# Extending Description Logics with Uncertainty Reasoning in Possibilistic Logic

Guilin Qi[1], Jeff Z. Pan[2], and Qiu Ji[1]

[1] Institute AIFB, University of Karlsruhe, Germany
{gqi,qiji}@aifb.uni-karlsruhe.de
[2] Department of Computing Science, The University of Aberdeen
Aberdeen AB24 3FX
jpan@csd.abdn.ac.uk

**Abstract.** Possibilistic logic provides a convenient tool for dealing with inconsistency and handling uncertainty. In this paper, we propose possibilistic description logics as an extension of description logics. We give semantics and syntax of possibilistic description logics. We then define two inference services in possibilistic description logics. Since possibilistic inference suffers from the *drowning problem*, we consider a drowning-free variant of possibilistic inference, called linear order inference. Finally, we implement the algorithms for inference services in possibilistic description logics using KAON2 reasoner.

## 1 Introduction

Dealing with uncertainty in the Semantic Web has been recognized as an important problem in the recent decades. Two important classes of languages for representing uncertainty are probabilistic logic and possibilistic logic. Arguably, another important class of language for representing uncertainty is fuzzy set theory or fuzzy logic. Many approaches have been proposed to extend description logics with probabilistic reasoning, such as approaches reported in [12,10]. The work on fuzzy extension of ontology languages has also received a lot of attention (e.g., [18,17]). By contrast, there is relatively few work on combining possibilistic logic and description logic.

Possibilistic logic [5] or possibility theory offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. It is very powerful to represent partial or incomplete knowledge [4]. There are two different kinds of possibility theory: one is qualitative and the other is quantitative. Qualitative possibility theory is closely related to default theories and belief revision [7,3] while quantitative possibility can be related to probability theory and can be viewed as a special case of belief function [8].

The application of possibilistic logic to deal with uncertainty in the Semantic Web is first studied in [13] and is then discussed in [6]. When we obtain an ontology using ontology learning techniques, the axioms of the ontology are often attached with confidence degrees and the learned ontology may be inconsistent

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 828–839, 2007.

**Table 1.** Semantics of $\mathcal{ALC}$-concepts

| Constructor | Syntax | Semantics |
|---|---|---|
| top | $\top$ | $\Delta^{\mathcal{I}}$ |
| bottom | $\bot$ | $\emptyset$ |
| concept name | $\mathsf{CN}$ | $\mathsf{CN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ |
| general negation ($\mathcal{C}$) | $\neg C$ | $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$ |
| conjunction | $C \sqcap D$ | $C^{\mathcal{I}} \cap D^{\mathcal{I}}$ |
| disjunction ($\mathcal{U}$) | $C \sqcup D$ | $C^{\mathcal{I}} \cup D^{\mathcal{I}}$ |
| exists restriction ($\mathcal{E}$) | $\exists R.C$ | $\{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ |
| value restriction | $\forall R.C$ | $\{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$ |

[11]. In this case, possibilistic logic provides a flexible framework to interpret the confidence values and to reason with the inconsistent ontology under uncertainty.

However, there exist problems which need further discussion. First, there is no formal definition of the semantics of possibilistic description logics. The semantic extension of possibilistic description logic is not trivial because we need negation of axioms to define the *necessity measure* from a *possibility distribution*. However, negation of axioms are not allowed in description logics. Second, there is no implementation of possibilistic inference in description logics.

In this paper, we present a possibilistic extension of description logics. We first give the syntax and semantics of possibilistic logics. We then define two inference services in possibilistic description logics. Since possibilistic inference suffers from the *drowning problem*, we consider a drowning-free variant of possibilistic inference, called linear order inference. Finally, we implement the algorithms for inference services in possibilistic description logics using KAON2 reasoner.

The rest of this paper proceeds as follows. Preliminaries on possibilistic logic and description logics are given in Section 2. Both syntax and semantics of possibilistic description logics are provided in Section 3. The inference services in possibilistic description logics are also given. After that, we provide algorithms for implementing reasoning problems in Section 4. Finally, we report preliminary results on implementation in Section 5.

## 2 Preliminaries

### 2.1 Description Logics

Due to the limitation of space, we do not provide a detailed introduction of Description Logics (DLs), but rather point the reader to [1]. A DL knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ consists a set $\mathcal{T}$ (TBox) of concepts axioms[1] and a set $\mathcal{A}$ (ABox) of individual axioms. Concept axioms have the form $C \sqsubseteq D$ where $C$ and $D$ are (possibly complex) concept descriptions. The ABox contains *concept assertions* of the form $a : C$ where $C$ is a concept and $a$ is an individual name, and *role*

[1] TBox could contain some role axioms, for some expressive DLs such as $\mathcal{SHOIQ}$ [14].

*assertions* of the form $\langle a, b\rangle : R$, where $R$ is a role, and $a$ and $b$ are individual names. A *concept description* (or simply *concept*) of the smallest propositionally closed DL $\mathcal{ALC}$ is defined by the following syntactic rules, where $\mathsf{CN}$ is a concept name, $R$ is a role, $C$, $C_1$ and $C_2$ are concept descriptions:

$$\top \mid \bot \mid \mathsf{CN} \mid \neg C_1 \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C \mid \forall R.C.$$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of the domain of the interpretation $\Delta^{\mathcal{I}}$ (a non-empty set) and the interpretation function $\cdot^{\mathcal{I}}$, which maps each concept name $\mathsf{CN}$ to a set $\mathsf{CN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role name $\mathsf{RN}$ to a binary relation $\mathsf{RN}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual $a$ to an object in the domain $a^{\mathcal{I}}$. The interpretation function can be extended to give semantics to concept descriptions (see Table 1). An interpretation $\mathcal{I}$ *satisfies* a concept axiom $C \sqsubseteq D$ (a concept assertion $a : C$ and a role assertion $\langle a, b\rangle : R$, resp.) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$ resp.). An interpretation $\mathcal{I}$ *satisfies* a knowledge base $\Sigma$ if it satisfies all axioms in $\Sigma$; in this case, we say $\mathcal{I}$ is an interpretation of $\Sigma$. A knowledge base is *consistent* if it has an interpretation. A concept is unsatisfiable in $\Sigma$ iff it is interpreted as an empty set by all the interpretation of $\Sigma$.

Most DLs are fragments of classical first-order predicate logic (FOL). An $\mathcal{ALC}$ knowledge bases can be translated to a $\mathcal{L}^2$ (the decidable fragment of FOL with no function symbols and only 2 variables [16]) theory. For example, the concept axiom $C \sqsubseteq D \sqcap \exists R.E$ can be translated into the following $\mathcal{L}^2$ axiom: $\forall x(\phi_C(x) \rightarrow \phi_D(x) \wedge \exists y(\phi_R(x, y) \wedge \phi_E(y)))$, where $\phi_C, \phi_D, \phi_E$ are unary predicates and $\phi_R$ is a binary predicate.

### 2.2 Possibilistic Logic

Possibilistic logic [5] is a weighted logic where each classical logic formula is associated with a number in $(0, 1]$. Semantically, the most basic and important notion is *possibility distribution* $\pi$: $\Omega \rightarrow [0, 1]$, where $\Omega$ is the set of all classical interpretations. $\pi(\omega)$ represents the degree of compatibility of interpretation $\omega$ with available beliefs. From *possibility distribution* $\pi$, two measures can be determined, one is the possibility degree of formula $\phi$, defined as $\Pi(\phi) = max\{\pi(\omega) : \omega \models \phi\}$, the other is the necessity or certainty degree of formula $\phi$, defined as $N(\phi) = 1 - \Pi(\neg\phi)$.

At syntactical level, a *possibilistic formula* is a pair $(\phi, \alpha)$ consisting of a classical logic formula $\phi$ and a degree $\alpha$ expressing certainty or priority. A possibilistic knowledge base is the set of possibilistic formulas of the form $B = \{(\phi_i, \alpha_i) : i = 1, ..., n\}$. The classical base associated with $B$, denoted $B^*$, is defined as $B^* = \{\phi_i | (\phi_i, \alpha_i) \in B\}$. A possibilistic knowledge base is consistent iff its classical base is consistent.

Given a possibilistic knowledge base $B$ and $\alpha \in (0, 1]$, the $\alpha$-cut (strict $\alpha$-cut) of $B$ is $B_{\geq\alpha} = \{\phi \in B^* | (\phi, \beta) \in B \text{ and } \beta \geq \alpha\}$ ($B_{>\alpha} = \{\phi \in B^* | (\phi, \beta) \in B \text{ and } \beta > \alpha\}$). The *inconsistency degree* of $B$, denoted $Inc(B)$, is defined as $Inc(B) = max\{\alpha_i : B_{\geq\alpha_i} \text{ is inconsistent}\}$.

There are two possible definitions of inference in possibilistic logic.

**Definition 1.** *Let $B$ be a possibilistic knowledge base.*

- *A formula $\phi$ is said to be a* plausible consequence *of $B$, denoted by $B \vdash_P \phi$, iff $B_{>Inc(B)} \vdash \phi$.*
- *A formula $\phi$ is said to be a* possibilistic consequence *of $B$ to degree $\alpha$, denoted by $B \vdash_\pi (\phi, \alpha)$, iff the following conditions hold: (1) $B_{\geq\alpha}$ is consistent, (2) $B_{\geq\alpha} \vdash \phi$, (3) $\forall \beta > \alpha$, $B_{\geq\beta} \not\vdash \phi$.*

According to Definition 1, an inconsistent possibilistic knowledge base can non-trivially infer conclusion, so it is inconsistency tolerent. However, it suffers from the "drowning problem" [2]. That is, given an inconsistent possibilistic knowledge base $B$, formulas whose certainty degrees are not larger than $Inc(B)$ are completely useless for nontrivial deductions. For instance, let $B = \{(p, 0.9), (\neg p, 0.8), (r, 0.6), (q, 0.7)\}$, it is clear that $B$ is equivalent to $B = \{(p, 0.9), (\neg p, 0.8)\}$ because $Inc(B) = 0.8$. So $(q, 0.7)$ and $(r, 0.6)$ are not used in the possibilistic inference.

Several variants of possibilistic inference have been proposed to avoid the drowning effect. One of them, called linear order inference, is defined as follows.

**Definition 2.** *Let $B = \{(\phi_i, \alpha_i) : i = 1, ..., n\}$ be a possibilistic knowledge base. Suppose $\beta_j$ $(j = 1, ..., k)$ are all distinct weights appearing in $B$ such that $\beta_1 > \beta_2 > ... > \beta_k$. Let $\Sigma_B = (S_1, ..., S_k)$, where $S_i = \{\phi_l : (\phi_l, \alpha_l) \in B, \alpha_l = \beta_i\}$, and $\Sigma_{LO,B} = \bigcup_{i=1}^{k} S_i'$, where $S_i'$ is defined by $S_i' = S_i$ if $S_i \cup \bigcup_{j=1}^{i-1} S_j'$ is consistent, $\emptyset$ otherwise. A formula $\phi$ is said to be a linear consequence of $B$, denoted $B \vdash_{LO} \phi$, iff $\Sigma_{LO,B} \vdash \phi$.*

The linear order approach does not stop at the inconsistency degree of possibilistic knowledge base $B$. It takes into account of formulas whose certainty degrees are less than the inconsistency degree.

# 3 Possibilistic Description Logics

## 3.1 Syntax

The syntax of possibilistic DL is based on the syntax of classical DL. A *possibilistic axiom* is a pair $(\phi, \alpha)$ consisting of an axiom $\phi$ and a weight $\alpha \in (0, 1]$. A *possibilistic TBox* (resp., *ABox*) is a finite set of possibilistic axioms $(\phi, \alpha)$, where $\phi$ is an TBox (resp., ABox) axoim. A possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ consists of a possibilistic TBox $\mathcal{T}$ and a possibilistic ABox $\mathcal{A}$. We use $\mathcal{T}^*$ to denote the classical DL axioms associated with $\mathcal{T}$, i.e., $\mathcal{T}^* = \{\phi_i : (\phi_i, \alpha_i) \in \mathcal{T}\}$ ($\mathcal{A}^*$ can be defined similarly). The classical base $\mathcal{B}^*$ of a possibilistic DL knowledge base is $\mathcal{B}^* = (\mathcal{T}^*, \mathcal{A}^*)$. A possibilistic DL knowledge base $\mathcal{B}$ is inconsistent if and only if $\mathcal{B}^*$ is inconsistent.

Given a possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ and $\alpha \in (0, 1]$, the $\alpha$-cut of $\mathcal{T}$ is $\mathcal{T}_{\geq\alpha} = \{\phi \in \mathcal{B}^* | (\phi, \beta) \in \mathcal{T} \ and \ \beta \geq \alpha\}$ (the $\alpha$-cut of $\mathcal{A}$, denoted as $\mathcal{A}_{\geq\alpha}$, can be defined similarly). The strict $\alpha$-cut of $\mathcal{T}$ (resp., $\mathcal{A}$) can be defined similarly as the strict cut in possibilistic logic. The $\alpha$-cut (resp., strict $\alpha$-cut) of $\mathcal{B}$ is $\mathcal{B}_{\geq\alpha} = (\mathcal{T}_{\geq\alpha}, \mathcal{A}_{\geq\alpha})$ (resp., $\mathcal{B}_{>\alpha} = (\mathcal{T}_{>\alpha}, \mathcal{A}_{>\alpha})$). The *inconsistency degree* of $\mathcal{B}$, denoted $Inc(\mathcal{B})$, is defined as $Inc(\mathcal{B}) = max\{\alpha_i : \mathcal{B}_{\geq\alpha_i} \ is \ inconsistent\}$.

We use the following example as a running example throughout this paper.

*Example 1.* Suppose we have a possibilistic DL knowledge base $\mathcal{B} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} = \{(Eat_{fish} \sqsubseteq Swim, 0.6), (Bird \sqsubseteq Fly, 0.8), (HasWing \sqsubseteq Bird,\ 0.95)\}$ and $\mathcal{A} = \{(Bird\ (chirpy), 1), (HasWing(tweety), 1), (\neg Fly(tweety), 1)\}$. The TBox $\mathcal{T}$ states that it is rather certain that birds can fly and it is almost certain that something with wing is a bird. The ABox $\mathcal{A}$ states that it is certain that tweety has wing and it cannot fly, and chirpy is a bird. Let $\alpha = 0.8$. We then have $\mathcal{B}_{\geq 0.8} = (\mathcal{T}_{\geq 0.8}, \mathcal{A}_{\geq 0.8})$, where $\mathcal{T}_{\geq 0.8} = \{Bird \sqsubseteq Fly, HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{\geq 0.8} = \{HasWing(tweety), \neg Fly(tweety), Bird(chirpy)\}$. It is clear that $\mathcal{B}_{\geq \alpha}$ is inconsistent. Now let $\alpha = 0.95$. Then $\mathcal{B}_{\geq \alpha} = (\mathcal{T}_{\geq 0.95}, \mathcal{A}_{\geq 0.95})$, where $\mathcal{T}_{\geq 0.95} = \{HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{\geq 0.95} = \{HasWing(tweety),\ \neg Fly(tweety), Bird\ (chirpy)\}$. So $\mathcal{B}_{\geq \alpha}$ is consistent. Therefore, $Inc(\mathcal{B}) = 0.8$.

### 3.2 Semantics

The semantics of possibilistic DL is defined by a *possibility distribution* $\pi$ over the set $\mathbf{I}$ of all classical description logic interpretations, i.e., $\pi : \mathbf{I} \rightarrow [0, 1]$. $\pi(I)$ represents the degree of compatibility of interpretation $I$ with available information. For two interpretations $I_1$ and $I_2$, $\pi(I_1) > \pi(I_2)$ means that $I_1$ is preferred to $I_2$ according to the available information. Given a possibility distribution $\pi$, we can define the possibility measure $\Pi$ and necessity measure $N$ as follows: $\Pi(\phi) = max\{\pi(I) : I \in \mathbf{I}, I \models \phi\}$ and $N(\phi) = 1 - max\{\pi(I) : I \not\models \phi\}^2$. Unlike possibilistic logic, the necessary measure cannot be not defined by the possibility measure because the negation of an axiom is not defined in traditional DLs. However, given a DL axiom $\phi$, let us define the negation of $\phi$ as $\neg\phi = \exists(C \sqcap \neg D)$ if $\phi = C \sqsubseteq D$ and $\neg\phi = \neg C(a)$ if $\phi = C(a)$, where $\exists(C \sqcap \neg D)$ is an existence axiom (see the discussion of negation of a DL axiom in [9]), then it is easy to check that $N(\phi) = 1 - \Pi(\neg\phi)$. Given two possibility distributions $\pi$ and $\pi'$, we say that $\pi$ is more specific (or more informative) than $\pi'$ iff $\pi(I) \leq \pi'(I)$ for all $I \in \Omega$. A possibility distribution $\pi$ satisfies a possibilistic axiom $(\phi, \alpha)$, denoted $\pi \models (\phi, \alpha)$, iff $N(\phi) \geq \alpha$. It satisfies a possibilistic DL knowledge base $\mathcal{B}$, denoted $\pi \models \mathcal{B}$, iff it satisfies all the possibilistic axioms in $\mathcal{B}$.

Given a possibilistic DL knowledge base $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, we can define a possibility distribution from it as follows: for all $I \in \mathbf{I}$,

$$\pi_{\mathcal{B}}(I) = \begin{cases} 1 & \text{if } \forall \phi_i \in \mathcal{T}^* \cup \mathcal{A}^*, I \models \phi_i, \\ 1 - max\{\alpha_i | I \not\models \phi_i, (\phi_i, \alpha_i) \in \mathcal{T} \cup \mathcal{A}\} & \text{otherwise.} \end{cases} \quad (1)$$

As in possibilistic logic, we can also show that the possibility distribution defined by Equation 1 is the least specific possibility distribution satisfying $\mathcal{B}$. Let us consider Example 1 again. $I = \langle \Delta^I, \cdot^I \rangle$ is an interpretation, where $\Delta^I = \{tweety, chirpy\}$ and $Bird^I = \{tweety, chirpy\}$, $Fly^I = \{chirpy\}$, and $HasWing^I = \{tweety\}$. It is clear that $I$ satisfies all the axioms except $Bird \sqsubseteq Fly$ (whose weight is 0.8), so $\pi_{\mathcal{B}}(I) = 0.2$.

[2] The definition of necessity measure is pointed out by one of the reviewers.

Let us give some properties of the possibility distribution defined by Equation (1).

**Theorem 1.** *Let $\mathcal{B}$ be a possibilistic DL knowledge base and $\pi_{\mathcal{B}}$ be the possibility distribution obtained by Equation (1). Then $\mathcal{B}$ is consistent if and only if there exists an interpretation $\mathcal{I}$ such that $\pi_{\mathcal{B}}(\mathcal{I}) = 1$.*

**Proposition 1.** *Let $\mathcal{B}$ be a possibilistic DL knowledge base and $\pi_{\mathcal{B}}$ be the possibility distribution obtained by Equation 1. Then $Inc(\mathcal{B}) = 1 - max_{I \in \mathbf{I}} \pi_{\mathcal{B}}(I)$.*

### 3.3 Possibilistic Inference in Possibilistic DLs

We consider the following inference services in possibilistic DLs.

- Instance checking: an individual $a$ is a *plausible* instance of a concept $C$ with respect to a possibilistic DL knowledge base $\mathcal{B}$, written $\mathcal{B} \models_P C(a)$, if $\mathcal{B}_{>Inc(\mathcal{B})} \models C(a)$.
- Instance checking with necessity degree: an individual $a$ is an instance of a concept $C$ to degree $\alpha$ with respect to $\mathcal{B}$, written $\mathcal{B} \models_\pi (C(a), \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C(a)$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C(a)$.
- Instance checking with necessity degree: an individual $a$ is an instance of a concept $C$ to degree $\alpha$ with respect to $\mathcal{B}$, written $\mathcal{B} \models_\pi (C(a), \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C(a)$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C(a)$.
- Subsumption with necessity degree: a concept $C$ is subsumed by a concept $D$ to a degree $\alpha$ with respect to a possibilistic DL knowledge base $\mathcal{B}$, written $\mathcal{B} \models_\pi (C \sqsubseteq D, \alpha)$, if the following conditions hold: (1) $\mathcal{B}_{\geq\alpha}$ is consistent, (2) $\mathcal{B}_{\geq\alpha} \models C \sqsubseteq D$, (3) for all $\beta{>}\alpha$, $\mathcal{B}_{\geq\beta} \not\models C \sqsubseteq D$.

We illustrate the inference services by reconsidering Example 1.

*Example 2.* (Example 1 continued) According to Example 1, we have $Inc(\mathcal{B}) = 0.8$ and $\mathcal{B}_{>0.8} = (\mathcal{T}_{>0.8}, \mathcal{A}_{>0.8})$, where $\mathcal{T}_{>0.8} = \{HasWing \sqsubseteq Bird\}$ and $\mathcal{A}_{>0.8} = \{HasWing(tweety), \neg Fly\,(tweety), Bird(chirpy)\}$. Since $\mathcal{B}_{>0.8} \models Bird(tweety)$, we can infer that *tweety* is plausible to be a bird from $\mathcal{B}$. Furthermore, since $\mathcal{B}_{\geq 0.95} \models Bird(tweety)$ and $\mathcal{B}_{\geq 1} \not\models Bird(tweety)$, we have $\mathcal{B} \models_\pi (Bird(tweety), 0.95)$. That is, we are almost certain that tweety is a bird.

### 3.4 Linear Order Inference in Possibilistic DLs

Possibilistic inference in possibilistic DL inherits the drowning effect of possibilistic inference in possibilistic logic. We adapt and generalize the linear order inference to deal with the drowning problem.

**Definition 3.** *Let $\mathcal{B} = (\mathcal{T}, \mathcal{A})$ be a possibilistic DL knowledge base. Suppose $\beta_j$ ($j = 1, ..., k$) are all distinct weights appearing in $\mathcal{B}$ such that $\beta_1 > \beta_2 > ... > \beta_k$. Let $\mathcal{B}' = \cup \mathcal{T} \cup \mathcal{A}$. Let $\Sigma_{\mathcal{B}} = (\mathcal{S}_1, ..., \mathcal{S}_k)$, where $\mathcal{S}_i = \{(\phi_l, \alpha_l) : (\phi_l, \alpha_l) \in \mathcal{B}', \alpha_l = \beta_i\}$, and $\Sigma_{LO,\mathcal{B}} = \bigcup_{i=1}^{k} \mathcal{S}_i'$, where $\mathcal{S}_i'$ is defined by $\mathcal{S}_i' = \mathcal{S}_i$ if $\mathcal{S}_i \cup \bigcup_{j=1}^{i-1} \mathcal{S}_j'$ is consistent, $\emptyset$ otherwise. Let $\phi$ be a query of the form $C(a)$ or $C \sqsubseteq D$. Then*

**Algorithm 1.** Compute the inconsistency degree

```
Data: B = ⟨T, A⟩, where T ∪ A = {(φ_i, α_i) : α_i ∈ (0,1], i = 1,...,n}, where n is
      the number of axioms in the testing ontology B;
Result: The inconsistency degree d
begin
    b := 0          // b is the begin pointer of the binary search
    m := 0          // m is the middle pointer of the binary search
    d := 0.0        // The initial value of inconsistency degree d is set to be 0.0
    W = Asc(α_1, ..., α_n)
    W(−1) = 0.0 // The special element −1 of W is set to be 0.0
    e := |W| − 1  // e is the end pointer of the binary search
    if B_{≥W(0)} is consistent then
        d:=0.0
    else
        while b ≤ e do
            if b = e then
                return b
            m := ⌜(b + e)/2⌝
            if B_{≥W(m)} is consistent then
                e := m − 1
            else
                b := m + 1
        d := W(b)
end
```

- *$\phi$ is said to be a consequence of $\mathcal{B}$ w.r.t the linear order policy, denoted $\mathcal{B} \vdash_{LO} \phi$, iff $(\Sigma_{LO,\mathcal{B}})^* \vdash \phi$.*
- *$\phi$ is said to be a weighted consequence of $\mathcal{B}$ to a degree $\alpha$ w.r.t the linear order policy, denoted $\mathcal{B} \vdash_{LO} (\phi, \alpha)$, iff $\Sigma_{LO,\mathcal{B}} \vdash_{\pi} (\phi, \alpha)$.*

In Definition 3, we not only define the consequence of a possibilistic DL knowledge base *w.r.t* the linear order policy, but also the weighted consequence of it. The weighted consequence of $\mathcal{B}$ is based on the possibilistic inference.

*Example 3.* (Example 1 continued) Let $\phi = Eat_{fish} \sqsubseteq Swim$. According to Example 2, $\phi$ is not a consequence of $\mathcal{B}$ *w.r.t.* the possibilistic inference. Since $\Sigma_{\mathcal{B}} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4)$, where $\mathcal{S}_1 = \mathcal{A}$, $\mathcal{S}_2 = \{(HasWing \sqsubseteq Bird, 0.95)\}$, $\mathcal{S}_3 = \{(Bird \sqsubseteq Fly, 0.8)\}$ and $\mathcal{S}_4 = \{(Eat_{fish} \sqsubseteq Swim, 0.6)\}$, we have $\Sigma_{LO,\mathcal{B}} = S_1 \cup S_2 \cup S_4$. It is easy to check that $\mathcal{B} \vdash_{LO} (Eat_{fish} \sqsubseteq Swim, 0.6)$.

## 4 Algorithms for Inference in Possibilistic DLs

We give algorithms for the inference in possibilistic DLs.

Algorithm 1 computes the inconsistency degree of a possibilistic DL knowledge base using a binary search. The function *Asc* takes a finite set of numbers in $(0, 1]$ as input and returns a vector which contains those distinct numbers in the

**Algorithm 2.** Possibilistic inference with certainty degrees

**Data**: $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \cup \mathcal{A} = \{(\phi_i, \alpha_i) : \alpha_i \in (0, 1], i = 1, ..., n\}$; a DL axiom $\phi$.
**Result**: The certainty degree $w$ associated with a query $\phi$
**begin**
  $m := 0$
  $w := 0.0$ // The initial certainty degree of $\phi$ is set to be 0.0
  $W = Asc(\alpha_1, ..., \alpha_n)$
  $W(-1) = 0.0$
  $e := |W| - 1$
  compute $l$ such that $W(l) = Inc(\mathcal{B})$ //$Inc(\mathcal{B})$ is computed by Algorithm 1
  $b := l + 1$
  **if** $\mathcal{B}_{\geq W(b)} \models \phi$ **then**
    **while** $b \leq e$ **do**
      **if** $b = e$ **then**
        return $b$
      $m := \ulcorner (b + e)/2 \urcorner$
      **if** $\mathcal{B}_{\geq W(m)} \not\models \phi$ **then**
        $e := m - 1$
      **else**
        $b := m + 1$
    $w := W(b)$
**end**

set in an ascending order. For example, $Asc(0.2, 0.3, 0.3, 0.1) = (0.1, 0.2, 0.3)$. Let $W = (\beta_1, ..., \beta_n)$ is a vector consisting of $n$ distinct numbers, then $W(i)$ denotes $\beta_i$. If the returned inconsistency degree is 0, that is $W(-1) = 0$, it shows the ontology to be queried is consistent.

Since Algorithm 1 is based on binary search, to compute the inconsistency degree, it is easy to check that the algorithm requires at most $\ulcorner log_2 n \urcorner$+1 satisfiability check using a DL reasoner.

Algorithm 2 returns the necessity degree of an axiom inferred from a possibilistic DL knowledge base *w.r.t* the possibilistic inference. We compute the inconsistency degree of the input ontology. If the axiom is a plausible consequence of a possibilistic DL knowledge base, then we compute its necessity degree using a binary search (see the first "if" condition). Otherwise, its necessity degree is 0, i.e., the default value given to $w$. Note that our algorithm is different from the algorithm given in [15] for computing the necessity of a formula in possibilistic logic (this algorithm needs to compute the negation of a formula, which is computationally hard in DLs according to [9]). We consider only subsumption checking here. However, the algorithm can be easily extended to reduce instance checking as well.

In Algorithm 3, we call Algorithm 1 and Algorithm 2 to compute the certainty degree of the query $\phi$ *w.r.t* the linear order inference. In the "while" loop, the first "if" condition checks if the inconsistency degree is greater than 0 and then

**Algorithm 3.** Linear order inference with certainty degrees

**Data**: $\mathcal{B} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} \cup \mathcal{A} = \{(\phi_i, \alpha_i) : \alpha_i \in (0, 1], i = 1, ..., n\}$; a DL axiom $\phi$.

**Result**: The certainty degree $w$ associated with a query $\phi$

```
begin
    d := 0.0 // The initial inconsistency degree is set to be 0.0
    w := 0.0 // The initial certainty degree of φ is set to be 0.0
    hasAnswer := false
    W = Asc(α_1, ..., α_n)
    e := |W| − 1 // e is a global variable to pass values to the subroutines
    while !hasAnswer do
        if d > 0 then
            e := d − 1
            B := B \ B_{=d}
            W := W \ d
        d := alg1(B), where alg1 is Algorithm 1
        if B_{>d} |= φ then
            hasAnswer := true
        if d ≤ 0 then
            break
    if hasAnswer then
        w := alg2(B, φ), where alg2 is Algorithm 2
end
```

delete the axioms whose necessity degrees are equal to the inconsistency degree. After that, we call Algorithm 1 to compute the inconsistency degree of the initial knowledge base or knowledge base obtained from the first "if" loop. Then the second "if" condition checks if the axiom is a plausible consequence of the possibilistic DL knowledge base and end the "while" loop if the answer is positive. The final "if" condition simply tests if the possibilistic DL knowledge base is consistent or not and terminate the "while" loop if the answer is positive. Finally, we compute the certainty degree of $\phi$ by calling Algorithm 2. This algorithm need to call polynomial times of satisfiability check using a DL reasoner.

Algorithms 2 and 3 compute inference with certainty degree because it is more difficult to obtain the certainty degree of an inferred axiom. They can be easily revised to compute plausible consequence. Because of the page limit, we do not provide the details here.

**Proposition 2.** *Let $\mathcal{B}$ be a possibilistic DL knowledge base and $\phi$ be a DL axiom. Deciding whether $\mathcal{B} \models_P \phi$ requires $\ulcorner log_2 n \urcorner$+1 satisfiability check using a DL reasoner, where n is the number of distinct certainty degrees in $\mathcal{B}$. Furthermore, deciding whether $\mathcal{B} \models_\pi (\phi, \alpha)$ requires at most $\ulcorner log_2 n \urcorner + \ulcorner log_2 n - l \urcorner + 1$ satisfiability check using a DL reasoner, where where n is the number of distinct certainty degrees in $\mathcal{B}$ and l is the inconsistency degree of $\mathcal{B}$.*

## 5 Implementation and Results

To test our algorithms, we have implemented them in Java using KAON2[3]. All tests were performed on a laptop computer with a 1.7GHz Intel processor, 1 GB of RAM, running Windows XP Service Pack 2. Sun's Java 1.5.0 Update 6 is used, and the virtual memory of the Java virtual machine was limited to 800M.

### 5.1 Results

We use ontologies miniTambis[4] and *proton_100_all*[5] as test data. The first ontology contains more than 170 concepts, 35 properties, 172 axioms and 30 unsatisfiable concepts. The second ontology has 175 concepts, 266 properties, 3 unsatisfiable concepts and about 1100 axioms. Both ontologies are consistent but contain some unsatisfiable concepts. We added some instances to the unsatisfiable concepts to make the ontology inconsistent. We get possibilistic DL knowledge bases from miniTambis and *proton_100_all* by randomly attaching certainty degrees to them and using a separate ontology to store the information on the certain degrees. Given a set of certainty degrees $W = (w_1, w_2, ..., w_n),\ w_i \in (0, 1],\ i = 1, ..., n$, an automatic mechanism is applied to randomly choose a certainty degree $w_i$ for each axiom in the ontology to be queried.

In Table 2, some results based on the two ontologies above are given, where $|W|$ means the number of different certainty degrees for testing. The rows corresponding to Algorithm 2 and Algorithm 3 describe the time spending on a specific reasoning task which is instance checking, i.e., the third row shows the time spent by executing Algorithm 2 and the last row is for Algorithm 3. For each column in an ontology, we randomly attach the certainty degrees to axioms in the ontology and give the time spending on a specific reasoning task. Therefore, different columns gives results for different possibilistic DL knowledge bases which may generate from the same ontology.

According to the table, in some cases, the time spent on query by Algorithm 2 and Algorithm 3 is almost the same (see columns 1 and 2 for miniTambis). For example, when the axiom $\phi$ to be queried can be inferred by $\mathcal{B}_{Inc(\mathcal{B})}$. In other cases, it takes much more time for Algorithm 3 to return the result than Algorithm 2. For example, see columns 3 for miniTambis, it takes 2 seconds to

**Table 2.** The results from Algorithm 2 and Algorithm 3

| Ontology | *miniTambis* | | | | | | | | *proton_100_all* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|W\|$ | 10 | | | | 30 | | | | 10 | | | | 30 | | | |
| Algorithm 2 ($s$) | 6 | 8 | 2 | 10 | 16 | 9 | 13 | 8 | 5 | 11 | 5 | 6 | 9 | 12 | 8 | 13 |
| Algorithm 3 ($s$) | 6 | 9 | 12 | 23 | 16 | 8 | 33 | 44 | 5 | 10 | 15 | 12 | 8 | 12 | 19 | 24 |

[3] http://kaon2.semanticweb.org/
[4] http://www.mindswap.org/2005/debugging/ontologies/
[5] http://wasp.cs.vu.nl/knowledgeweb/d2163/learning.html

get result from Algorithm 2 and 12 second from Algorithm 3. This is because Algorithm 2 stops when $\mathcal{B}_{\geq W(d)} \models \phi$ is not satisfied. But for Algorithm 3, it will not stop until $\mathcal{B}_{\geq W(d)} \models \phi$ is satisfied, or no more inconsistency degree can be found.

## 6 Related Work

Our work differs from existing work on extending description logics by possibilistic logic in following points: (1) we provided semantics of the possibilistic description logic, (2) we considered two inference services and give algorithms for computing the consequences of the inference, (3) we proposed a linear order inference which is a drowning-free variant of possibilistic inference and provided algorithm for it, (4) we implemented the proposed algorithm and provided for evaluation results.

Other approaches that extend description logics with uncertainty reasoning are probabilistic description logics [12,10] and fuzzy extension of description logics (e.g., [18,17]). The main difference between possibilistic extension and probabilistic extension lies in the fact that possibilistic logic is a qualitative representation of uncertainty, whilst probabilistic extension is on quantitative aspects of uncertainty. Furthermore, possibilistic DLs can be used to deal with inconsistency and probabilistic DLs are not used for this purpose. Arguably, fuzzy description logics can be used to deal with uncertainty. In possibilistic DLs, the truth value of an axiom is still two-valued, whilst in fuzzy DLs, the truth value of an axiom is multi-valued.

## 7 Conclusions and Future Work

We gave a possibilistic extension of description logics in this paper. We first defined syntax and semantics of possibilistic description logics. Then we consider inference problems in our logics: possibilistic inference and linear order inference. Algorithms were given to check the inference and we implemented the algorithms. As far as we know, this is the first work which discusses how to implement possibilistic description logics. Finally, we report some preliminary but encouraging experimental results.

The algorithms for possibilistic inference of our logics proposed in this paper is independent of DL reasoner. In our future work, we plan to give more efficient reasoning approaches by generalizing the resolution-based reasoning approach for KAON2. Another future work is that we may interpret the concept axioms by possibilistic conditioning and explore the nonmonotonic feature of possibilistic description logics.

*Acknowledgements.* Thank the anonymous referees for their helpful comments. Research presented in this paper was partially supported by the European Commission NeOn (IST-2006-027595, http://www.neon-project.org/), the Knowledge Web projects (IST-2004-507842, http://knowledgeweb.semanticweb. org/), and the

X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

## References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook: Theory, implementation and applications. Cambridge University Press, Cambridge (2003)
2. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. of IJCAI'93, pp. 640–647. Morgan Kaufmann, San Francisco (1993)
3. Benferhat, S., Dubois, D., Prade, H.: Representing default rules in possibilistic logic. In: Proc. of KR'92, pp. 673–684 (1992)
4. Benferhat, S., Lagrue, S., Papini, O.: Reasoning with partially ordered information in a possibilistic logic framework. Fuzzy Sets and Systems 144(1), 25–41 (2004)
5. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, pp. 439–513. Oxford University Press, Oxford (1994)
6. Dubois, D., Mengin, J., Prade, H.: Possibilistic uncertainty and fuzzy features in description logic: A preliminary discussion. In: Capturing Intelligence: Fuzzy Logic and the Semantic WEb, pp. 101–113. Elsevier, Amsterdam (2006)
7. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. Artif. Intell. 50(2), 223–239 (1991)
8. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 169–226 (1998)
9. Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proc. of AAAI'06 (2006)
10. Giugno, R., Lukasiewicz, T.: P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 86–97. Springer, Heidelberg (2002)
11. Haase, P., Völker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In: Proc. of URSW'05, pp. 45–55 (2005)
12. Heinsohn, J.: Probabilistic description logics. In: Proc. of UAI'94, pp. 311–318 (1994)
13. Hollunder, B.: An alternative proof method for possibilistic logic and its application to terminological logics. Int. J. Approx. Reasoning 12(2), 85–109 (1995)
14. Horrocks, I., Sattler, U.: A tableaux decision procedure for SHOIQ. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005) (2005)
15. Lang, J.: Possibilistic logic: complexity and algorithms. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 179–220 (2000)
16. Mortimer, M.: On languages with two variables. Zeitschr. f. math. Logik u. Grundlagen d. Math. 21, 135–140 (1975)
17. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with very expressive fuzzy description logics. J. Artif. Intell. Res. (2007)
18. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. 14, 137–166 (2001)