Consistency checking for extended description logics

Olivier Couchariere[ab], Marie-Jeanne Lesot[b], and Bernadette Bouchon-Meunier[b]

[a]Thales Communications, 160 boulevard de Valmy, F-92704 Colombes Cedex France
[b]Université Pierre et Marie Curie-Paris6, CNRS UMR 7606, LIP6, 104, avenue du Président Kennedy, Paris, F-75016, France
Olivier.Couchariere@fr.thalesgroup.com
{Marie-Jeanne.Lesot,Bernadette.Bouchon-Meunier}@lip6.fr

Abstract. In this paper we consider the extensions of description logics that were proposed to represent uncertain or vague knowledge, focusing on the fuzzy and possibilistic formalisms. We compare these two approaches and comment on their differences concentrating on the consistency issue of knowledge bases represented in these extended frameworks. We present a classification of existing algorithms and describe a new method for the possibilistic case that yields an inconsistency degree and not only a binary answer to the consistency question. The proposed algorithm is based on a direct extension of the tableau algorithm to the possibilistic case, for which we introduce appropriate clash and completion rule definitions.

Key words: Description logics, possibilistic logic, fuzzy logic, inconsistency, tableau algorithm

1 Introduction

Description logics (DLs) constitute a knowledge representation framework equipped with tools to automatically infer implicit information present in a knowledge base: they belong to a subset of first order logic, which endows them with theoretically well-founded and computationally efficient methods.

Being based on classic binary logic, description logics are not appropriate to deal with imprecise or uncertain information, although the latter is frequently observed. To overcome this limit, several extensions of DLs have been proposed relying on various extensions of classic logic: there have been probabilistic [1, 2], possibilistic [3, 4], and fuzzy [5–8] extensions that have fundamental differences in terms of semantics and thus in the types of information they can model. In a nutshell, as will be detailed in the following, the probabilistic and possibilistic approaches capture uncertainty and make it possible to represent uncertain information, whereas the fuzzy approach captures vagueness and allows to represent imprecise knowledge.

In this paper, we first present a survey of the fuzzy and possibilistic extensions of DLs. We underline their semantic differences, as well as the consequences the latter imply. Focusing on the central inconsistency checking reasoning service, we present a classification of the existing algorithms in three categories, detailing and commenting each of them. We then describe a new method, based on a direct extension of the tableau algorithm to handle the necessity degrees associated with possibilistic formulas. The proposed method does not only determine whether the knowledge base is consistent or not, it computes an inconsistency degree that quantifies the level of inconsistency of the knowledge base.

The paper is organised as follows: in Section 2 we recall the principles and main characteristics of the fuzzy and possibilistic extensions of description logics. In Section 3 we provide a structured survey of existing algorithms to check the consistency of a knowledge base represented in the fuzzy or in the possibilistic DL formalisms. Section 4 describes our extended tableau algorithm to compute an inconsistency degree for a possibilistic description logic base. Lastly Section 5 concludes the paper and presents some perspectives.

2 Fuzzy and possibilistic DLs: extending DLs to deal with vagueness and uncertainty

2.1 Motivation

Description logics constitute a theoretical framework to formalise and represent knowledge. They rely on first order logic, which endows them with well-founded reasoning and deduction capabilities for knowledge inference. We assume that the reader is familiar with description logics and refer to [9] for a detailed presentation.

The fact that they are a subset of first order logic makes DLs appropriate to model and deal with crisp information. Yet real world applications is often tainted with uncertainty and/or imprecision or vagueness, which cannot be modeled by classic DLs. In order to overcome these weaknesses, several extensions have been provided, among which the fuzzy and possibilistic description logics we detail in the following. They respectively rely on fuzzy logic [10] and possibility logic [11].

It is to be underlined that although they share some syntactical features, these two approaches present fundamental differences: in both cases, the logical formulas of classic logic are enriched with numerical coefficients of the interval $[0, 1]$, but the meanings of these coefficients differ. Indeed, in the fuzzy case, they are interpreted in terms of truth degrees, whereas in the possibilistic case, they correspond to uncertainty degrees. For instance, in fuzzy DLs, the formula (john:Tall, 0.6) means it can be asserted to the degree at least 0.6 that "John is tall": the concept "tall" is a vague one, and the truth value of the assertion cannot be reduced to a binary notion. In the possibilistic case, the truth value of the formula remains binary, but a degree of certainty is expressed, underlying a possible lack of confidence in the assertion. It is to be noted that these differences actually trace back to the properties and differences of the fuzzy and possibility logic theories (see for instance [12]).

In the following, we detail the differences at a more formal level, as well as the main features of the corresponding languages.

2.2 Fuzzy description logics

The fuzzy logic theory was first introduced by Zadeh [10] to model vague and imprecise concepts, and perform logical reasoning from them. The fuzzy extension to description logics was provided by [13–15] and later extensively studied by several authors [6, 16, 7, 8], who present different extensions of the classic definitions, leading to frameworks with different properties.

Syntax In fuzzy description logics, a formula is a couple of the type $(\phi \bowtie \alpha)$, where ϕ is any classic DL formula, e.g., of the form $C \sqsubseteq D, C \equiv D, a : C$, or $(a, b) : R$, using the classic notations; $\bowtie \in \{\leq, \geq, >, <\}$, and α is a numerical value in the interval $[0, 1]$. It is interpreted as a formula associated with a lower or upper bound on its truth value. Tresp and Molitor [15] consider the special case where the coefficient associated with a formula equals its truth degree.

Semantics The formal semantics of fuzzy DLs are related to the fuzzy set theory: given a domain of interpretation $\Delta^{\mathcal{I}}$, the interpretation function $\cdot^{\mathcal{I}}$ maps each concept to a fuzzy subset of $\Delta^{\mathcal{I}}$, or equivalently to a membership function $\Delta^{\mathcal{I}} \rightarrow [0, 1]$. Similarly, roles are mapped to fuzzy subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, whereas individuals are simply mapped to elements of $\Delta^{\mathcal{I}}$ as in classic DLs.

As a consequence the connectives are interpreted as operations on fuzzy sets: the conjunction operator for instance corresponds to the computation of the intersection of the fuzzy sets, for which any t-norm t can be used, defining $\forall x \in \Delta^{\mathcal{I}}, (C \sqcap D)^{\mathcal{I}}(x) = t(C^{\mathcal{I}}(x), D^{\mathcal{I}}(x))$. Likewise, the disjunction operation depends on the choice of a t-conorm, the existential and universal quantifiers on the choice of a fuzzy implication. The combined selection of these operators determines the property of the induced logic, for instance the possible preservation of properties such as the excluded-middle. A common choice is the min and max norms and the Kleene-Dienes implication.

Regarding the extension of the inclusion axiom, several approaches have been proposed: the direct method [6, 8] consists in considering that an interpretation satisfies $C \sqsubseteq D$ iff $\forall x \in \Delta^{\mathcal{I}}, C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$, relying on the fuzzy set inclusion definition. A fuzzy definition of concept inclusion has also been proposed [15, 16]: it defines a degree of subsumption between two concepts C and D as $\inf_{x \in \Delta^{\mathcal{I}}} i(C^{\mathcal{I}}(x), D^{\mathcal{I}}(x))$ where i denotes the implication function.

Satisfiability and consistency Given a domain $\Delta^{\mathcal{I}}$ and operators for the semantics of $\sqcap, \sqcup$ and the implication, an interpretation $\cdot^{\mathcal{I}}$ satisfies a formula $(\phi \geq \alpha)$ if the truth value of ϕ in this interpretation, $\phi^{\mathcal{I}}$, respects the constraint $\phi^{\mathcal{I}} \geq \alpha$. For a fuzzy logic formula with the other inequality constraints $(\leq, <, >)$, likewise, the interpretation satisfies the formula if the truth degree it leads to verifies the constraint. A set of formulas is consistent if there exists an interpretation that satisfies all formulas in the set.

Contradictory pairs Determining whether a knowledge base is inconsistent or not can be seen as identifying pairs of incompatible formulas. In the fuzzy DL formalism contradictory pairs are of the form [6]

$$\{(\phi \geq \alpha), (\phi < \beta)\} \text{ with } \alpha \geq \beta$$

Three other contradiction types are obtained when considering the other comparison operators [6]. This is due to the fact that the truth degree x cannot simultaneously verify the two constraints $\alpha \leq x < \beta$ if $\alpha \geq \beta$.

Additional features It can be noticed that fuzzy DLs present other features that are specifically adapted to the fuzzy logic framework and improve its expressiveness. The "∀" and "∃" quantifiers have for instance been generalised to fuzzy quantifiers such as "most", "some", or "usually", leading to concepts like "most R.C" [17]. Besides fuzzy modifiers were proposed like "very" to define new concepts from atomic ones, as for instance "very C" [5, 15].

2.3 Possibilistic description logics

The possibilistic logic was introduced as a logical framework to handle uncertain information and perform reasoning from it [11]. Its application to description logics was first introduced by [3] and recently further studied by [4, 18].

Syntax In possibilistic DLs a formula is a couple of the type (ϕ, N, α) or (ϕ, Π, α), depending on whether it is necessarily or possibly valued; ϕ denotes a classic DLs formula, and α is a numerical value belonging to the interval $[0, 1]$.

For necessarily valued formulas, the coefficient represents a lower bound on the necessity degree of ϕ, i.e. a constraint of the form $N(\phi) \geq \alpha$: the higher α, the more certain the formula. For possibly valued formulas, the coefficient is an upper bound on ϕ possibility degree, i.e. it expresses the constraint $\Pi(\phi) \leq \alpha$. Formulas subject to constraints like $N(\phi) \leq \alpha$ or $\Pi(\phi) \geq \beta$ are not considered, because they do not provide enough information [19].

It is to be noted that $\Pi(\phi) = 1 - N(\neg\phi)$, and thus (ϕ, Π, β) is equivalent to $(\neg\phi, N, 1-\beta)$. Thus in the following, without loss of generality, we only consider necessity-valued formulas, and simplify the notations to (ϕ, α).

Semantics The semantics associated with possibilistic DLs are formally very different from the semantics associated with the fuzzy case: they rely on the definition of a possibilistic distribution over the set of (classic) interpretations [19, 3, 4]. Denoting Ω this set of interpretations, a possibility distribution on Ω is a function $\pi : \Omega \rightarrow [0, 1]$ that associates each interpretation with its possibility degree. It must moreover attain the value 1 for at least one interpretation. π induces a possibility and a necessity measure for the formulas, defined as $\Pi(\phi) = \sup\{\pi(\omega) : \omega \models \phi\}$ and $N(\phi) = 1 - \Pi(\neg\phi)$.

It can be underlined that, contrary to the fuzzy case, there is no room for selecting operators in the possibilistic case.

Satisfiability and consistency A possibility distribution π defined on Ω then satisfies a formula (ϕ, N, α) if the necessity degree it implies, $N(\phi)$, verifies $N(\phi) \geq \alpha$. A set of possibilistic formulas is consistent if there exists a possibility distribution that verifies all its formulas.

Moreover, in the case of possibilistic logic, the notion of consistency is extended to the definition of a level of inconsistency [19], defined as

$$\text{inc}(\mathcal{F}) = \max\{\alpha | \mathcal{F} \vdash (\bot, \alpha)\}$$

for a given set of formulas $\mathcal{F}$, where $\bot$ denotes the contradiction and $\vdash$ the syntatic deduction. If $\text{inc}(\mathcal{F}) = 0$, the knowledge base is consistent in the classic sense, as there is no certainty about contradiction; if it is 1, contradiction is certain, and the knowledge base is inconsistent in the usual sense.

Contradictory pairs For possibilistic logic, contradiction is derived from pairs of the form

$$\mathcal{F} = \{(\phi, \alpha), (\neg\phi, \beta)\}$$

Indeed from such a pair, using the definition of $\wedge$ in possibilistic logic, one can derive $\mathcal{F} \vdash (\phi \wedge \neg\phi, \min(\alpha, \beta))$ and thus $\mathcal{F} \vdash (\bot, \min(\alpha, \beta))$. This derivation implies that the inconsistency degree associated with $\mathcal{F}$ equals $\min(\alpha, \beta)$.

It is to be noted that in the fuzzy framework, such a pair is not necessary contradictory: interpreting the necessity as an upper bound on the coefficient, the corresponding formulas in the fuzzy framework are $(\phi \geq \alpha)$ and $(\neg\phi \geq \beta)$. Now $(\neg\phi \geq \beta) \iff (\phi < 1 - \beta)$ which is consistent with $(\phi \geq \alpha)$ provided $\alpha < 1 - \beta$, i.e. $\alpha + \beta < 1$.

On the contrary, a contradictory fuzzy DL pair, $\{(\phi \geq \alpha), (\phi < \beta)\}$ with $\alpha \geq \beta$, is also inconsistent when interpreted in a possibilistic sense: considering upper bounds as necessity constraints and lower bounds as possibility ones, the pair corresponds to $N(\phi) \geq \alpha$ and $\Pi(\phi) < \beta$. The latter implies that $N(\neg\phi) \geq 1 - \beta$, which leads to an inconsistency when combined with $N(\phi) \geq \alpha$, with degree $\min(\alpha, 1 - \beta)$. It is to be noted that in the case of the fuzzy framework, both α and β belong to the same scale of values whereas in the possibilistic one, α belongs to the scale of necessity values and β belongs to the scale of possibility values.

3 Algorithms for consistency checking for extended DLs

3.1 Principles

Given a knowledge base in the DL framework or in one of its extensions, the objective is to derive new information that are implicitly contained in the knowledge base. Several types of services can be considered, as for instance subsumption, instantiation or role verification (see [9] for more details). Now these services can be reduced as a consistency issue, applied to the knowledge base to which some formulas are added. The fundamental reasoning service is therefore inconsistency checking.

Tableau algorithms For classic DLs, consistency checking is usually performed by the tableau algorithms [20]: this family of methods, whose variants depend on the expressiveness of the considered connectives, basically consists in decomposing the formulas of the knowledge base into simpler ones, using completion rules, i.e. consistency preserving transformation rules. The obtained transformed knowledge bases are then tested for the presence of clashes. A clash is defined as a contradiction, coming from the presence in a knowledge base of a formula and its negation, i.e. of a pair $\{\phi, \neg\phi\}$ where ϕ is any DL formula. Indeed such a pair shows that the knowledge base is not consistent. A more detailed description is provided in [20].

Consistency checking for extended DLs For the various extensions of description logics, the problem of consistency checking must also be solved. As mentioned in the previous section, the definition of consistency differs from one extension to the other, several algorithms have thus been defined. For each extension, various solutions have been proposed, differing in the considered assumptions (e.g. expressiveness of the considered language, acyclic concept definitions), in the algorithmic expression, or even the kind of the result (some algorithms provide a binary decision concerning consistency, others compute a numerical coefficient corresponding to an inconsistency degree).

We propose a classification distinguishing between three types of approaches, from an algorithmic point of view: the first one relies on the application of the classic tableau algorithm to crisp knowledge bases, derived from the extended ones through an α-cut transformation. The second type of methods transposes the completion rules to the numerical domain, associating each completion rule with an inequality constraint on numerical coefficients. The third one provides a direct extension of the tableau algorithm, so that it can handle numerical coefficients. This third approach has only been applied in the fuzzy case, we describe in Section 4 a consistency checking algorithm for possibilistic DLs following its principle.

3.2 Approach based on α-cuts

The α-cut-based approach [3, 4, 18] consists in exploiting the classic tableau algorithm and applying it to crisp knowledge bases derived from the initial base. It has been applied to the possibilistic DL extension and is based on the α-cuts of the possibilistic base. Its theoretical foundations were studied by Hollunder [3] who developed a new proof method for possibilistic logic and applied it to descriptive logics. It relies on the following theorem [3]: given a possibilistic knowledge base Φ, $\alpha \in [0,1]$ and $\Phi_\alpha = \{\phi \mid (\phi, \alpha') \in \Phi \wedge \alpha' \geq \alpha\}$ the α-cut of Φ, the following equivalence holds:

$$\Phi \models (\phi, N, \alpha) \qquad \Longleftrightarrow \qquad \Phi_\alpha \models \phi$$

Hollunder [3] thus proposes to perform a dichotomic search to determine the highest α value so that Φ_α is inconsistent: this not only indicates whether the knowledge base is consistent or not, it also provides a numerical inconsistency degree. Qi *et al.* [4, 18] extend this principle to address the entailment problem. Moreover, they provide a linear order variant of their algorithm that makes it possible to take into account formulas whose certainty degrees are higher than the inconsistency degree to perform deduction.

3.3 Approach based on inequality constraints

The approach based on inequality constraints was first introduced by [15] in the case of fuzzy DLs, and was later generalised by Haarslev *et al.* [21, 22] to a general framework that can handle probabilistic, possibilistic, as well as fuzzy DLs.

The underlying principle is to introduce a numerical variable for each formula derived when applying the completion rules, retaining the constraints it is submitted to: the transformation of a formula leads, on one hand, to the creation of a new formula associated with a variable corresponding to its coefficient, on the other hand to the storage of the numerical constraints imposed on this coefficient in a constraint set. For instance, if $(a : C \sqcap D, \alpha)$ is considered, two formulas $(a : C, x)$ and $(a : D, y)$ are added. In the constraint set, one adds $\min(x, y) = \alpha$. The constraint expressions are derived from the properties of the connectives in the considered DL extension.

If the transformed set of formulas contains a contradiction, the knowledge base is inconsistent; otherwise, an inequality solver is applied to solve the set of constraints: the knowledge base is consistent if and only if a solution can be found.

Tresp and Molitor [15] provide completion rules of this form for the fuzzy extension of the $\mathcal{ALC}$ language, additionally considering fuzzy modifiers for the concepts, and proposing the corresponding completion rules. In the general framework presented by Haarslev *et al.* [21, 22], the values associated with the formula can even be intervals. Abstract rules are defined, that can be instantiated according to the aggregation functions chosen for the considered connectives and thus specialised to the considered extension framework.

This inequality constraint method leads to binary results, indicating whether the knowledge base is consistent or not. It can also be applied to the entailment problem, providing the tightest bound on the degree with which a considered formula can be derived from a given knowledge base.

3.4 Direct extension of the tableau algorithm

The third approach performs a direct extension of the tableau algorithm, to directly associate the infered formulas with their coefficients. It has been applied to fuzzy extensions of DLs, such as fuzzy $\mathcal{ALC}$ [6] and fuzzy $\mathcal{SHIN}$ [8]. This approach requires the definition of clashes and completion rules that integrate the numerical coefficients associated with each formula.

The definition of clashes does not depend on the language expressiveness, but only on the underlying logic. In the case of fuzzy DLs, considered both by [6] and [8], they are defined as the contradictory pairs presented in Section 2.2.

The completion rules must be defined and justified for each connective [6]: in the case of conjunction for instance, when considering a formula with upper bound on the coefficient, $(a : C \sqcap D \geq \alpha)$, the two formulas $(a : C \geq \alpha)$ and $(a : D \geq \alpha)$ are added to the knowledge base [6].

4 Possibilistic direct extension of the tableau algorithm

The direct extension of the tableau algorithm described in the previous subsection has no counterpart in the case of possibilistic DLs. In this section, we present such an extension for the case of the $\mathcal{ALC}$ (see also [23] for more algorithmic details). It relies on the proposition of new definitions for the clash notion and for the completion rules, adapted to the possibilistic framework. One of its characteristic feature is the fact that it does not only determine whether the knowledge base is consistent or not, but also computes an inconsistency degree that quantifies the level of inconsistency of the knowledge base.

4.1 Possibilistic clash definition

Regarding the definition of clash, we simply use the possibilistic contradictory pairs presented in Section 2.3: a clash occurs if the transformed knowledge base contains a pair of the form $\mathcal{F} = \{(\phi, \alpha), (\neg\phi, \beta)\}$

The inconsistency degree is then defined as $\min(\alpha, \beta)$, in agreement with the possibilistic logic theory (see Section 2.3).

4.2 Possibilistic completion rules

The proposed completion rules are extensions of the classic rules [20] taking into account necessity degrees. For each connective we present the associated rule and justify it. In the case of the $\sqcap$ and $\sqcup$-rules we mathematically demonstrate the derivations, we only give an idea of the demonstration for the $\forall$ and $\exists$-rules.

In all cases infered formulas are added to the knowledge base. It is to be noted that they are only conditionally added, in order to avoid duplicating rules in the knowledge base, as in the classic tableau algorithm: if a formula (ϕ, α) is to be added and if ϕ already occurs in the knowledge base, say associated with coefficient β, then a single exemplar of ϕ is maintained, namely the one with the highest necessity degree. Indeed, according to possibilistic logic [19], the following weight weakening property holds: for $\alpha \geq \beta, (\phi, \alpha) \vdash (\phi, \beta)$ Thus maintaining two copies of ϕ with different weights would be redundant as the one with lower weight can be deduced from the other.

$\sqcap$-rule : If $\mathcal{B}$ contains $(a : C \sqcap D, \alpha)$, then add both $\{(a : C, \alpha), (a : D, \alpha)\}$

The proof of this rule validity must show that the consistency of the modified knowledge base is the same as the initial base, i.e. that $(a : C \sqcap D, \alpha) \iff \{(a : C, \alpha), (a : D, \alpha)\}$. Now from the definition of the concept conjunction in description logics, it can be proved that $a : C \sqcap D \iff (a : C) \wedge (a : D)$. Besides, according to the definition of logical "and" in possibilistic logic [19], for any formulas ϕ and ψ, one has $N(\phi \wedge \psi) = \min(N(\phi), N(\psi))$ Applying this property to $\phi = a : C$ and $\psi = a : D$ leads to

$$N(a : C \sqcap D) \geq \alpha \iff \begin{cases} N(a : C) \geq \alpha \\ N(a : D) \geq \alpha \end{cases}$$

which is equivalent to $(a : C \sqcap D, \alpha) \iff \{(a : C, \alpha), (a : D, \alpha)\}$ according to the syntax of possibilistic logic.

$\sqcup$-rule : If $\mathcal{B}$ contains $(a : C \sqcup D, \alpha)$, then add $\{(a : C, \alpha)\}$ or $\{(a : D, \alpha)\}$

A disjunction formula provides by far less information than a conjunction one, due to the fact that the definition of the possibilistic logical "or" only states $N(\phi \vee \psi) \geq \max(N(\phi), N(\psi))$ and there is no equality relation between $N(\phi \vee \psi)$, $N(\phi)$ and $N(\psi)$.

Still, denoting $K = \{(\phi \vee \psi, \alpha)\} \cup \mathcal{B}$, $K_1 = \{(\phi, \alpha)\} \cup \mathcal{B}$ and $K_2 = \{(\psi, \alpha)\} \cup \mathcal{B}$, it holds that

$$\text{inc}(K) = \min(\text{inc}(K_1), \text{inc}(K_2)) \tag{1}$$

This follows from the fact that, for any interpretation ω, the possibility distributions induced by K, K_1 and K_2 respectively (as defined by [19]) verify $\pi_K(\omega) = \max(\pi_{K_1}(\omega), \pi_{K_2}(\omega))$, which can be shown by case disjunction on whether ω is a model of ϕ, ψ, both of them, or none of them.

Equation (1) proves that the determination of the inconsistency degree of K can be derived from that of K_1 and K_2: as in the case of the classic tableau algorithm, the rule proposed for the $\sqcup$ connective is a non-deterministic one, that requires to examine two knowledge bases alternatively.

Thus, even if no equality relation between $N(\phi \vee \psi)$, $N(\phi)$ and $N(\psi)$ exists, one can derive information from a disjunctive formula: the possibility distribution induced by a knowledge base follows the minimum specificity principle [11], i.e. corresponds to the least specific possibility distribution compatible with the formulas. Reasoning with this distribution is equivalent to the "most cautious" choice, the one that introduces the least unjustified additional information. It suffices for the determination of the inconsistency degree.

$\forall$-rule : If $\mathcal{B}$ contains $(a : \forall R.C, \alpha)$, then add $\{(b : C, \gamma_b)\}$ for all b such that $\mathcal{B}$ contains $((a, b) : R, \beta)$, with $\gamma_b = \min(\alpha, \beta)$.

To justify this rule, we first remind that $F = (a : \forall R.C)$ corresponds, in the classic logic syntax, to $\forall x(R(a, x) \rightarrow C(x))$. Now from the definition of modus ponens in possibilistic logic [19], $(\phi \rightarrow \psi, \alpha), (\phi, \beta) \vdash (\psi, \min(\alpha, \beta))$. Applied to $\phi = R(a, b)$ and $\psi = C(b)$ this implies that, for all b occurring in formulas of the type $(R(a,b),\beta)$, the formula $(C(b),\min(\alpha,\beta))$ can be derived. Reciprocally, if all such formulas are added, then F is verified, a model of the modified base is also a model of the initial base. This justifies the proposed rule.

∃-rule If $\mathcal{B}$ contains $(a : \exists R.C, \alpha)$ and if there is no b such that $\mathcal{B}$ contains both $(b : C, \beta)$ and $((a,b) : R, \gamma)$ with $\min(\beta,\gamma) \geq \alpha$, then add $\{(b : C, \alpha), ((a,b) : R, \alpha)\}$ where b is a constant that does not occur in $\mathcal{B}$.

For the ∃-rule justification, we consider that the knowledge base $\mathcal{B}$ contains the formula $F = (\phi, \alpha)$, with $\phi = a : \exists R.C$. In the classic logic syntax ϕ is written $\exists b(R(a,b) \wedge C(b))$.

Now if $\mathcal{B}$ contains two formulas of the type $F_1 = ((a,c) : R, \beta)$ and $F_2 = (c : C, \gamma)$ with $\beta \geq \alpha$ and $\gamma \geq \alpha$, then F is redundant with F_1 and F_2 because their conjunction implies $(\phi, \min(\beta,\gamma))$, which in turn implies F due to the weight weakening property. Thus F can be suppressed as a redundant formula.

Otherwise, i.e. if $\mathcal{B}$ does not contain two such formulas, or if they are associated with degrees such that $\min(\beta,\gamma) < \alpha$, then new formulas must be added to rewrite the information contained in F. This requires to introduce a new individual, that does not occur in the current knowledge base, say b, and to add the two formulas $F_1 = ((a,b) : R, \alpha)$ and $F_2 = (b : C, \alpha)$.

5 Conclusion

We considered in this paper the problem of consistency checking for extended description logics, providing an overview and a classification of existing approaches and commenting on the differences between the fuzzy and possibilistic DL formalisms. We also described a new method based on a direct extension of the tableau algorithm for which we provided and justified completion rules directly taking into account necessity degrees. This method yields an inconsistency degree characterising the knowledge base more finely than a binary consistency decision.

Ongoing works concern the implementation of the provided method to allow for a study of its computational performances. At a theoretical level, perspectives aim at studying soundness, completeness and complexity issues and extending the provided rules to more expressive possibilistic DLs, for instance taking into account numerical constraints on the roles.

References

1. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Proc. of the 4th Int. Conf. on Principles of Knowledge Representation and Reasoning. (1994)
2. Lukasiewicz, T., Straccia, U.: Description logic programs under probabilistic uncertainty and fuzzy vagueness. In: Proc. of ECSQARU'07. (2007) 187–198
3. Hollunder, B.: An alternative proof method for possibilistic logic and its application to terminological logics. Int. Journal of Approximate Reasoning **12** (1995) 85–109

4. Qi, G., Pan, J.Z., Ji, Q.: Extending description logics with uncertainty reasoning in possibilistic logic. In: Proc. of ECSQARU'07. (2007) 828–839
5. Yen, J.: Generalizing term subsumption languages to fuzzy logic. In: Proc of IJCAI'91. (1991) 472–477
6. Straccia, U.: Reasoning within fuzzy description logics. JAIR **14** (2001) 137–166
7. Hájek, P.: Making fuzzy description logic more general. Fuzzy Sets and Systems **154** (2005) 1–15
8. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with Very Expressive Fuzzy Description Logics. JAIR **30** (2007) 273–320
9. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook Theory, Implementation and Applications. Cambridge University Press (2003)
10. Zadeh, L.A.: Fuzzy logic and its application to approximate reasoning. In: IFIP Congress. (1974) 591–594
11. Dubois, D., Prade, H.: Basic notions of possibility theory. In Ruspini, E.H., Bonissone, P.P., Pedrycz, W., eds.: Handbook of Fuzzy Computation. IOP Publishing Ltd (1998)
12. Dubois, D., Prade, H.: An introduction to possibilistic and fuzzy logics. In Shafer, G., Pearl, J., eds.: Readings in uncertain reasoning. Morgan Kaufmann Publishers Inc. (1990) 742–761
13. Meghini, C., Sebastiani, F., Straccia, U.: Reasoning about the form and content for multimedia objects (extended abstract). In: Proc. of AAAI'97. (1997)
14. Straccia, U.: A fuzzy description logic. In: Proc. of AAAI'98. (1998) 594–599
15. Tresp, C.B., Molitor, R.: A description logic for vague knowledge. In: Proc. of ECAI'98. (1998) 361–365
16. Straccia, U.: Description logics with fuzzy concrete domains. In: Proc. of UAI'05. (2005) 559–567
17. Sánchez, D., Tettamanzi, A.G.B.: Generalizing quantification in fuzzy description logics. In: Computational Intelligence, Theory and Applications. Proc. of the 8th Fuzzy Days in Dortmund. (2004) 397–412
18. Qi, G., Pan, J.Z., Ji, Q.: A possibilistic extension of description logics. In: Proc. of DL'07. (2007)
19. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In Gabbay, D., Hogger, C.J., Robinson, J.A., eds.: Handbook of Logic in Artificial Intelligence and Logic Programming. Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning. Oxford University Press (1994) 439–513
20. Baader, F., Sattler, U.: An overview of tableau algorithms for description logics. Studia Logica **69** (2001) 5–40
21. Haarslev, V., Pai, H.I., Shiri, N.: Completion rules for uncertainty reasoning with the description logic $\mathcal{ALC}$. In: Proc. of the Canadian Semantic Web Working Symp. Volume 4. (2006) 205–225
22. Haarslev, V., Pai, H.I., Shiri, N.: Optimizing tableau reasoning in $\mathcal{ALC}$ extended with uncertainty. In: Proc. of DL'07. (2007) 307–314
23. Lesot, M.J., Couchariere, O., Bouchon-Meunier, B., Rogier, J.L.: Inconsistency degree computation for possibilistic description logic: an extension of the tableau algorithm. In: Proc. of NAFIPS'08. (2008)