it will be minimized to $\mathbf{f}$ in (M, N) due to knowledge minimization by condition (2) of Definition 3.10.

Otherwise, if the body of some rule with head $\mathbf{K}\,H$ is either *inconsistent* or *suspiciously true* in some p-model (M, N) of $\mathcal{K}_G$ or $\mathbf{K}\,H$ is forced by the ontology to be *suspiciously true* in some p-model (M, N) of $\mathcal{K}_G$, then it will be minimized to **st** in (M, N). As a result, $\mathbf{K}\,H$ is *suspiciously true* or *false* in every p-model of $\mathcal{K}_G$. Hence, item (2) of this proposition clearly holds for all modal **K**-atoms in U which fulfill the conditions (Ui)-(Uii) without any reference to U.

Due to Lemma 4.17, the only possible reason why some $\mathbf{K}\,H \in \mathbf{N}_n$ is in U consists in the fact that at least one **K**-atom $\mathbf{K}\,A_i$ in each hybrid dependence set S of $\mathbf{K}\,H$ does not occur in $\mathbf{N}_{n+1}$ anymore. Since the sequence of $\mathbf{P}_i$ is monotonically decreasing according to Lemma 4.7, the previous can only be the case if for every hybrid dependence set S of $\mathbf{K}\,H$ and some $\mathbf{K}\,A_i \in S$ a rule with head $\mathbf{K}\,A_i$ was removed in the MKNF transform because a modal **not**-atom in the rule of the body occurs in $\mathbf{P}_n$, or because $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \models_p \neg A_i$ holds. This implies that each $\mathbf{K}\,H \in U$ hybrid-depends on some modal **K**-atom that fulfills the conditions (Ui)-(Uii) without any reference to U. Consequently, the conditions (Ui)-(Uii) are also fulfilled by those $\mathbf{K}\,H$ which are in $\mathbf{N}_n$, but not in $\mathbf{N}_{n+1}$. As a result, it is the case that for all $\mathbf{K}\,H$ in U it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_n} \models_p \neg H$ or that $\mathbf{K}\,H$ fulfills the complete conditions (Ui)-(Uii) including the reference to U, and we derive that all $\mathbf{K}\,H$ in U are *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$. □

On the basis of Proposition 4.18, it is now straightforward to show the correspondence of the well-founded pair and the well-founded p-model of a ground hybrid MKNF knowledge base. The statement and the proof mirror Theorem 5 and the corresponding proof from [KAH11].

Theorem 4.19 (Soundness and completeness of the procedural definition)**.** *Let $\mathcal{K}_G$ be a ground hybrid MKNF knowledge base and (I_P, I_N) the p-model of $\mathcal{K}_G$ induced by the well-founded pair (T_W, F_W). For any p-model (M, N) of $\mathcal{K}_G$ we have $(M, N) \succeq_k (I_P, I_N)$. Indeed, (I_P, I_N) is the well-founded p-model of $\mathcal{K}_G$.*

Proof. We have shown in Proposition 3.37 that any p-model (M, N) of $\mathcal{K}_G$ induces a pair (T, F) which in turn gives rise to the same p-model (via the objective knowledge). By Proposition 4.18, $\mathbf{K}\,H \in T_W$ implies that $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true* or *true* in every p-model of $\mathcal{K}_G$, and $\mathbf{K}\,H \in F_W$ implies that $\mathbf{K}\,H$ is *inconsistent*, *suspiciously true*, *false* or *classically false* in every p-model of $\mathcal{K}_G$. We conclude that $T_W \subseteq T$ and $F_W \subseteq F$. Furthermore, we know that $I_P = \{I \mid I \models \mathsf{OB}_{\mathcal{O},T_W}\}$ and $I_N = \{I \mid I \models \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F_W}\}$, and also that $M = \{I \mid I \models \mathsf{OB}_{\mathcal{O},T}\}$ and $N = \{I \mid I \models \mathsf{OB}_{\mathcal{O},\mathsf{KA}(\mathcal{K}_G)\setminus F}\}$. It is straightforward to see that $M \subseteq I_P$ and $I_N \subseteq N$, which by Definition 3.31 finishes the proof. □

4.6 Faithfulness of the Program Semantics w.r.t. $WFSX_p$

One main goal and contribution of this thesis consists in establishing a paraconsistent semantics for hybrid knowledge bases which is faithful w.r.t. to some well-studied [Dam96;

PA92] paraconsistent semantics for logic programs, namely $WFSX_p$, if the ontology component is constrained. After having proven that the procedural computation of well-founded p-models by means of the alternating fixpoint construction introduced in the previous chapter is sound and complete, we can use this result to show that the paraconsistent well-founded model $WFM_p(\Pi)$ of an extended logic program Π, as defined by Damásio [Dam96], matches exactly the well-founded pair of the corresponding hybrid MKNF knowledge base. However, we have to impose a restriction on the extended logic programs for which this correspondence holds since classical negation cannot be used in MKNF rules of hybrid MKNF knowledge bases employed in our approach. As a consequence, we are able to show the correspondence if and only if classical negation is only used in front of ground program atoms consisting of a unary predicate and a constant (i.e. *unary program atoms*). In this case, the faithfulness result w.r.t. $WFSX_p$ follows naturally because the procedural computation defined in this thesis works nearly identical to the alternating fixpoint construction developed by Damásio, under the condition that the ontology component is only used to ensure coherence and not to derive **K**-atoms by means of the operator $D_{\mathcal{K}_G}$.

As described in Section 2.3.2, the Coherence Principle is enforced in the alternating fixpoint definition of $WFSX_p$, by not using the original program Π when computing the result of the operator Γ_s (which is the equivalent of the operator $\Gamma'_{\mathcal{K}_G}$ in our approach) but a transformation of the original program, called its *semi-normal version* and denoted by Π_s (cf. Section 2.3.2). In this way, it can be ensured that an objective literal is not derivable by means of the operator Γ_s whenever its explicit complement can be derived by the operator $\Gamma\Gamma_s$, due to the removal of the respective rule from the *GL-transformation* (which resembles the MKNF transform of our approach).

Although the syntax of hybrid MKNF knowledge bases does not allow classically negated atoms in the program component, classical negation in MKNF rules can easily be simulated by designating a prefix for marking the classical negation of unary program atoms. For instance, we can specify that for every first-order atom $H(a)$ in a ground program component, the first-order atom $neg\_H(a)$ denotes the classical negation of $H(a)$. Note that the literals of the form A and $\neg A$ in an extended logic program can also be viewed as independent entities when the Coherence Principle is not taken into account [Dam96]. Thus, they are only connected by the Coherence Principle and in order to simulate classical negation in the program component we just have to make sure that p-models of the knowledge base are coherent w.r.t. to the atoms carrying the prefix $neg\_$. This can be done by delegating the enforcement of the Coherence Principle to the ontology component. For this reason, we do not deal with an empty ontology component here.

In order to show the correspondence of the well-founded pair (T_W, F_W) of a ground hybrid MKNF knowledge base $\mathcal{K}_G$ and the paraconsistent well-founded model of a extended logic program Π (restricted in the aforementioned sense), we define a translation between Π and $\mathcal{K}_G$ and show that $WFM_p(\Pi)$ and (T_W, F_W) are semantically equivalent. To establish the semantical relation between unary program atoms and their explicit

complements, in addition to the program component, we also take a restricted ontology component into account.

Definition 4.20 (MKNF-translation of an extended logic program)**.** Let Π be a ground extended logic program [Dam96] where classical negation is only used in front of unary program atoms. The MKNF-translation of Π is a hybrid MKNF knowledge base $\mathcal{K}_G^\Pi = (\mathcal{O}^\Pi, \mathcal{P}_G^\Pi)$ that fulfills the following condition.

(1) The program component $\mathcal{P}_G^\Pi$ contains precisely one MKNF rule

$$\mathbf{K}\,H \leftarrow \mathbf{K}\,A_1, \ldots, \mathbf{K}\,A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$$

for every rule of the form

$$H \leftarrow A_1, \ldots, A_n, \mathbf{not}\,B_1, \ldots, \mathbf{not}\,B_m$$

in Π, where all objective literals in Π of the form $\neg A$ (consisting of a unary predicate symbol and a constant) are replaced by the first-order atom neg_A in $\mathcal{P}_G^\Pi$.

(2) The ontology component $\mathcal{O}^\Pi$ only contains two axioms of the form

$$P \sqsubseteq \neg neg\_P$$
$$neg\_P \sqsubseteq \neg P$$

for every unary predicate P occurring in Π.

Theorem 4.21 (Faithfulness w.r.t. $WFSX_p$)**.** *Let Π be a ground extended logic program where classical negation is only used in front of unary program atoms, $WFM_p(\Pi)$ the paraconsistent well-founded model of Π, $\mathcal{K}_G^\Pi$ the MKNF-translation of Π and (T_W, F_W) the well-founded pair of $\mathcal{K}_G^\Pi$. Then $\mathbf{K}\,H \in T_W$ if and only if $H \in WFM_p(\Pi)$, and $\mathbf{K}\,H \in F_W$ if and only if $\mathbf{not}\,H \in WFM_p(\Pi)$.*

Proof. First, assume that Π does not contain any classically negated atom in the head of a rule. In this case, the claim follows trivially since the fixpoint computation for obtaining (T_W, F_W) and $WFM_p(\Pi)$ correspond exactly. Note that the previous holds since the operator Γ_s can be replaced by Γ in the computation of $WFM_p(\Pi)$, and the operator $\Gamma'_{\mathcal{K}_G}$ can be replaced by $\Gamma_{\mathcal{K}_G}$ in the computation of the well-founded pair, in this case.

If Π contains a classically negated atom in the head of some rule, applying the operator Γ_s to the semi-normal version of Π ensures that the explicit complement of an objective literal L cannot be derived by Γ_s if L can be derived by Γ, by removing all rules where the explicit complement of L occurs in the head. Now, the deletion of those **K**-atoms $\mathbf{K}\,H$ from the result of the operator $T'_{\mathcal{K}_G,\mathbf{P}_i}$ in the sequence of $\mathbf{N}_i$ for which it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \models_p \neg H$ (according to Definition 4.1) has exactly the same effect. That is to say, whenever $\mathbf{K}\,A$ can be derived by means of $\Gamma_{\mathcal{K}_G}$, $\mathbf{K}\,neg\_A$ cannot be derived

by $\Gamma'_{\mathcal{K}_G}$, and vice versa. The reason is that the definition of the ontology component $\mathcal{O}$ of $\mathcal{K}_G^{\Pi}$ makes sure that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \models_p \neg A$ holds iff it is the case that $neg\_A \in \mathbf{P}_i$, and that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \models_p \neg neg\_A$ holds iff it is the case that $A \in \mathbf{P}_i$. Consequently, the alternating fixpoint computation of (T_W, F_W) and $WFM_p(\Pi)$ correspond and hence, they are semantically equivalent. □

The previous theorem proves that properties that have been shown for the logic $WFSX_p$ can directly be lifted to the program semantics provided by our framework. For instance, the Coherence Principle formalized in Definition 2.6 holds in our semantics, and Contradiction Support Detection is implemented for the program component in our framework (cf. Theorem 1). In addition, the model-based definition of our semantics provides a novel and concise characterization of $WFSX_p$.

4.7 Data Complexity of Computing the Well-Founded Pair

As discussed in the first chapter, using the WFS as the foundation for the logic program semantics of hybrid MKNF knowledge bases has the distinct advantage that, in contrast to the stable model semantics, a low worst-case complexity can be achieved. As a result, the tractability of the whole formalism only depends on the expressivity of the description logic used to formalize the ontology component. Since the procedural computation of the well-founded pair largely resembles[3] the computation of the well-founed partition of Knorr et al. [KAH11], the complexity result of the three-valued well-founded MKNF semantics presented by Knorr et al. carries over to our approach.

The complexity result is stated for data complexity which is measured in terms of the number of concept and role assertions in the ABox of the ontology component and the number of facts in the program component [KAH08].

Theorem 4.22 (Data Complexity w.r.t. the Well-Founded Pair [KAH11])**.** *Let $\mathcal{K}$ be a hybrid MKNF knowledge base. Assuming that entailment of ground DL-atoms in the DL used to formalize the ontology component is decidable with data complexity $\mathcal{C}$, the data complexity of computing the well-founded pair is in $P^{\mathcal{C}}$.*

Proof. Knorr et al. have shown that this claim holds for the alternating fixpoint construction utilized to compute the well-founded partition in their framework [KAH08], by resorting to a result proven by Motik and Rosati that shows that the computation performed by the operator $T_{\mathcal{K}_G}$ is PTime-complete [MR10]. The computation of the alternating fixpoint presented in Chapter 5 only differs from the definition by Knorr et al. in that the deletion of those modal **K**-atoms $\mathbf{K}\,\xi$ for which it holds that $\mathsf{OB}_{\mathcal{O},\mathbf{P}_i} \models_p \neg\xi$, in the sequence of $\mathbf{N}_i$, is moved into the operator $T'_{\mathcal{K}_G,\mathcal{C}}$, which does not alter the data complexity of the formalism. Furthermore, using the paraconsistent entailment operator $\models_p$ instead of the classical entailment operator $\models$ for deriving consequences from the

[3] Only the operator $\Gamma'_{\mathcal{K}_G}$ is adapted to extend the Coherence Principle to the ontology.

ontology does not change the complexity result either since the ontology component is considered to be an oracle in the proof of Motik and Rosati. Hence, the theorem still holds like for the three-valued well-founded MKNF semantics. □

The previous theorem implies that if consequences can be derived from the ontology component in polynomial time by means of the p-entailment operator $\models_p$, the computation of the well-founded pair is tractable as well. For instance, an ontology component expressed in the syntax of the tractable DL $\mathcal{EL}^{++}$ can be used in our framework and, since Maier et al. show that tractability of reasoning in $\mathcal{EL}^{++}$ can be preserved when applying their four-valued semantics [MMH13], the well-founded p-model can be computed in polynomial time then.

5

Conclusions, Related and Future Work

In this thesis, a new six-valued semantics for hybrid MKNF knowledge bases has been developed that allows for paraconsistent reasoning. In this way, our approach extends the expressive power of the three-valued semantics introduced by Knorr et al. in that it also gives a meaning to *inconsistent* knowledge bases, propagates dependencies on contradictions that arise in the program component, and distinguishes between pieces of knowledge that are *false* by default and those which can be proven to be *classically false*.

In Chapter 3, it has been shown that our approach is faithful w.r.t. previous semantics for hybrid MKNF knowledge bases in case the knowledge base is consistent, so that our framework constitutes a proper extension of the work carried out in [MR10] and [KAH11]. Moreover, it has been proven that the paraconsistent semantics assigned to the ontology component in our framework corresponds to the paraconsistent DL semantics $\mathcal{ALC}4$ (without gaps) of Maier et al. [MMH13]. As a result, in many cases the semantics of the ontology component can be computed by means of standard DL-reasoners by applying one of the translations provided by Maier et al. in [MMH13]. Furthermore, in Chapter 3, a definition of a specific p-model among the p-models of a hybrid MKNF knowledge base has been provided, the so-called well-founded p-model, which is the most "skeptical" p-model w.r.t. derivable knowledge.

After the discussion of the model-theoretic characterization of our six-valued hybrid MKNF semantics, a procedural computation of such well-founded p-models has been provided in Chapter 4, which is similar to the procedural computation of paraconsistent well-founded models for LPs [Dam96], and closely resembles the construction developed

by Knorr et al. in [KAH11]. It has been proven that the well-founded pair of a hybrid MKNF knowledge base obtained by an alternating fixpoint construction is sound w.r.t. our model-based definition in that it corresponds to a p-model of the knowledge base (where it is presupposed that the DL used for formalizing the ontology component of the respective hybrid MKNF knowledge base always admits a 4-model). Furthermore, when only considering well-founded p-models, the construction constitutes a sound and complete computation of these particular p-models. Since we have shown that the alternating fixpoint construction has a least fixpoint and thus, always computes a unique well-founded pair of a knowledge base, this implies that a well-founded p-model of a hybrid MKNF knowledge base always exists and that it is unique if the ontology component is always p-satisfied by some p-interpretation. In addition, it has been shown that there is a translation of an extended normal logic program, in which classical negation is only used in front of unary program atoms, into a hybrid MKNF knowledge base such that its well-founded pair exactly corresponds to the paraconsistent well-founded model of the program, i.e. our semantics is faithful w.r.t. $WFSX_p$ under the mentioned condition. Finally, we could show that the efficiency of the approach by Knorr et al. [KAH08] in terms of data complexity holds for our approach as well, so that our procedural characterization describes a tractable algorithm for obtaining the well-founded p-model of a hybrid MKNF knowledge base (provided that paraconsistent reasoning in the employed DL is tractable).

We will now conclude this thesis by reviewing two related approaches and by shortly discussing possible future extensions of our work.

5.1 Related Work

As the related work that is generalized by our approach (i.e. paraconsistent LPs and DLs, as well as approaches that combine rules and ontologies) has already been treated when we discussed the background of our work in Chapter 2, we can restrict ourselves to approaches that pursue the same goal as our approach here. To the best of our knowledge, there are only two other approaches that have addressed the problem of assigning a paraconsistent semantics to hybrid knowledge bases so far. The first one has been published by Huang et al. in [HLH11] and has been extended recently in [HHL14]. Another approach has been developed by Fink and is discussed in [Fin12]. Both of these approaches have in common that they are founded in the SMS for logic programs. Accordingly, the main difference between these two approaches and the work presented here is characterized by the tradeoff between the semantical strength and the computational complexity realized by the SMS and the WFS, respectively, as well as a different underlying formalism in the case of the approach by Fink.

The approach most closely related to the work presented here is the one by Huang et al. [HLH11] since the authors have also developed a paraconsistent semantics for hybrid MKNF knowledge bases. However, their framework is an extension of the hybrid

MKNF semantics developed by Motik and Rosati [MR10] founded in the SMS. For this reason, the authors also consider MKNF rules where disjunctions may occur in the heads of rules. The approach of Huang et al. presented in [HLH11] uses the four truth values of Belnap's logic $\mathcal{FOUR}$ and its implication operator is defined like internal implication in [MMH13], such that it equals the implication operator used in this thesis w.r.t. the truth values $\mathbf{b}$, $\mathbf{t}$ and $\mathbf{f}$. In contrast to our approach, the authors use the same four truth values for interpreting first-order formulas and formulas containing modal operators and therefore, their semantics is founded on four-valued first-order interpretations which are different from the paraconsistent (first-order) interpretations we define. As a result, the authors show the faithfulness of their approach w.r.t. the semantics of Maier et al. [MHL07] where gaps are not removed in the ontology semantics. Moreover, the semantics assigned to the program component in the approach by Huang et al. is faithful w.r.t. the *paraconsistent stable models* introduced by Sakama and Inoue [SI95] and discussed in Section 2.3.1. Further, the authors also present a fixpoint computation of their paraconsistent MKNF models and a linear and consequence preserving translation of four-valued into two-valued hybrid MKNF knowledge bases. A problem that exists in the model-theoretic characterization of Huang et al. and which has been solved in our approach consists in the fact that the truth value *undefined* is minimal w.r.t. the minimization order employed by the authors and thus, *falsity* is minimized to *undefinedness* regarding modal **K**-atoms. However, this fact does not influence the definition of paraconsistent satisfaction provided by Huang et al.

In [HHL14], Huang et al. extend their work in two directions. Firstly, they define a paracoherent semantics for hybrid MKNF knowledge bases by extending their logic to nine truth values. In this way, they are able to assign a paraconsistent MKNF model also to incoherent programs which contain a rule such as $a \leftarrow \mathbf{not}\, a$ and thus, do not have a model in the four-valued approach discussed before. The paracoherent approach of Huang et al. is based on work by Sakama and Inoue [SI95] and the bilattice underlying their nine-valued logic is identical to the bilattice **IX** depicted in Figure 2.2. Secondly, suspicious reasoning in hybrid MKNF knowledge bases is considered in [HHL14] that follows the same strategy followed by Sakama and Inoue [SI95], which has been discussed in Section 2.3.1 and uses the six-valued lattice shown in Figure 2.2. Moreover, Huang et al. also provide procedural definitions of their paracoherent and suspicious MKNF semantics, respectively, and show that the construction of models is sound and complete w.r.t. their model-based definition in the former case, and sound in the latter case.

Like the framework of Huang et al. [HHL14], the approach introduced by Fink [Fin12] also takes inconsistencies as well as incoherencies in hybrid knowledge bases into account. The approach by Fink has its roots in the intuitionistic *Quantified Logic of Here-and-There* and its non-monotonic extension in form of *Quantified Equilibrium Logic* (*QEL*). QEL has been used by de Bruijn et al. to provided a common framework for rules and ontologies [BPPV07]. Though rule and ontology predicates are distinguished in [BPPV07], such

that the former are interpreted classically and the latter are interpreted non-monotonically, Eiter et al. remark that the approach should still be viewed as a full integration approach since it defines a unifying logic for both components [EIKP08]. The approach by Fink is based on a generalization of hybrid knowledge bases as defined by de Bruijn to so-called *hybrid theories*, and on a nine-valued extension of *semi-equilibrium semantics* (which has been introduced by Eiter et al. in order to deal with incoherencies in *Answer Set Programs* [EFM10]) to so-called *paraconsistent semi-equilibrium semantics*. As a result, the framework developed by Fink is able to deal with classical inconsistencies and incoherencies in hybrid knowledge bases simultaneously. Like in the approach by Huang et al., classical predicates in the ontology component are paraconsistently interpreted by means of four truth values, in contrast to just three truth values used in our approach, and the implication operator is defined like internal implication by Fink in this case. Accordingly, the author can show that the semantics is faithful w.r.t. paraconsistent first-order semantics as used e.g. in [MMH13]. Moreover, Fink states in [Fin12] that the paraconsistent semi-equilibrium semantics of the program component of the framework corresponds to well-known paraconsistent and paracoherent ASP semantics, which have been published in [EFM10] and [SI95]. The author further provides a detailed complexity analysis of his approach, for which the condition of *weak DL-safeness* is adopted (also applied in $\mathcal{DL}$+*log* knowledge bases of Rosati [Ros06], which have been described in Section 2.4). Since the approach by Fink extends the SMS, reasoning is generally of non-tractable computational complexity in the framework. Suspicious reasoning is not discussed in the paper [Fin12] by Fink.

5.2 Future Work

Regarding future extensions of our work, there are two obvious directions that can be pursued to expand, respectively enhance, the approach developed in this thesis. Firstly, since a query-driven top-down procedure for deriving the three-valued well-founded MKNF model, named $SLG(\mathcal{O})$, has already been developed by Knorr et al. [AKS13], it would be an evident next step to adapt this procedure in order to compute well-founded p-models. For query-answering, a top-down derivation procedure is considerably faster than the bottom-up computation by means of the alternating fixpoint construction presented in this thesis because only the part of the model which is relevant for answering the respective query has to be computed in the former case [AKS13]. Since the ontology component is treated as an oracle in the $SLG(\mathcal{O})$ procedure, standard reasoners could be directly applied for deriving paraconsistent consequences from the ontology component by utilizing the translations of paraconsistent DLs into classical DLs provided by [MMH13]. The $SLG(\mathcal{O})$ procedure has also been implemented in the *NoHR*-plug-in for the ontology editor *Protégé* [IKL13]. Consequently, by adapting the $SLG(\mathcal{O})$ procdure to our paraconsistent framework it would be possible to allow for paraconsistent reasoning in a future version of the *NoHR*-plug-in.

Secondly, Contradiction Support Detection could be improved in our approach. As discussed in Section 2.3.3, detection of dependencies on contradiction fails in $WFSX_p$ whenever a program atom that is not *inconsistent* itself but can only be derived by involving contradictory information also occurs in the head of a rule whose body is *undefined*. Since our six-valued semantics is constructed in a way such that the meaning assigned to the program component corresponds to $WFSX_p$, the same problem is present in our approach. To tackle this issue, Damásio and Pereira have defined a program transformation that deletes certain rules that have an *undefined* body from a program [DP97]. This transformation is then iteratively applied in the alternating fixpoint construction of the paraconsistent well-founded model. The alternating fixpoint construction presented in this thesis could be adapted in a similar way in order to ensure that Contradiction Support Detection is complete w.r.t. the program component. Regarding the model-theoretic characterization of our semantics, the semantics of the implication operator would have to be changed accordingly in order to maintain the respective soundness and completeness results.

Furthermore, for the same reasons as given to motivate Contradiction Support Detection in the program component, it would be desirable that Contradiction Support Detection would also work for inconsistencies that arise within the ontology component. However, to implement propagation of inconsistencies in the semantics of an ontology is more difficult compared to the implementation in LP semantics since, in contrast to the program component, the ontology component lacks a rule-like structure. To the best of our knowledge, whether and how Contradiction Support Detection could be defined in case of the ontology semantics is an open question.

Bibliography

[ADP02] J. Alcântara, C. V. Damásio, and L. M. Pereira. "Paraconsistent Logic Programs". In: *Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September, 23-26, Proceedings*. Ed. by S. Flesca, S. Greco, N. Leone, and G. Ianni. Vol. 2424. Lecture Notes in Computer Science. Springer, 2002, pp. 345–356. ISBN: 3-540-44190-5. DOI: 10.1007/3-540-45757-7_29. URL: http://dx.doi.org/10.1007/3-540-45757-7_29.

[AGM85] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. "On the Logic of Theory Change: Partial Meet Contraction and Revision Functions". In: *J. Symb. Log.* 50.2 (1985), pp. 510–530. DOI: 10.2307/2274239. URL: http://dx.doi.org/10.2307/2274239.

[ADP94] J. J. Alferes, C. V. Damásio, and L. M. Pereira. "SLX - A Top-down Derivation Procedure for Programs with Explicit Negation". In: *Logic Programming, Proceedings of the 1994 International Symposium, Ithaca, New York, USA, November 13-17, 1994*. Ed. by M. Bruynooghe. MIT Press, 1994, pp. 424–438. ISBN: 0-262-52191-1.

[ADP95] J. J. Alferes, C. V. Damásio, and L. M. Pereira. "A Logic Programming System for Nonmonotonic Reasoning". In: *J. Autom. Reasoning* 14.1 (1995), pp. 93–147. DOI: 10.1007/BF00883931. URL: http://dx.doi.org/10.1007/BF00883931.

[AKS13] J. J. Alferes, M. Knorr, and T. Swift. "Query-Driven Procedures for Hybrid MKNF Knowledge Bases". In: *ACM Trans. Comput. Log.* 14.2 (2013), p. 16. DOI: 10.1145/2480759.2480768. URL: http://doi.acm.org/10.1145/2480759.2480768.

[Alf93] J. J. A. Alferes. "Semantics of logic programs with explicit negation". PhD thesis. 1993.

[AB94] K. R. Apt and R. N. Bol. "Logic Programming and Negation: A Survey". In: *J. Log. Program.* 19/20 (1994), pp. 9–71. DOI: 10.1016/0743-1066(94)

90024-8. URL: http://dx.doi.org/10.1016/0743-1066(94)90024-8.

[Ari04] O. Arieli. "Paraconsistent Preferential Reasoning by Signed Quantified Boolean Formulae". In: *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*. Ed. by R. L. de Mántaras and L. Saitta. IOS Press, 2004, pp. 773–777. ISBN: 1-58603-452-9.

[AA96] O. Arieli and A. Avron. "Reasoning with Logical Bilattices". In: *Journal of Logic, Language and Information* 5.1 (1996), pp. 25–63. DOI: 10.1007/BF00215626. URL: http://dx.doi.org/10.1007/BF00215626.

[AA00] O. Arieli and A. Avron. "Bilattices and paraconsistency". In: *Frontiers in Paraconsistent Logic* (2000), pp. 11–27.

[BCMNP03] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. ISBN: 0-521-78176-0.

[BG94] C. Baral and M. Gelfond. "Logic Programming and Knowledge Representation". In: *J. Log. Program.* 19/20 (1994), pp. 73–148. DOI: 10.1016/0743-1066(94)90025-6. URL: http://dx.doi.org/10.1016/0743-1066(94)90025-6.

[Bat07] D. Batens. "A Universal Logic Approach to Adaptive Logics". In: *Logica Universalis* 1.1 (2007), pp. 221–242. DOI: 10.1007/s11787-006-0012-5. URL: http://dx.doi.org/10.1007/s11787-006-0012-5.

[BJ77a] N. D. Belnap Jr. "A useful four-valued logic". In: *Modern uses of multiple-valued logic*. Springer, 1977, pp. 5–37.

[BJ77b] N. D. Belnap Jr. "How a computer should think". In: *Contemporary Aspects of Philosophy* (1977). Ed. by G. Ryle, pp. 30–56.

[BPPV07] J. de Bruijn, D. Pearce, A. Polleres, and A. Valverde. "Quantified Equilibrium Logic and Hybrid Rules". In: *Web Reasoning and Rule Systems, First International Conference, RR 2007, Innsbruck , Austria, June 7-8, 2007, Proceedings*. Ed. by M. Marchiori, J. Z. Pan, and C. de Sainte Marie. Vol. 4524. Lecture Notes in Computer Science. Springer, 2007, pp. 58–72. ISBN: 978-3-540-72981-5. DOI: 10.1007/978-3-540-72982-2_5. URL: http://dx.doi.org/10.1007/978-3-540-72982-2_5.

[CM01] W. A. Carnielli and J. Marcos. "Ex contradictione non sequitur quodlibet". In: *Proceedings of the Advanced Reasoning Forum Conference*. Vol. 1. 2001, pp. 89–109.

[CW96] W. Chen and D. S. Warren. "Tabled Evaluation With Delaying for General Logic Programs". In: *J. ACM* 43.1 (1996), pp. 20–74. DOI: 10.1145/227595.227597. URL: http://doi.acm.org/10.1145/227595.227597.

[Cos74] N. C. A. da Costa. "On the theory of inconsistent formal systems". In: *Notre Dame Journal of Formal Logic* 15.4 (1974), pp. 497–510. DOI: 10.1305/ndjfl/1093891487. URL: http://dx.doi.org/10.1305/ndjfl/1093891487.

[CD95] N. C. A. da Costa and F. A. Doria. "On Jaskowski's discussive logic". In: *Studia Logica* 54.1 (1995), pp. 33–60. DOI: 10.1007/BF01058531. URL: http://dx.doi.org/10.1007/BF01058531.

[DCKB07] N. C. Da Costa, D. Krause, and O. Bueno. "Paraconsistent logics and paraconsistency". In: *2007* (2007), pp. 791–911.

[Dam96] C. V. Damásio. "Paraconsistent extended logic programming with constraints". PhD thesis. 1996.

[DP95] C. V. Damásio and L. M. Pereira. "A Model Theory for Paraconsistent Logic Programming". In: *Progress in Artificial Intelligence, 7th Portuguese Conference on Artificial Intelligence, EPIA '95, Funchal, Madeira Island, Portugal, October 3-6, 1995, Proceedings*. Ed. by C. A. Pinto-Ferreira and N. J. Mamede. Vol. 990. Lecture Notes in Computer Science. Springer, 1995, pp. 377–386. ISBN: 3-540-60428-6. DOI: 10.1007/3-540-60428-6_32. URL: http://dx.doi.org/10.1007/3-540-60428-6_32.

[DP97] C. V. Damásio and L. M. Pereira. "A Paraconsistent Semantics with Contradiction Support Detection". In: *Logic Programming and Nonmonotonic Reasoning, 4th International Conference, LPNMR'97, Dagstuhl Castle, Germany, July 28-31, 1997, Proceedings*. Ed. by J. Dix, U. Furbach, and A. Nerode. Vol. 1265. Lecture Notes in Computer Science. Springer, 1997, pp. 224–243. ISBN: 3-540-63255-7. DOI: 10.1007/3-540-63255-7_18. URL: http://dx.doi.org/10.1007/3-540-63255-7_18.

[DP98] C. V. Damásio and L. M. Pereira. "A survey of paraconsistent semantics for logic programs". In: *Reasoning with Actual and Potential Contradictions*. Springer, 1998, pp. 241–320.

[DEGV01] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. "Complexity and expressive power of logic programming". In: *ACM Comput. Surv.* 33.3 (2001), pp. 374–425. DOI: 10.1145/502807.502810. URL: http://doi.acm.org/10.1145/502807.502810.

[Dix95] J. Dix. "A Classification Theory of Semantics of Normal Logic Programs: II. Weak Properties". In: *Fundam. Inform.* 22.3 (1995), pp. 257–288. DOI: 10.3233/FI-1995-2234. URL: http://dx.doi.org/10.3233/FI-1995-2234.

[DLNNS98] F. M. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. "An Epistemic Operator for Description Logics". In: *Artif. Intell.* 100.1-2 (1998), pp. 225–274. DOI: 10.1016/S0004-3702(98)00009-5. URL: http://dx.doi.org/10.1016/S0004-3702(98)00009-5.

[EFM10] T. Eiter, M. Fink, and J. Moura. "Paracoherent Answer Set Programming". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. Ed. by F. Lin, U. Sattler, and M. Truszczynski. AAAI Press, 2010. URL: http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1339.

[EFS13] T. Eiter, M. Fink, and D. Stepanova. "Data Repair of Inconsistent DL-Programs". In: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. Ed. by F. Rossi. IJCAI/AAAI, 2013. ISBN: 978-1-57735-633-2. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6832.

[EG92] T. Eiter and G. Gottlob. "On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals". In: *Artif. Intell.* 57.2-3 (1992), pp. 227–270. DOI: 10.1016/0004-3702(92)90018-S. URL: http://dx.doi.org/10.1016/0004-3702(92)90018-S.

[EIKP08] T. Eiter, G. Ianni, T. Krennwallner, and A. Polleres. "Rules and Ontologies for the Semantic Web". In: *Reasoning Web, 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures*. Ed. by C. Baroglio, P. A. Bonatti, J. Maluszynski, M. Marchiori, A. Polleres, and S. Schaffert. Vol. 5224. Lecture Notes in Computer Science. Springer, 2008, pp. 1–53. ISBN: 978-3-540-85656-6. DOI: 10.1007/978-3-540-85658-0_1. URL: http://dx.doi.org/10.1007/978-3-540-85658-0_1.

[EILS11] T. Eiter, G. Ianni, T. Lukasiewicz, and R. Schindlauer. "Well-founded semantics for description logic programs in the semantic web". In: *ACM Trans. Comput. Log.* 12.2 (2011), p. 11. DOI: 10.1145/1877714.1877717. URL: http://doi.acm.org/10.1145/1877714.1877717.

[EILST08] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. "Combining answer set programming with description logics for the Semantic Web". In: *Artif. Intell.* 172.12-13 (2008), pp. 1495–1539. DOI: 10.1016/j.artint.2008.04.002. URL: http://dx.doi.org/10.1016/j.artint.2008.04.002.

[EIST06] T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits. "Effective Integration of Declarative Rules with External Evaluations for Semantic-Web Reasoning". In: *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*. Ed. by Y. Sure and J. Domingue. Springer, 2006, pp. 273–287. DOI: 10.1007/11762256_22. URL: http://dx.doi.org/10.1007/11762256_22.

[Fin12] M. Fink. "Paraconsistent Hybrid Theories". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. Ed. by G. Brewka, T. Eiter, and S. A. McIlraith. AAAI Press, 2012. ISBN: 978-1-57735-560-1. URL: http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4552.

[Fit90] M. Fitting. "Bilattices in logic programming". In: *Multiple-Valued Logic, 1990., Proceedings of the Twentieth International Symposium on*. IEEE. 1990, pp. 238–246.

[Fit91a] M. Fitting. "Bilattices and the Semantics of Logic Programming". In: *J. Log. Program.* 11.1&2 (1991), pp. 91–116. DOI: 10.1016/0743-1066(91)90014-G. URL: http://dx.doi.org/10.1016/0743-1066(91)90014-G.

[Fit91b] M. Fitting. "Kleene's Logic, Generalized". In: *J. Log. Comput.* 1.6 (1991), pp. 797–810. DOI: 10.1093/logcom/1.6.797. URL: http://dx.doi.org/10.1093/logcom/1.6.797.

[Fit96] M. Fitting. *First-Order Logic and Automated Theorem Proving, Second Edition*. Graduate Texts in Computer Science. Springer, 1996. ISBN: 978-1-4612-7515-2. DOI: 10.1007/978-1-4612-2360-3. URL: http://dx.doi.org/10.1007/978-1-4612-2360-3.

[Gel89] A. V. Gelder. "The Alternating Fixpoint of Logic Programs with Negation". In: *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 29-31, 1989, Philadelphia, Pennsylvania, USA*. Ed. by A. Silberschatz. ACM Press, 1989, pp. 1–10. ISBN: 0-89791-308-6. DOI: 10.1145/73721.73722. URL: http://doi.acm.org/10.1145/73721.73722.

[GRS91a] A. V. Gelder, K. A. Ross, and J. S. Schlipf. "The Well-Founded Semantics for General Logic Programs". In: *J. ACM* 38.3 (1991), pp. 620–650. DOI: 10.1145/116825.116838. URL: http://doi.acm.org/10.1145/116825.116838.

[GRS91b] A. V. Gelder, K. A. Ross, and J. S. Schlipf. "The Well-Founded Semantics for General Logic Programs". In: *J. ACM* 38.3 (1991), pp. 620–650. DOI: 10.

1145/116825.116838. URL: http://doi.acm.org/10.1145/116825.116838.

[GL88] M. Gelfond and V. Lifschitz. "The Stable Model Semantics for Logic Programming". In: *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, August 15-19, 1988 (2 Volumes)*. Ed. by R. A. Kowalski and K. A. Bowen. MIT Press, 1988, pp. 1070–1080. ISBN: 0-262-61056-6.

[GL91] M. Gelfond and V. Lifschitz. "Classical Negation in Logic Programs and Disjunctive Databases". In: *New Generation Comput.* 9.3/4 (1991), pp. 365–386. DOI: 10.1007/BF03037169. URL: http://dx.doi.org/10.1007/BF03037169.

[Gin88] M. L. Ginsberg. "Multivalued logics: a uniform approach to reasoning in artificial intelligence". In: *Computational Intelligence* 4 (1988), pp. 265–316. DOI: 10.1111/j.1467-8640.1988.tb00280.x. URL: http://dx.doi.org/10.1111/j.1467-8640.1988.tb00280.x.

[GGOP13] L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. "A non-monotonic Description Logic for reasoning about typicality". In: *Artif. Intell.* 195 (2013), pp. 165–202. DOI: 10.1016/j.artint.2012.10.004. URL: http://dx.doi.org/10.1016/j.artint.2012.10.004.

[Got14] S. Gottwald. "Many-Valued Logic". In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta. Spring 2014. 2014.

[GS00] J. Grant and V. Subrahmanian. "Applications of paraconsistency in data and knowledge bases". In: *Synthese* 125.1-2 (2000), pp. 121–132.

[GHVD03] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. "Description logic programs: combining logic programs with description logic". In: *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*. Ed. by G. Hencsey, B. White, Y. R. Chen, L. Kovács, and S. Lawrence. ACM, 2003, pp. 48–57. DOI: 10.1145/775152.775160. URL: http://doi.acm.org/10.1145/775152.775160.

[HNV05] S. Heymans, D. V. Nieuwenborgh, and D. Vermeir. "Guarded Open Answer Set Programming". In: *Logic Programming and Nonmonotonic Reasoning, 8th International Conference, LPNMR 2005, Diamante, Italy, September 5-8, 2005, Proceedings*. Ed. by C. Baral, G. Greco, N. Leone, and G. Terracina. Springer, 2005, pp. 92–104. DOI: 10.1007/11546207_8. URL: http://dx.doi.org/10.1007/11546207_8.

[HP09] P. Hitzler and B. Parsia. "Ontologies and Rules". In: *Handbook on Ontologies*. Ed. by S. Staab and R. Studer. 2nd ed. Springer, 2009, pp. 111–132.

[HP04] I. Horrocks and P. F. Patel-Schneider. "A proposal for an owl rules language". In: *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*. Ed. by S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills. ACM, 2004, pp. 723–731. ISBN: 1-58113-844-X. DOI: 10.1145/988672.988771. URL: http://doi.acm.org/10.1145/988672.988771.

[HHL14] S. Huang, J. Hao, and D. Luo. "Incoherency Problems in a Combination of Description Logics and Rules". In: *J. Applied Mathematics* 2014 (2014). DOI: 10.1155/2014/604753. URL: http://dx.doi.org/10.1155/2014/604753.

[HLH11] S. Huang, Q. Li, and P. Hitzler. "Paraconsistent Semantics for Hybrid MKNF Knowledge Bases". In: *Web Reasoning and Rule Systems - 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings*. Ed. by S. Rudolph and C. Gutierrez. Vol. 6902. Lecture Notes in Computer Science. Springer, 2011, pp. 93–107. ISBN: 978-3-642-23579-5. DOI: 10.1007/978-3-642-23580-1_8. URL: http://dx.doi.org/10.1007/978-3-642-23580-1_8.

[Hun02] A. Hunter. "Measuring Inconsistency in Knowledge via Quasi-Classical Models". In: *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*. Ed. by R. Dechter and R. S. Sutton. AAAI Press / The MIT Press, 2002, pp. 68–73. URL: http://www.aaai.org/Library/AAAI/2002/aaai02-011.php.

[HMS05] U. Hustadt, B. Motik, and U. Sattler. "Data Complexity of Reasoning in Very Expressive Description Logics". In: *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Ed. by L. P. Kaelbling and A. Saffiotti. Professional Book Center, 2005, pp. 466–471. URL: http://www.ijcai.org/papers/0326.pdf.

[IKL13] V. Ivanov, M. Knorr, and J. Leite. "A Query Tool for *EL* with Non-monotonic Rules". In: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*. Ed. by H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz. Vol. 8218. Lecture Notes in Computer Science. Springer, 2013, pp. 216–231. ISBN: 978-3-642-41334-6. DOI: 10.1007/978-3-642-41335-3_14. URL: http://dx.doi.org/10.1007/978-3-642-41335-3_14.

[Jaś69] S. Jaśkowski. "Propositional calculus for contradictory deductive systems". In: *Studia Logica* 24.1 (1969), pp. 143–157.

[KAH08] M. Knorr, J. J. Alferes, and P. Hitzler. "A Coherent Well-founded Model for Hybrid MKNF Knowledge Bases". In: *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*. Ed. by M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris. Vol. 178. Frontiers in Artificial Intelligence and Applications. IOS Press, 2008, pp. 99–103. ISBN: 978-1-58603-891-5. DOI: 10.3233/978-1-58603-891-5-99. URL: http://dx.doi.org/10.3233/978-1-58603-891-5-99.

[KAH11] M. Knorr, J. J. Alferes, and P. Hitzler. "Local closed world reasoning with description logics under the well-founded semantics". In: *Artif. Intell.* 175.9-10 (2011), pp. 1528–1554. DOI: 10.1016/j.artint.2011.01.007. URL: http://dx.doi.org/10.1016/j.artint.2011.01.007.

[Lan06] A. C. Lang. "Four-valued logics for paraconsistent reasoning". PhD thesis. Diplomarbeit. Dresden University of Technology, 2006.

[Lif91] V. Lifschitz. "Nonmonotonic Databases and Epistemic Queries". In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*. Ed. by J. Mylopoulos and R. Reiter. Morgan Kaufmann, 1991, pp. 381–386.

[Llo87] J. W. Lloyd. *Foundations of Logic Programming, 2nd Edition*. Springer, 1987. ISBN: 3-540-18199-7.

[MH09] Y. Ma and P. Hitzler. "Paraconsistent Reasoning for OWL 2". In: *Web Reasoning and Rule Systems, Third International Conference, RR 2009, Chantilly, VA, USA, October 25-26, 2009, Proceedings*. Ed. by A. Polleres and T. Swift. Vol. 5837. Lecture Notes in Computer Science. Springer, 2009, pp. 197–211. ISBN: 978-3-642-05081-7. DOI: 10.1007/978-3-642-05082-4_14. URL: http://dx.doi.org/10.1007/978-3-642-05082-4_14.

[MHL07] Y. Ma, P. Hitzler, and Z. Lin. "Algorithms for Paraconsistent Reasoning with OWL". In: *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings*. Ed. by E. Franconi, M. Kifer, and W. May. Vol. 4519. Lecture Notes in Computer Science. Springer, 2007, pp. 399–413. ISBN: 978-3-540-72666-1. DOI: 10.1007/978-3-540-72667-8_29. URL: http://dx.doi.org/10.1007/978-3-540-72667-8_29.

[Mai10] F. Maier. "Extending Paraconsistent SROIQ". In: *Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings*. Ed. by P. Hitzler and T. Lukasiewicz. Vol. 6333. Lecture Notes in Computer Science. Springer, 2010, pp. 118–132.

ISBN: 978-3-642-15917-6. DOI: 10.1007/978-3-642-15918-3_10. URL: http://dx.doi.org/10.1007/978-3-642-15918-3_10.

[MMH13] F. Maier, Y. Ma, and P. Hitzler. "Paraconsistent OWL and related logics". In: *Semantic Web* 4.4 (2013), pp. 395–427. DOI: 10.3233/SW-2012-0066. URL: http://dx.doi.org/10.3233/SW-2012-0066.

[McC86] J. McCarthy. "Applications of Circumscription to Formalizing Common-Sense Knowledge". In: *Artif. Intell.* 28.1 (1986), pp. 89–116. DOI: 10.1016/0004-3702(86)90032-9. URL: http://dx.doi.org/10.1016/0004-3702(86)90032-9.

[Mid11] C. A. Middelburg. "A Survey of Paraconsistent Logics". In: *CoRR* abs/1103.4324 (2011). URL: http://arxiv.org/abs/1103.4324.

[MR06] B. Motik and R. Rosati. *Closing Semantic Web Ontologies*. Tech. rep. University of Manchester, UK, 2006. URL: web.comlab.ox.ac.uk/people/Boris.Motik/pubs/mr06closing-report.pdf.

[MR07] B. Motik and R. Rosati. "A Faithful Integration of Description Logics with Logic Programming". In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. Ed. by M. M. Veloso. 2007, pp. 477–482. URL: http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-075.pdf.

[MR10] B. Motik and R. Rosati. "Reconciling description logics and rules". In: *J. ACM* 57.5 (2010). DOI: 10.1145/1754399.1754403. URL: http://doi.acm.org/10.1145/1754399.1754403.

[MSS05] B. Motik, U. Sattler, and R. Studer. "Query Answering for OWL-DL with rules". In: *J. Web Sem.* 3.1 (2005), pp. 41–60. DOI: 10.1016/j.websem.2005.05.001. URL: http://dx.doi.org/10.1016/j.websem.2005.05.001.

[PA92] L. M. Pereira and J. J. Alferes. "Well Founded Semantics for Logic Programs with Explicit Negation". In: *ECAI*. 1992, pp. 102–106.

[Pri79] G. Priest. "The logic of paradox". In: *Journal of Philosophical logic* 8.1 (1979), pp. 219–241.

[Rei80] R. Reiter. "A Logic for Default Reasoning". In: *Artif. Intell.* 13.1-2 (1980), pp. 81–132. DOI: 10.1016/0004-3702(80)90014-4. URL: http://dx.doi.org/10.1016/0004-3702(80)90014-4.

[Ros06] R. Rosati. "DL+log: Tight Integration of Description Logics and Disjunctive Datalog". In: *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*. Ed. by P. Doherty, J. Mylopoulos, and C. A. Welty. AAAI

Press, 2006, pp. 68–78. URL: http://www.aaai.org/Library/KR/2006/kr06-010.php.

[RKH08] S. Rudolph, M. Krötzsch, and P. Hitzler. "Cheap Boolean Role Constructors for Description Logics". In: *Logics in Artificial Intelligence, 11th European Conference, JELIA 2008, Dresden, Germany, September 28 - October 1, 2008. Proceedings*. Ed. by S. Hölldobler, C. Lutz, and H. Wansing. Vol. 5293. Lecture Notes in Computer Science. Springer, 2008, pp. 362–374. ISBN: 978-3-540-87802-5. DOI: 10.1007/978-3-540-87803-2_30. URL: http://dx.doi.org/10.1007/978-3-540-87803-2_30.

[Sak92] C. Sakama. "Extended Well-Founded Semantics for Paraconsistent Logic Programs". In: *FGCS*. 1992, pp. 592–599.

[SI95] C. Sakama and K. Inoue. "Paraconsistent Stable Semantics for Extended Disjunctive Programs". In: *J. Log. Comput.* 5.3 (1995), pp. 265–285. DOI: 10.1093/logcom/5.3.265. URL: http://dx.doi.org/10.1093/logcom/5.3.265.

[SLS11] M. Slota, J. Leite, and T. Swift. "Splitting and updating hybrid knowledge bases". In: *TPLP* 11.4-5 (2011), pp. 801–819. DOI: 10.1017/S1471068411000317. URL: http://dx.doi.org/10.1017/S1471068411000317.

[Wik14] Wikipedia. *Semantic Web — Wikipedia, The Free Encyclopedia*. [Online; accessed 15-September-2014]. 2014. URL: http://en.wikipedia.org/w/index.php?title=Semantic_Web&oldid=624033348.

[ZL08] X. Zhang and Z. Lin. "Paraconsistent Reasoning with Quasi-classical Semantic in". In: *Web Reasoning and Rule Systems, Second International Conference, RR 2008, Karlsruhe, Germany, October 31-November 1, 2008. Proceedings*. Ed. by D. Calvanese and G. Lausen. Vol. 5341. Lecture Notes in Computer Science. Springer, 2008, pp. 222–229. ISBN: 978-3-540-88736-2. DOI: 10.1007/978-3-540-88737-9_18. URL: http://dx.doi.org/10.1007/978-3-540-88737-9_18.

[ZLW10] X. Zhang, Z. Lin, and K. Wang. "Towards a Paradoxical Description Logic for the Semantic Web". In: *Foundations of Information and Knowledge Systems, 6th International Symposium, FoIKS 2010, Sofia, Bulgaria, February 15-19, 2010. Proceedings*. Ed. by S. Link and H. Prade. Vol. 5956. Lecture Notes in Computer Science. Springer, 2010, pp. 306–325. ISBN: 978-3-642-11828-9. DOI: 10.1007/978-3-642-11829-6_20. URL: http://dx.doi.org/10.1007/978-3-642-11829-6_20.

[ZQML09] X. Zhang, G. Qi, Y. Ma, and Z. Lin. "Quasi-Classical Semantics for Expressive Description Logics". In: *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*. Ed. by B. C.

Grau, I. Horrocks, B. Motik, and U. Sattler. Vol. 477. CEUR Workshop Proceedings. CEUR-WS.org, 2009. URL: http://ceur-ws.org/Vol-477/paper_1.pdf.

[ZXL09] X. Zhang, G. Xiao, and Z. Lin. "A Tableau Algorithm for Handling Inconsistency in OWL". In: *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*. Ed. by L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. P. B. Simperl. Vol. 5554. Lecture Notes in Computer Science. Springer, 2009, pp. 399–413. ISBN: 978-3-642-02120-6. DOI: 10.1007/978-3-642-02121-3_31. URL: http://dx.doi.org/10.1007/978-3-642-02121-3_31.