Tightly Coupled Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web

Thomas Lukasiewicz\textsuperscript{1,2} Umberto Straccia\textsuperscript{3}

Abstract

We present a novel approach to fuzzy description logic programs (or simply fuzzy dl-programs) under the answer set semantics, which is a tight integration of fuzzy disjunctive logic programs under the answer set semantics with fuzzy description logics. From a different perspective, it is a generalization of tightly coupled disjunctive dl-programs by fuzzy vagueness in both the description logic and the logic program component. We show that the new formalism faithfully extends both fuzzy disjunctive logic programs and fuzzy description logics, and that under suitable assumptions, reasoning in the new formalism is decidable. We present a polynomial reduction of certain fuzzy dl-programs to tightly coupled disjunctive dl-programs, and we analyze the complexity of consistency checking and query processing for certain fuzzy dl-programs. Furthermore, we provide a special case of fuzzy dl-programs for which deciding consistency and query processing can both be done in polynomial time in the data complexity.

Key words: tightly coupled fuzzy description logic programs, answer set semantics, Semantic Web, fuzzy disjunctive logic programs, fuzzy description logics, tightly coupled description logic programs, fuzzy vagueness, tractability.

Introduction

The Semantic Web (Berners-Lee, 1999; Fensel et al., 2002) aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind it are to add a machine-readable meaning to Web pages, to use ontologies for a precise definition of shared terms in Web resources, to use KR technology for automated reasoning from

\textsuperscript{1}Computing Laboratory, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK; e-mail: thomas.lukasiewicz@comlab.ox.ac.uk.
\textsuperscript{2}Alternative address: Institut für Informationssysteme, Technische Universität Wien, Favoritenstr. 9-11, 1040 Vienna, Austria; e-mail: lukasiewicz@kr.tuwien.ac.at.
\textsuperscript{3}ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy; e-mail: straccia@isti.cnr.it.
\textsuperscript{*}This paper is a significantly extended and revised version of a paper that appeared in: Proc. RR-2007, LNCS 4524, pp. 289–298. Springer, 2007.
Web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web consists of several hierarchical layers, where the Ontology layer, in form of the OWL Web Ontology Language (W3C, 2004; Horrocks et al., 2003), is currently the highest layer of sufficient maturity. OWL consists of three increasingly expressive sublanguages, namely, OWL Lite, OWL DL, and OWL Full. OWL Lite and OWL DL are essentially very expressive description logics with an RDF syntax (Horrocks et al., 2003). As shown in (Horrocks & Patel-Schneider, 2004), ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $SHIF(D)$ (resp., $SHOIN(D)$). As a next step in the development of the Semantic Web, one aims especially at sophisticated representation and reasoning capabilities for the Rules, Logic, and Proof layers of the Semantic Web.

In particular, there is a large body of work on integrating rules and ontologies, which is a key requirement of the layered architecture of the Semantic Web. Significant research efforts focus on hybrid integrations of rules and ontologies, called description logic programs (or dl-programs), which are of the form $KB = (L, P)$, where $L$ is a description logic knowledge base, and $P$ is a finite set of rules involving either queries to $L$ in a loose integration (see especially (Eiter et al., 2008; Eiter et al., 2004; Eiter et al., 2006)) or concepts and roles from $L$ as unary resp. binary predicates in a tight integration (see especially (Lukasiewicz, 2007a; Rosati, 2006)).

Other works explore formalisms for handling uncertainty and vagueness/imprecision in the Semantic Web. In particular, formalisms for dealing with uncertainty and vagueness in ontologies have been applied in ontology mapping and information retrieval. Vagueness and imprecision also abound in multimedia information processing and retrieval. Moreover, handling vagueness is an important aspect of natural language interfaces to the Web. There are several recent extensions of description logics, ontology languages, and dl-programs for the Semantic Web by probabilistic uncertainty and by fuzzy vagueness. In particular, dl-programs under probabilistic uncertainty and under fuzzy vagueness have been proposed in (Lukasiewicz, 2005; Lukasiewicz, 2006b) and (Straccia, 2006c; Straccia, 2006b; Lukasiewicz, 2006a), respectively.

In this paper, we continue this line of research. We present tightly coupled fuzzy description logic programs (or simply fuzzy dl-programs) under the answer set semantics, which are a tight integration of fuzzy disjunctive programs under the answer set semantics with fuzzy generalizations of $SHIF(D)$ and $SHOIN(D)$. Even though there has been previous work on fuzzy positive dl-programs (Straccia, 2006c; Straccia, 2006b) and on loosely coupled fuzzy normal dl-programs (Lukasiewicz, 2006a), to our knowledge, this is the first approach to tightly coupled fuzzy disjunctive dl-programs (with default negation in rule bodies). The main contributions of this paper can be briefly summarized as follows:

- We present a novel approach to fuzzy dl-programs, which tightly integrates fuzzy disjunctive programs under the answer set semantics with fuzzy description logics. It generalizes the tightly coupled disjunctive dl-programs in (Lukasiewicz, 2007a) by fuzzy vagueness in both the ontological and the rule component.
- We show that the new fuzzy dl-programs have nice semantic features. In particular, all their answer sets are also minimal models, and the cautious answer set
semantics faithfully extends both fuzzy disjunctive programs and fuzzy description logics. The new approach also does not need the unique name assumption.

- As an important property, in the large class of fuzzy dl-programs that are defined over a finite number of truth values, the problems of deciding consistency, cautious consequence, and brave consequence are all decidable.
- We present a polynomial reduction for certain fuzzy dl-programs to the tightly coupled disjunctive dl-programs in (Lukasiewicz, 2007a), and analyze the complexity of consistency checking and query processing for certain fuzzy dl-programs.
- Furthermore, we delineate a special case of fuzzy dl-programs where deciding consistency and query processing are both data tractable.

Combination Strategies

Rather than being restricted to an ordinary binary truth value among false and true, vague propositions may also have a truth value strictly between false and true. In the sequel, we use the unit interval \([0, 1]\) as the set of all possible truth values, where 0 and 1 represent the ordinary binary truth values false and true, respectively. For example, the vague proposition ‘John is a tall man’ may be more or less true, and it is thus associated with a truth value in \([0, 1]\), depending on the body height of John.

In order to combine and modify the truth values in \([0, 1]\), we assume combination strategies, namely, conjunction, disjunction, implication, and negation strategies, denoted \(\otimes, \oplus, \triangleright, \neg\), respectively, which are functions \(\otimes, \oplus, \triangleright : [0, 1] \times [0, 1] \to [0, 1]\) and \(\neg : [0, 1] \to [0, 1]\) that generalize the ordinary Boolean operators \(\land, \lor, \rightarrow, \neg\), respectively, to the set of truth values \([0, 1]\). For \(a, b \in [0, 1]\), we then call \(a \otimes b\) (resp., \(a \oplus b, a \triangleright b\)) the conjunction (resp., disjunction, implication) of \(a\) and \(b\), and we call \(\neg a\) the negation of \(a\).

As usual, we assume that combination strategies have some natural algebraic properties, namely, the ones shown in Tables 1 and 2. Note that conjunction and disjunction strategies (with the properties in Table 1) are also called triangular norms (or t-norms) and triangular co-norms (or s-norms) (Hájek, 1998), respectively. We do not assume properties that relate the combination strategies to each other (such as De Morgan’s law); although one may additionally assume such properties, they are not required here.

**Example 1** The combination strategies of various fuzzy logics are shown in Table 3. Some of their further properties are highlighted in Table 4. Note that we cannot enforce that a choice of the combination strategies satisfies all these properties, because then the fuzzy logic would collapse to classical Boolean propositional logic.

Fuzzy Description Logics

We assume fuzzy generalizations of the expressive crisp description logics \(\mathcal{SHIF}(D)\) and \(\mathcal{SHOIN}(D)\), which stand behind OWL Lite and OWL DL, respectively. We now recall the syntax and the semantics of fuzzy \(\mathcal{SHIF}(D)\) and fuzzy \(\mathcal{SHOIN}(D)\).
Table 1: Axioms for conjunction and disjunction strategies.

<table>
<thead>
<tr>
<th>Axiom Name</th>
<th>Conjunction Strategy</th>
<th>Disjunction Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tautology / Contradiction</td>
<td>( a \otimes 0 = 0 )</td>
<td>( a \oplus 1 = 1 )</td>
</tr>
<tr>
<td>Identity</td>
<td>( a \otimes 1 = a )</td>
<td>( a \oplus 0 = a )</td>
</tr>
<tr>
<td>Commutativity</td>
<td>( a \otimes b = b \otimes a )</td>
<td>( a \oplus b = b \oplus a )</td>
</tr>
<tr>
<td>Associativity</td>
<td>((a \otimes b) \otimes c = a \otimes (b \otimes c))</td>
<td>((a \oplus b) \oplus c = a \oplus (b \oplus c))</td>
</tr>
<tr>
<td>Monotonicity</td>
<td>( \text{if } b \leq c, \text{ then } a \otimes b \leq a \otimes c )</td>
<td>( \text{if } b \leq c, \text{ then } a \oplus b \leq a \oplus c )</td>
</tr>
</tbody>
</table>

Table 2: Axioms for implication and negation strategies.

<table>
<thead>
<tr>
<th>Axiom Name</th>
<th>Implication Strategy</th>
<th>Negation Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tautology / Contradiction</td>
<td>( \neg b = 1 )</td>
<td>( \neg 0 = 1 ), ( \neg 1 = 0 )</td>
</tr>
<tr>
<td>Antitonicity</td>
<td>( \text{if } a \leq b, \text{ then } a \triangleright c \geq b \triangleright c )</td>
<td>( \text{if } a \leq b, \text{ then } \neg a \geq \neg b )</td>
</tr>
<tr>
<td>Monotonicity</td>
<td>( \text{if } b \leq c, \text{ then } a \triangleright b \leq a \triangleright c )</td>
<td>( \text{if } b \leq c, \text{ then } a \triangleright b \leq a \triangleright c )</td>
</tr>
</tbody>
</table>

(Straccia, 2005; Straccia, 2006a) (see also (Stoilos et al., 2005)). For further details and background, see (Łukasiewicz & Straccia, 2008). There also exists an implementation of fuzzy \( SHOF(D) \) under Zadeh, Łukasiewicz, and classical semantics, called the \( fuzzyDL \) system, see (fuzzyDL, 2008; Bobillo & Straccia, 2008a).

Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between individuals, respectively. A description logic knowledge base encodes in particular subset relationships between classes of individuals, subset relationships between binary relations, the membership of individuals to classes, and the membership of pairs of individuals to binary relations. In fuzzy description logics, these relationships and memberships then have a degree of truth in \([0, 1]\).

Syntax

We first define the syntax of fuzzy \( SHOIN(D) \). The elementary ingredients are similar to the ones of crisp \( SHOIN(D) \), except that we now also have fuzzy datatypes and fuzzy modifiers. We assume a set of data values, a set of elementary datatypes, and a set of datatype predicates (each with a predefined arity \( n \geq 1 \)). A datatype is an elementary datatype or a finite set of data values. A fuzzy datatype theory \( D = (\Delta^D, \cdot^D) \) consists of a datatype domain \( \Delta^D \) and a mapping \( \cdot^D \) that assigns to each data value an element of \( \Delta^D \), to each elementary datatype a subset of \( \Delta^D \), and to each datatype predicate of arity \( n \) a fuzzy relation over \( \Delta^D \) of arity \( n \) (that is, a mapping \( (\Delta^D)^n \to [0, 1] \)). We extend \( \cdot^D \) to all datatypes by \( \{v_1, \ldots, v_n\}^D = \{v_1^D, \ldots, v_n^D\} \).

Example 2 A crisp unary datatype predicate \( \preceq_{18} \) over the natural numbers denoting the integers of at most 18 may be defined by \( \preceq_{18}(x) = 1 \), if \( x \leq 18 \), and \( \preceq_{18}(x) = 0 \), otherwise. Then, \( \text{Minor} = \text{Person} \land \exists \text{age.}\preceq_{18} \) defines a person of age at most 18.

Non-crisp predicates are usually defined by functions for specifying fuzzy set membership degrees, such as the well-known trapezoidal, triangular, left-shoulder, and
Table 3: Combination strategies of various fuzzy logics.

<table>
<thead>
<tr>
<th>Łukasiewicz Logic</th>
<th>Gödel Logic</th>
<th>Product Logic</th>
<th>Zadeh Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \otimes b$</td>
<td>$\max(a, b)$</td>
<td>$a \cdot b$</td>
<td>$\min(a, b)$</td>
</tr>
<tr>
<td>$a \oplus b$</td>
<td>$\min(a + b, 1)$</td>
<td>$a + b - a \cdot b$</td>
<td>$\max(a, b)$</td>
</tr>
<tr>
<td>$a \triangleright b$</td>
<td>$\min(1 - a + b, 1)$</td>
<td>$\min(1, b/a)$</td>
<td>$\max(1 - a, b)$</td>
</tr>
<tr>
<td>$\ominus a$</td>
<td>$1 - a$</td>
<td>$\min(1, b/a)$</td>
<td>$1 - a$</td>
</tr>
</tbody>
</table>

Table 4: Further properties of the combination strategies of various fuzzy logics.

<table>
<thead>
<tr>
<th>Property</th>
<th>Łukasiewicz Logic</th>
<th>Gödel Logic</th>
<th>Product Logic</th>
<th>Zadeh Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \otimes a$ = 0</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$a \oplus a$ = 1</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$a \otimes a$ = $a$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$\ominus a$ = $a$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$a \triangleright b = \ominus a \otimes b$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$\ominus (a \triangleright b) = a \otimes b$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$\ominus (a \otimes b) = \ominus a \otimes b$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>$\ominus (a \oplus b) = \ominus a \oplus b$</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

Example 3 A fuzzy unary datatype predicate Young over the natural numbers denoting the degree of youngness of a person’s age may be defined by $\text{Young}(x) = \text{ls}(x; 10, 30)$. Then, $\text{YoungPerson} = \text{Person} \land \exists \text{age}. \text{Young}$ denotes a young person.

Let $A$, $R_A$, $R_D$, $I$, and $M$ be pairwise disjoint sets of atomic concepts, abstract roles, datatype roles, individuals, and fuzzy modifiers, respectively. Here, a fuzzy modifier $m$ (Hölldobler et al., 2004; Tresp & Molitor, 1998) represents a function $f_m$ on $[0, 1]$, which changes the membership function of a fuzzy set.

Example 4 The fuzzy modifiers very resp. slightly may represent the two functions $\text{very}(x) = x^2$ resp. $\text{slightly}(x) = \sqrt{x}$. Then, the concept of sports cars may be defined as $\text{SportsCar} = \text{Car} \land \exists \text{speed. very(High)}$, where High is a fuzzy datatype predicate over the domain of speed in km/h, which may be defined as $\text{High}(x) = \text{rs}(x; 80, 250)$.

Roles and concepts in fuzzy SHOIN(D) are defined in nearly the same way as concepts in SHOIN(D), except that we now also allow fuzzy modifiers from $M$ as unary operators on concepts. A role is any element of $R_A \cup R_A^- \cup R_D$ (where $R_A^-$ is the set of inverses $R^-$ of all $R \in R_A$). We define concepts inductively as follows. Each $A \in A$ is a concept, $\bot$ and $\top$ are concepts, and if $a_1, \ldots, a_n \in I$, then $\{a_1, \ldots, a_n\}$ is a concept (called oneOf). If $C$, $C_1$, $C_2$ are concepts, $R \in R_A \cup R_A^-$, and $m \in M$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, $\neg C$, and $m(C)$ are concepts (called
conjunction, disjunction, negation, and fuzzy modification, respectively), as well as $\exists R.C$, $\forall R.C$, $\geq n R$, and $\leq n R$ (called existential, value, at least, and at most restriction, respectively) for an integer $n \geq 0$. If $D$ is a datatype and $T,T_1,\ldots,T_n \in R_D$, then $\exists T_1,\ldots,T_n.D$, $\forall T_1,\ldots,T_n.D$, $\geq n T$, and $\leq n T$ are concepts (called datatype existential, value, at least, and at most restriction, respectively) for an integer $n \geq 0$. We eliminate parentheses as usual.

A crisp axiom has one of the following forms:

(1) $C \sqsubseteq D$ (called concept inclusion axiom), where $C$ and $D$ are concepts (note that concept inclusion axioms $C \sqsubseteq D$ involve fully general concepts $C$ and $D$);

(2) $R \sqsubseteq S$ (called role inclusion axiom), where either $R,S \in R_A \cup R_A^{-1}$ or $R,S \in R_D$;

(3) $\text{Trans}(R)$ (called transitivity axiom), where $R \in R_A$;

(4) $C(a)$ (called concept assertion axiom), where $C$ is a concept and $a \in I$;

(5) $R(a,b)$ (resp., $T(a,v)$) (called role assertion axiom), where $R \in R_A$ (resp., $T \in R_D$) and $a,b \in I$ (resp., $a \in I$ and $v$ is a data value); and

(6) $a = b$ (resp., $a \neq b$) (equality (resp., inequality) axiom), where $a,b \in I$ (note that the equality (resp., inequality) in equality (resp., inequality) axioms is crisp).

We define fuzzy axioms as follows: A fuzzy concept inclusion (resp., fuzzy role inclusion, fuzzy concept assertion, fuzzy role assertion) axiom is of the form $\alpha \theta n$, where $\alpha$ is a concept inclusion (resp., role inclusion, concept assertion, role assertion) axiom, $\theta \in \{\leq, =, \geq\}$, and $n \in [0,1]$. Informally, $\alpha \leq n$ (resp., $\alpha = n$, $\alpha \geq n$) encodes that the truth value of $\alpha$ is at most (resp., equal to, at least) $n$. We often use $\alpha$ to abbreviate $\alpha = 1$. A fuzzy (description logic) knowledge base $L$ is a finite set of fuzzy axioms, transitivity axioms, and equality and inequality axioms. For decidability, number restrictions in $L$ are restricted to simple abstract roles; informally, such roles are abstract roles with no transitive subroles, where subrole is the reflexive and transitive closure of “$\sqsubseteq$” between roles (see (Horrocks et al., 1999) for details). Notice that $L$ may contain fuzzy concept inclusion axioms (between general concepts).

Fuzzy $\text{St}(\mathcal{F}(D))$ has the same syntax as fuzzy $\text{St}(\mathcal{ON}(D))$, but without the oneOf constructor and with the at least and at most constructors limited to 0 and 1.

**Example 5 (Shopping Agent)** A fuzzy description logic knowledge base $L$ encoding a car selling Web site (such as, e.g., the one at http://www.autos.com) may
contain in particular the following axioms:

\[
\begin{align*}
\text{Cars} & \sqcup \text{Trucks} \sqcup \text{Vans} \sqcup \text{SUVs} \subseteq \text{Vehicles}; \\
\text{PassengerCars} & \sqcup \text{LuxuryCars} \subseteq \text{Cars}; \\
\text{CompactCars} & \sqcup \text{MidSizeCars} \sqcup \text{SportyCars} \subseteq \text{PassengerCars}; \\
\text{Cars} & \sqsubseteq (\exists \text{hasReview}. \text{Integer}) \sqcap (\exists \text{hasInvoice}. \text{Integer}) \\
& \quad \sqcap (\exists \text{hasResellValue}. \text{Integer}) \sqcap (\exists \text{hasMaxSpeed}. \text{Integer}) \\
& \quad \sqcap (\exists \text{hasHorsePower}. \text{Integer}) \ldots ; \\
\text{MazdaMX5Miata} : \text{SportyCars} & \sqcap (\exists \text{hasInvoice}. 18883) \\
& \quad \sqcap (\exists \text{hasHorsePower}. 166) \ldots ; \\
\text{MitsubishiEclipseSpyder} : \text{SportyCars} & \sqcap (\exists \text{hasInvoice}. 24029) \\
& \quad \sqcap (\exists \text{hasHorsePower}. 162) \ldots .
\end{align*}
\]

Eqs. 1–3 describe the concept taxonomy of the site, while Eq. 4 describes the datatype attributes of the cars sold in the site. For example, all cars, trucks, vans, and SUVs are vehicles, and every car is related via the datatype role \text{hasReview} to an integer value. Furthermore, Eqs. 5–6 describe the properties of some sold cars. For example, the \text{MazdaMX5Miata} is a sports car, costs 18 883 €, and has a power of 166 HP.

We may then encode “costs at most about 22 000 €” and “has a power of around 150 HP” in a buyer’s request through the following concepts \(C\) and \(D\), respectively:

\[
C = \exists \text{hasInvoice}. \text{LeqAbout22000} \quad \text{and} \quad D = \exists \text{hasHorsePower}. \text{Around150},
\]

where \text{LeqAbout22000} and \text{Around150} are fuzzy datatype predicates for “at most about 22 000 €” and “around 150 HP”, which are defined by \text{LeqAbout22000}(x) = \text{ls}(x; 22000, 25000) and \text{Around150}(x) = \text{tri}(x; 125, 150, 175) (see Fig. 1), respectively. The former is modeled as a left shoulder function stating that if the prize is less than 22 000 €, then the degree of truth (degree of buyer’s satisfaction) is 1, else the truth is linearly decreasing to 0, reached at 25 000 €. In fact, we are modeling a case were the buyer would like to pay less than 22 000 €, though may still accept a higher price, up to 25 000 €, to a lesser degree. Similarly, the latter models the fuzzy concept “around 150 HP” as a triangular function with vertex in 150 HP.

**Semantics**

We now define the semantics of fuzzy \(SHIF(D)\) and fuzzy \(SHOLN(D)\). The main idea behind it is that concepts and roles are interpreted as fuzzy subsets of an interpretation’s domain. Therefore, rather than being satisfied (true) or unsatisfied (false) in an interpretation, axioms are associated with a degree of truth in \([0, 1]\). In the following, let \(\otimes, \oplus, \triangleright, \sqsubseteq\) be arbitrary but fixed conjunction, disjunction, implication, and negation strategies (see Table 3), respectively.

A fuzzy interpretation \(\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})\) relative to a fuzzy datatype theory \(D = (\Delta^D, \cdot^D)\) consists of a nonempty set \(\Delta^\mathcal{I}\) (called the domain), disjoint from \(\Delta^D\), and of a fuzzy interpretation function \(\cdot^\mathcal{I}\) that coincides with \(\cdot^D\) on every data value, datatype, and fuzzy datatype predicate, and it assigns
(1) to each individual $a \in I$ an element $a^I \in \Delta^I$;
(2) to each atomic concept $A \in A$ a function $A^I : \Delta^I \to [0,1]$;
(3) to each abstract role $R \in R_A$ a function $R^I : \Delta^I \times \Delta^I \to [0,1]$;
(4) to each datatype role $T \in R_D$ a function $T^I : \Delta^I \times \Delta_D \to [0,1]$;
(5) to each fuzzy modifier $m \in M$ the modifier function $m^I = f_m : [0,1] \to [0,1]$.

For properties of such modifier functions, we refer the reader to (Hájek, 1998).

The mapping $T^I$ is extended to all roles and concepts as follows (where $x, y \in \Delta^I$):

\[
(R^I)^T(x, y) = R^T(x, y);
\]
\[
(T^I(x) = 1;)
\]
\[
(\bot^I(x) = 0;)
\]
\[
(a_1, \ldots, a_n)^I(x) = \begin{cases} 1 & \text{if } x \in \{a_1^I, \ldots, a_n^I\}; \\ 0 & \text{otherwise}; \\ \end{cases}
\]
\[
(C_1 \sqcap C_2)^I(x) = C_1^I(x) \otimes C_2^I(x); \\
(C_1 \sqcup C_2)^I(x) = C_1^I(x) \oplus C_2^I(x); \\
(\neg C)^I(x) = \ominus C^I(x); \\
(m(C))^I(x) = m^I(C^I(x)); \\
(\forall R.C)^I(x) = \sup_{y \in \Delta^I} R^I(x, y) \otimes C^I(y); \\
(\forall R.C)^I(x) = \inf_{y \in \Delta^I} R^I(x, y) \triangleright C^I(y); \\
(\exists n R)^I(x) = \sup_{y_1, \ldots, y_n \in \Delta^I} \{y_1, \ldots, y_n\} = 1 \bigotimes_{i=1}^n R^I(x, y_i); \\
(\leq n R)^I(x) = \inf_{y_1, \ldots, y_{n+1} \in \Delta^I} \{y_1, \ldots, y_{n+1}\} = 1 \bigotimes_{i=1}^n R^I(x, y_i) > 0; \\
\forall T_1, \ldots, T_n, D)^I(x) = \sup_{y_1, \ldots, y_n \in \Delta_D} \{y_1, \ldots, y_n\} = 1 \bigotimes_{i=1}^n T^I(x, y_i); \\
\forall T_1, \ldots, T_n, D)^I(x) = \inf_{y_1, \ldots, y_n \in \Delta_D} \{y_1, \ldots, y_n\} = 1 \bigotimes_{i=1}^n T^I(x, y_i); \\
(\exists n T)^I(x) = \sup_{y_1, \ldots, y_{n+1} \in \Delta_D} \{y_1, \ldots, y_{n+1}\} = 1 \bigotimes_{i=1}^n T^I(x, y_i); \\
(\leq n T)^I(x) = \inf_{y_1, \ldots, y_{n+1} \in \Delta_D} \{y_1, \ldots, y_{n+1}\} = 1 \bigotimes_{i=1}^n T^I(x, y_i) > 0.
\]

The mapping $T^I$ is extended to concept inclusion, role inclusion, concept assertion, and role assertion axioms as follows (where $a, b \in I$ and $v \in \Delta^I$):

\[
(C \sqsubseteq D)^I = \inf_{x \in \Delta^I} C^I(x) \triangleright D^I(x); \\
(R \sqsubseteq S)^I = \inf_{x \in \Delta^I} R^I(x, y) \triangleright S^I(x, y); \\
(T \sqsubseteq U)^I = \inf_{x, y \in \Delta^I \times \Delta^I} T^I(x, y) \triangleright U^I(x, y); \\
(C(a))^I = C^I(a^I); \\
(R(a, b))^I = R^I(a^I, b^I); \\
(T(a, v))^I = T^I(a^I, v^I).
\]

The notion of a fuzzy interpretation $I$ satisfying a transitivity, equality, inequality, or fuzzy axiom $E$, or $I$ being a model of $E$, denoted $I \models E$, is defined as follows:

1. $I \models \text{Trans}(R)$ iff $R^I(x, y) \geq \sup_{z \in \Delta^I} R^I(x, z) \otimes R^I(z, y)$ for all $x, y \in \Delta^I$;
2. $I \models a = b$ iff $a^I = b^I$, and $I \models a \neq b$ iff $a^I \neq b^I$; and
3. $I \models a \theta n$ iff $a^I \theta n$.

A concept $C$ is satisfiable iff there exists an interpretation $I$ and an individual $x \in \Delta^I$ such that $C^I(x) > 0$. We say $I$ satisfies a fuzzy description logic knowledge base $L$. 

8
or \( \mathcal{I} \) is a model of \( L \), denoted \( \mathcal{I} \models L \), iff \( \mathcal{I} \) satisfies every \( E \in L \). We say \( L \) is satisfiable iff there exists a model of \( L \). A fuzzy axiom \( E \) is a logical consequence of \( L \), denoted \( L \models E \), iff every model of \( L \) satisfies \( E \).

**Example 6 (Shopping Agent cont’d)** Consider again the fuzzy description logic knowledge base \( L \) in Example 5. It is not difficult to verify that \( L \) is satisfiable, and that the crisp axioms \( \text{PassengerCars} \sqsubseteq \text{Vehicles} \) and \( \text{SportyCars}(\text{MazdaMX5Miata}) \) are logical consequences of \( L \). Informally, all passenger cars are vehicles, and the \( \text{MazdaMX5Miata} \) is a sports car, respectively. The following fuzzy axioms are some other logical consequences of \( L \) (under the Zadeh semantics of the connectives):

\[
\begin{align*}
C(\text{MazdaMX5Miata}) &= 1.0; & C(\text{MitsubishiEclipseSpyder}) &= 0.32; \\
D(\text{MazdaMX5Miata}) &= 0.36; & D(\text{MitsubishiEclipseSpyder}) &= 0.52.
\end{align*}
\]

Informally, the two cars \( \text{MazdaMX5Miata} \) and \( \text{MitsubishiEclipseSpyder} \) cost at most around 22 000 € to the degrees of truth 1.0 and 0.32, respectively, and they have a power of around 150 HP to the degrees of truth 0.36 and 0.52, respectively.

### Fuzzy Description Logic Programs

In this section, we present a tightly coupled approach to fuzzy disjunctive description logic programs (or simply fuzzy dl-programs) under the answer set semantics. We extend the tightly coupled disjunctive description logic programs in (Lukasiewicz, 2007a), which have very nice features compared to other tightly coupled description logic programs; see (Lukasiewicz, 2007a) for more details on these crisp programs and a comparison to related works in the literature. Note that differently from (Lukasiewicz, 2006a) (in addition to being a tightly coupled approach to fuzzy dl-programs), the fuzzy dl-programs here additionally allow for disjunctions in rule heads. We first introduce the syntax of fuzzy dl-programs and then their answer set semantics.

The basic idea behind the tightly coupled approach in this section is as follows. Suppose that we have a fuzzy disjunctive program \( P \). Under the answer set semantics, \( P \) is equivalent to its grounding \( \text{ground}(P) \). Suppose now that some of the ground atoms in \( \text{ground}(P) \) are additionally related to each other by a fuzzy description logic knowledge base \( L \). That is, some of the ground atoms in \( \text{ground}(P) \) actually represent concept and role memberships relative to \( L \). Thus, when processing \( \text{ground}(P) \), we also have to consider \( L \). However, we only want to do it to the extent that we actually need for processing \( \text{ground}(P) \). Hence, when taking a fuzzy Herbrand interpretation \( I \subseteq \text{HB}_\Phi \), we have to ensure that \( I \) represents a valid truth value assignment relative to \( L \). In other words, the main idea behind the semantics is to interpret \( P \) relative to Herbrand interpretations that also satisfy \( L \), while \( L \) is interpreted relative to general interpretations over a first-order domain. Thus, we modularly combine the standard semantics of fuzzy disjunctive programs and of fuzzy description logics as in (Lukasiewicz, 2006a), which allows for building on the standard techniques and the results of both areas. However, our new approach here allows for a much tighter integration of \( L \) and \( P \).
Syntax

We assume a function-free first-order vocabulary $\Phi$ with nonempty finite sets of constant and predicate symbols. We use $\Phi_e$ to denote the set of all constant symbols in $\Phi$. We also assume pairwise disjoint denumerable sets $A$, $R_A$, $R_D$, $I$, and $M$ of atomic concepts, abstract roles, datatype roles, individuals, and fuzzy modifiers, respectively, as in the previous section. We assume that $\Phi_e$ is a subset of $I$. This assumption guarantees that every ground atom constructed from atomic concepts, abstract roles, datatype roles, and constants in $\Phi_e$ can be interpreted in the description logic component. We do not assume any other restriction on the vocabularies, that is, $\Phi$ and $A$ (resp., $R_A \cup R_D$) may have unary (resp., binary) predicate symbols in common.

Let $X$ be a set of variables. A term is either a variable from $X$ or a constant symbol from $\Phi$. An atom is of the form $p(t_1, \ldots, t_n)$, where $p$ is a predicate symbol of arity $n \geq 0$ from $\Phi$, and $t_1, \ldots, t_n$ are terms. A literal $l$ is an atom $p$ or a default-negated atom $\lnot p$. Note that the default-negated atom $\lnot p$ refers to the lack of evidence about the truth of the atom $p$, and thus has a different meaning than the classically negated atom $\lnot p$, which refers to the presence of knowledge asserting the falsehood of the atom $p$. A disjunctive fuzzy rule (or simply fuzzy rule) $r$ is of the form

$$a_1 \lor \cdots \lor a_l \leftarrow \bigwedge a_{l+1} \lor \cdots \lor a_{l+m} \lor \bigwedge b_{l+m+1} \lor \cdots \lor b_{l+m+k} \lor \bigwedge \lnot b_{l+m+k+1} \lor \cdots \lor \lnot b_{l+m+1+k}$$

where $l \geq 1$, $m \geq k \geq 0$, $a_1, \ldots, a_l, b_{l+1}, \ldots, b_{l+m}$ are atoms, $b_1, \ldots, b_k$ are either atoms or truth values from $[0, 1]$, $\bigwedge$, $\bigvee$, $\lor$, $\bigotimes$, $\bigoplus$, $\bigotimes$, $\bigoplus$, $\land$, $\lor$, $\bot$ are conjunction strategies, $\bigwedge$, $\bigvee$, $\lor$, $\bot$ are disjunction strategies, $\bigotimes$, $\bigoplus$, $\land$, $\lor$, $\bot$ are negation strategies, and $v \in [0, 1]$. We refer to $a_1 \lor \cdots \lor a_l$ as the head of $r$, while the conjunction $b_1 \land \cdots \land b_m$ is the body of $r$. We define $H(r) = \{a_1, \ldots, a_l\}$ and $B(r) = B^+(r) \cup B^-(r)$, where $B^+(r) = \{b_1, \ldots, b_k\}$ and $B^-(r) = \{b_{k+1}, \ldots, b_m\}$. A disjunctive fuzzy program (or simply fuzzy program) $P$ is a finite set of fuzzy rules of the form (7).

We say $P$ is a normal fuzzy program iff $l = 1$ for all fuzzy rules (7) in $P$. We say $P$ is a positive fuzzy program iff $l = 1$ and $m = k$ for all fuzzy rules (7) in $P$.

A disjunctive fuzzy description logic program (or simply fuzzy dl-program) $KB = (L, P)$ consists of a fuzzy description logic knowledge base $L$ and a disjunctive fuzzy program $P$. It is called a normal fuzzy dl-program iff $P$ is a normal fuzzy program. It is called a positive fuzzy dl-program iff $P$ is a positive fuzzy program.

Example 7 (Shopping Agent cont’d) A fuzzy dl-program $KB = (L, P)$ is given by the fuzzy description logic knowledge base $L$ in Example 5 and the set of fuzzy rules $P$, which contains only the following fuzzy rule (where $x \otimes y = \min(x, y)$):

$$query(x) \leftarrow \bigotimes SportyCar(x) \land \bigotimes hasInvoice(x, y_1) \land \bigotimes hasHorsePower(x, y_2) \land \bigotimes LeqAbout22000(y_1) \land \bigotimes Around150(y_2) \geq 1.$$  

Informally, the predicate query collects all sports cars, and ranks them according to whether they cost at most around 22 000 € and have around 150 HP (such a car may be requested by a car buyer with economic needs). Another fuzzy rule is given as follows (where $\bigotimes x = 1 - x$ and $\bigotimes Around300 = \tri(x; 250, 300, 350)$):

$$query'(x) \leftarrow \bigotimes SportyCar(x) \land \bigotimes hasInvoice(x, y_1) \land \bigotimes hasMaxSpeed(x, y_2) \land \bigotimes \lnot LeqAbout22000(y_1) \land \bigotimes Around300(y_2) \geq 1.$$
Informally, this rule collects all sports cars, and ranks them according to whether they cost at least around 22,000 € and have a maximum speed of around 300 km/h (such a car may be requested by a car buyer with luxurious needs). Another fuzzy rule involving also a disjunction in its head is given as follows (where \( x \oplus y = \max(x, y) \) and \( \text{GeqAbout15000}(x) = \text{rs}(x; 12000, 15000) \)):

\[
\text{Small}(x) \oplus \text{Old}(x) \leftarrow \text{Car}(x) \land \text{hasInvoice}(x, y) \land \text{not}\_\oplus \text{GeqAbout15000}(y) \geq 0.7.
\]

This rule says that a car costing at most around 15,000 € is either small or old. Observe here that \( \text{Small} \) and \( \text{Old} \) may be two concepts in the fuzzy description logic knowledge base \( L \). That is, the tightly coupled approach to fuzzy dl-programs under the answer set semantics also allows for using the rules in \( P \) to express relationships between the concepts and roles in \( L \). This is not possible in the loosely coupled approach to fuzzy dl-programs under the answer set semantics in (Lukasiewicz, 2006a), since the dl-queries of that framework can only occur in rule bodies, but not in rule heads.

**Semantics**

We now define the answer set semantics of fuzzy dl-programs via a generalization of the standard Gelfond-Lifschitz transformation (Gelfond & Lifschitz, 1991).

In the sequel, let \( KB = (L, P) \) be a fuzzy dl-program. A *ground instance* of a rule \( r \in P \) is obtained from \( r \) by replacing every variable that occurs in \( r \) by a constant symbol from \( \Phi_c \). We denote by \( \text{ground}(P) \) the set of all ground instances of rules in \( P \). The *Herbrand base* relative to \( \Phi \), denoted \( \text{HB}_\Phi \), is the set of all ground atoms constructed with constant and predicate symbols from \( \Phi \). Observe that we define the Herbrand base relative to \( \Phi \) and not relative to \( P \). This allows for reasoning about ground atoms from the description logic component that do not necessarily occur in \( P \). Observe, however, that the extension from \( P \) to \( \Phi \) is only a notational simplification, since we can always make constant and predicate symbols from \( \Phi \) occur in \( P \) by “dummy” rules such as \( \text{constant}(c) \leftarrow \) and \( \text{p}(e) \leftarrow \text{p}(e) \), respectively. We denote by \( DL_A \) the set of all ground atoms in \( \text{HB}_\Phi \) that are constructed from atomic concepts in \( A \), abstract roles in \( R_A \), concrete roles in \( R_D \), and constant symbols in \( \Phi_c \).

We define Herbrand interpretations and the truth of fuzzy dl-programs in them as follows. An *interpretation* \( I \) is a mapping \( I : \text{HB}_\Phi \rightarrow [0, 1] \). We write \( \text{HB}_\Phi \) to denote the interpretation \( I \) such that \( I(a) = 1 \) for all \( a \in \text{HB}_\Phi \). For interpretations \( I \) and \( J \), we write \( I \subseteq J \) iff \( I(a) \leq J(a) \) for all \( a \in \text{HB}_\Phi \), and we define the *intersection* of \( I \) and \( J \), denoted \( I \cap J \), by \( (I \cap J)(a) = \min(I(a), J(a)) \) for all \( a \in \text{HB}_\Phi \). Observe that \( I \subseteq \text{HB}_\Phi \) for all interpretations \( I \). We say that \( I \) is a *model* of a ground fuzzy rule \( r \) of the form (7), denoted \( I \models r \), iff

\[
I(a_1) \oplus_1 \cdots \oplus_{l-1} I(a_l) \geq \begin{cases} 
I(b_1) \oplus_1 \cdots \oplus_{k-1} I(b_k) \oplus_k \cdots \oplus_{k+1} I(b_{k+1}) \oplus_{k+1} \cdots \oplus_{m-1} \oplus_m I(b_m) \oplus_0 v & \text{if } m \geq 1; \\
\oplus_{k+1} \cdots \oplus_{m-1} \oplus_m I(b_m) \oplus_0 v & \text{otherwise}.
\end{cases}
\]  

(8)

Here, we implicitly assume that the disjunction strategies \( \oplus_1, \ldots, \oplus_{l-1} \) and the conjunction strategies \( \otimes_1, \ldots, \otimes_{m-1}, \otimes_0 \) are evaluated from left to right (as they may not
coincide). Notice also that the above definition implicitly assumes an implication strategy $\triangleright$ that is defined by $a \triangleright b = \sup \left\{ c \in [0, 1] \mid a \otimes_0 c \leq b \right\}$ for all $a, b \in [0, 1]$ (and thus for $n, m \in [0, 1]$ and $a = n$, it holds that $a \triangleright b \geq m$ iff $b \geq n \otimes_0 m$, if we assume that the conjunction strategy $\otimes_0$ is continuous). Observe that such a relationship between the implication strategy $\triangleright$ and the conjunction strategy $\otimes$ (including also the continuity of $\otimes$) holds in Łukasiewicz, Gödel, and Product Logic (see Table 3).

We say that $I$ is a model of a fuzzy program $P$, denoted $I \models P$, iff $I \models r$ for all $r \in \text{ground}(P)$. We say $I$ is a model of a fuzzy description logic knowledge base $L$, denoted $I \models L$, iff $L \cup \{ a = I(a) \mid a \in HB_\Phi \}$ is satisfiable. Intuitively, $I$ is compatible with $L$. An interpretation $I \subseteq HB_\Phi$ is a model of a fuzzy dl-program $KB = (L, P)$, denoted $I \models KB$, iff $I \models L$ and $I \models P$. We say $KB$ is satisfiable iff it has a model.

The Gelfond-Lifschitz transform of a fuzzy dl-program $KB = (L, P)$ relative to an interpretation $I \subseteq HB_\Phi$, denoted $KB^I$, is defined as the fuzzy dl-program $(L, P^I)$, where $P^I$ is the set of all fuzzy rules obtained from $\text{ground}(P)$ by replacing all default-negated atoms $\text{not} \otimes_j b_j$ by the truth value $\otimes_j I(b_j)$. We are now ready to define the answer set semantics of fuzzy dl-programs as follows.

**Definition 8** Let $KB = (L, P)$ be a fuzzy dl-program. An interpretation $I \subseteq HB_\Phi$ is an answer set of $KB$ iff $I$ is a minimal model of $KB^I$. We say that $KB$ is consistent (resp., inconsistent) iff $KB$ has an (resp., no) answer set.

**Example 9 (Shopping Agent cont’d)** Consider again the fuzzy dl-program $KB = (L, P)$ of Example 7. It is not difficult to verify that $KB$ has an answer set, and so is consistent. In fact, $KB$ has a unique answer set $M$, and the following holds for $M$:

$$M(\text{query(MazdaMX5Miata)}) = 0.36; \quad M(\text{query(MitsubishiEclipseSpyder)}) = 0.32.$$ 

For example, the first value follows from $\min(\min(\min(\min(1, 1), 1), \text{ls}(18883; 22000, 25000)), \text{tri}(166; 125, 150, 175)), 1) = 0.36$.

We finally define the notions of cautious (resp., brave) reasoning from fuzzy dl-programs under the answer set semantics as follows.

**Definition 10** Let $KB = (L, P)$ be a fuzzy dl-program. Let $a \in HB_\Phi$ and $n \in [0, 1]$. Then, $a \triangleright n$ is a cautious (resp., brave) consequence of a fuzzy dl-program $KB$ under the answer set semantics iff $I(a) \geq n$ for every (resp., some) answer set $I$ of $KB$.

**Example 11 (Shopping Agent cont’d)** Consider again the fuzzy dl-program $KB$ of Example 7. By Example 9, $\text{query(MazdaMX5Miata)} \geq 0.36$ and $\text{query(MitsubishiEclipseSpyder)} \geq 0.32$ are both cautious and brave consequences of $KB$.

**Semantic Properties**

In this section, we summarize some important semantic properties of fuzzy dl-programs under the above answer set semantics.
Minimal Models

The following theorem shows that, like for ordinary disjunctive programs, every answer set of a fuzzy dl-program $KB$ is also a minimal model of $KB$, and the answer sets of a positive fuzzy dl-program $KB$ are the minimal models of $KB$.

**Theorem 12** Let $KB = (L, P)$ be a fuzzy dl-program. Then, (a) every answer set of $KB$ is a minimal model of $KB$, and (b) if $KB$ is positive, then the set of all answer sets of $KB$ is given by the set of all minimal models of $KB$.

Faithfulness

An important property of integrations of rules and ontologies for the Semantic Web is that they are a faithful (Motik et al., 2006) extension of both rules and ontologies.

The following theorem shows that the answer set semantics of fuzzy dl-programs faithfully extends its counterpart for fuzzy programs. That is, the answer set semantics of a fuzzy dl-program $KB = (L, P)$ with empty fuzzy description logic knowledge base $L$ coincides with the answer set semantics of its fuzzy program $P$.

**Theorem 13** Let $KB = (L, P)$ be a fuzzy dl-program such that $L = \emptyset$. Then, the set of all answer sets of $KB$ coincides with the set of all answer sets of the fuzzy program $P$.

The next theorem shows that the answer set semantics of fuzzy dl-programs also faithfully extends the first-order semantics of fuzzy description logic knowledge bases. That is, for $a \in HB_{\Phi}$ and $n \in [0, 1]$, it holds that $a \geq n$ is true in all answer sets of a positive fuzzy dl-program $KB = (L, P)$ iff $a \geq n$ is true in all fuzzy first-order models of $L \cup \text{ground}(P)$. The theorem holds also when $a$ is a ground formula constructed from $HB_{\Phi}$ using $\land$ and $\lor$, along with conjunction and disjunction strategies $\otimes$ resp. $\oplus$.

**Theorem 14** Let $KB = (L, P)$ be a positive fuzzy dl-program, and let $a \in HB_{\Phi}$ and $n \in [0, 1]$. Then, $a \geq n$ is true in all answer sets of $KB$ iff $a \geq n$ is true in all fuzzy first-order models of $L \cup \text{ground}(P)$.

As an immediate corollary, we obtain that $a \geq n$ is true in all answer sets of a fuzzy dl-program $KB = (L, \emptyset)$ iff $a \geq n$ is true in all fuzzy first-order models of $L$.

**Corollary 15** Let $KB = (L, P)$ be a fuzzy dl-program with $P = \emptyset$, and let $a \in HB_{\Phi}$ and $n \in [0, 1]$. Then, $a \geq n$ is true in all answer sets of $KB$ iff $a \geq n$ is true in all fuzzy first-order models of $L$.

Unique Name Assumption

Another aspect that may not be very desirable in the Semantic Web (Horrocks & Patel-Schneider, 2006) is the unique name assumption (which says that any two distinct constant symbols in $\Phi_c$ represent two distinct domain objects). It turns out that we actually do not have to make this assumption, since the fuzzy description logic knowledge base of a fuzzy dl-program may very well contain or imply equalities between individuals. Intuitively, since we have no unique name assumption in $L$, we also do not have to make the unique name assumption in $P$. 

13
Example 16: The unique answer set of the fuzzy dl-program $KB = (L, P) = (\{a = b\}, \{p(a) \geq 0.7\})$, where $a, b \in \Phi_c \cap I$ and $p \in \Phi \cap A$, associates with both ground atoms $p(a)$ and $p(b)$ the value 0.7, since $L$ contains the equality axiom $a = b$, and $P$ contains the fuzzy fact $p(a) \geq 0.7$.

This result is included in the following theorem, which shows an alternative characterization of the satisfaction of $L$ in $I \subseteq HB_\Phi$: Rather than being enlarged by a set of axioms of exponential size, $L$ is enlarged by a set of axioms of polynomial size. This characterization essentially shows that the satisfaction of $L$ in $I \subseteq HB_\Phi$ corresponds to checking that (i) $I$ restricted to $DL_\Phi$ satisfies $L$, and (ii) $I$ restricted to $HB_\Phi - DL_\Phi$ does not violate any equality axioms that follow from $L$. In the theorem, an equivalence relation $\sim$ on $\Phi_c$ is admissible with an interpretation $I \subseteq HB_\Phi$ iff $I(p(c_1, \ldots, c_n)) = I(p(c'_1, \ldots, c'_n))$ for all $n$-ary predicate symbols $p$, where $n > 0$, and constant symbols $c_1, \ldots, c_n, c'_1, \ldots, c'_n \in \Phi_c$ such that $c_i \sim c'_i$ for all $i \in \{1, \ldots, n\}$.

**Theorem 17** Let $L$ be a fuzzy description logic knowledge base, and let $I \subseteq HB_\Phi$. Then, $L \cup \{a = I(a) \mid a \in HB_\Phi\}$ is satisfiable iff $L \cup \{a = I(a) \mid a \in DL_\Phi\} \cup \{c \neq c' \mid c \sim c'\}$ is satisfiable for some equivalence relation $\sim$ on $\Phi_c$ admissible with $I$.

### Reduction of Fuzzy DL-Programs to DL-Programs

In this section, we present a polynomial reduction of fuzzy dl-programs to the tightly coupled dl-programs in (Lukasiewicz, 2007a). Hence, reasoning in fuzzy dl-programs under the answer set semantics can be reduced to

1. reasoning in tightly coupled dl-programs under the answer set semantics, and
2. reasoning in fuzzy description logics.

Similarly, reasoning in fuzzy description logics is additionally reduced to reasoning in crisp description logics, as shown in (Bobillo et al., 2006; Bobillo & Straccia, 2008b; Bobillo et al., 2008; Straccia, 2004).

The reduction applies to all fuzzy dl-programs $KB$ that

1. are closed under $TV_n = \{0, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for some $n > 0$, and
2. contain only combination strategies from Zadeh Logic.

Here, $KB$ is **closed** under $TV_n$ iff

1. every datatype predicate in $KB$ is interpreted by a mapping to $TV_n$,
2. every fuzzy modifier $m$ in $KB$ is interpreted by a mapping $f_m: TV_n \to TV_n$,
3. every truth value in $KB$ is from $TV_n$, and
4. every combination strategy in $KB$ is closed under $TV_n$ (which holds, e.g., for the combination strategies of Łukasiewicz, Gödel, and Zadeh Logic).
Note that for fuzzy dl-programs $KB$ that are closed under $TV_n$, the problems of deciding consistency, cautious consequences, and brave consequences are all decidable, since we only have to consider the finite number of interpretations $I \subseteq HB_\emptyset$ to $TV_n$.

We first give some preparative definitions as follows. We denote by $\Phi^\alpha$ the alphabet that is obtained from the alphabet $\Phi$ by replacing every predicate symbol $p$ by the new predicate symbols $p^\alpha$ with $\alpha \in TV_n^+ = TV_n \setminus \{0\}$. For atoms $a = p(t_1, \ldots, t_k)$ and $\alpha \in TV_n^+$, the atom $\alpha^a$ over $\Phi^\alpha$ is defined by $\alpha^a = p^\alpha(t_1, \ldots, t_k)$. Every fuzzy interpretation $I \subseteq HB_\emptyset$ is associated with the binary interpretation $t(I) = \{ \alpha^a \mid a \in HB_\emptyset, \alpha \in TV_n^+, I(a) \geq \alpha \}$.

We are now ready to define the polynomial reduction of fuzzy dl-programs to the tightly coupled dl-programs in (Łukasiewicz, 2007a). The crisp transform of a fuzzy dl-program $KB = (L, P)$ is the crisp dl-program $t(KB) = (t(L), t(P))$, where $t(L)$ is the crisp transformation of $L$ according to (Bobillo et al., 2008) and $t(P)$ is the set (i) of all rules $p^\beta(x_1, \ldots, x_k) \leftarrow p^\alpha(x_1, \ldots, x_k)$ such that $p$ is a $k$-ary predicate symbol from $\Phi$, $x_1, \ldots, x_k$ are distinct variables, $\alpha \in TV_n^+ \setminus \{\frac{1}{n}\}$, and $\beta = \alpha - \frac{1}{n}$, and (ii) of all rules $a_1^\alpha \lor \cdots \lor a_k^\alpha \leftarrow b_1^\beta \land \cdots \land b_k^\beta \land \neg b_{k+1}^\gamma \land \cdots \land \neg b_n^\gamma$ such that a rule of the form (7) belongs to $P$, $\alpha \in TV_n^+$, $\alpha \leq v$, and $\gamma = 1 - \alpha + \frac{1}{n}$. Note here that the generated crisp description logic component $t(L)$ and the generated crisp logic program component $t(P)$ have both a polynomial size in $KB$ and $TV_n$ (assuming a unary number encoding for the truth values).

**Example 18 (Shopping Agent cont’d)** The last fuzzy dl-rule of Example 7 is translated into the following dl-rules in the crisp transform (for $TV_{10} = \{0, 0.1, \ldots, 1\}$):

$$\begin{align*}
\text{Small}^{0.1}(x) \lor \text{Old}^{0.1}(x) & \leftarrow \text{Car}^{0.1}(x) \land \text{hasInvoice}^{0.1}(x, y) \land \neg \text{GeqAbout1500}^{1.0}(y), \\
\text{Small}^{0.2}(x) \lor \text{Old}^{0.2}(x) & \leftarrow \text{Car}^{0.2}(x) \land \text{hasInvoice}^{0.2}(x, y) \land \neg \text{GeqAbout1500}^{0.9}(y), \\
\text{Small}^{0.3}(x) \lor \text{Old}^{0.3}(x) & \leftarrow \text{Car}^{0.3}(x) \land \text{hasInvoice}^{0.3}(x, y) \land \neg \text{GeqAbout1500}^{0.8}(y), \\
\text{Small}^{0.4}(x) \lor \text{Old}^{0.4}(x) & \leftarrow \text{Car}^{0.4}(x) \land \text{hasInvoice}^{0.4}(x, y) \land \neg \text{GeqAbout1500}^{0.7}(y), \\
\text{Small}^{0.5}(x) \lor \text{Old}^{0.5}(x) & \leftarrow \text{Car}^{0.5}(x) \land \text{hasInvoice}^{0.5}(x, y) \land \neg \text{GeqAbout1500}^{0.6}(y), \\
\text{Small}^{0.6}(x) \lor \text{Old}^{0.6}(x) & \leftarrow \text{Car}^{0.6}(x) \land \text{hasInvoice}^{0.6}(x, y) \land \neg \text{GeqAbout1500}^{0.5}(y), \\
\text{Small}^{0.7}(x) \lor \text{Old}^{0.7}(x) & \leftarrow \text{Car}^{0.7}(x) \land \text{hasInvoice}^{0.7}(x, y) \land \neg \text{GeqAbout1500}^{0.4}(y).
\end{align*}$$

The following theorem shows that, for certain fuzzy dl-programs $KB$, the answer sets of $KB$ correspond to the answer sets of the crisp transform of $KB$.

**Theorem 19** Let $KB = (L, P)$ be a fuzzy dl-program that (i) is closed under $TV_n = \{0, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for some $n > 0$ and (ii) contains only combination strategies from Zadeh Logic. Then, $I \subseteq HB_\emptyset$ is an answer set of $KB$ iff $t(I)$ is an answer set of $t(KB)$.

**Complexity**

The following theorem shows that, for certain fuzzy dl-programs, consistency checking and cautious / brave reasoning are complete for $\text{NEXPNP}$ and $\text{co-NEXPNP} / \text{NEXPNP}^P$, respectively, and thus have the same complexity as consistency checking and cautious / brave reasoning for the tightly coupled disjunctive dl-programs in (Łukasiewicz,
These results follow from the complexity of consistency checking and cautious/brave reasoning for the tightly coupled disjunctive dl-programs in (Łukasiewicz, 2007a): the lower complexity bounds, because the tightly coupled disjunctive dl-programs in (Łukasiewicz, 2007a) are a special case of fuzzy dl-programs, and the upper complexity bounds, because of the polynomial reduction in Theorem 19.

**Theorem 20**

Given (i) a finite set of truth values \( TV_n = \{0, \frac{1}{n}, \ldots, \frac{n}{n} \} \) with \( n > 0 \), (ii) a fuzzy dl-program \( KB = (L, P) \) such that (ii.1) \( KB \) is closed under \( TV_n \), (ii.2) \( KB \) has only combination strategies from Zadeh Logic, and (ii.3) \( L \) is in fuzzy \( SHIF(D) \) or fuzzy \( SHOIN(D) \), and (iii) \( a \in HB_a \) and \( v \in [0, 1] \),

(a) deciding whether \( KB \) has an answer set is complete for \( NEXP^{NP} \), and
(b) deciding whether \( a \geq v \) is true in every (resp., some) answer set of \( KB \) is complete for \( co-NEXP^{NP} \) (resp., \( NEXP^{NP} \)).

**Tractability Results**

In this section, we present a special class of fuzzy dl-programs \( KB \) for which the problems of deciding consistency and of query processing are both data tractable. These fuzzy dl-programs are defined relative to fuzzy DL-Lite (Straccia, 2006b) (see also (DL-Media, 2008; Straccia & Visco, 2007)), which is a fuzzy generalization of the description logic DL-Lite (Calvanese et al., 2007). Note that DL-Lite and its variants are able to capture the main notions of both ontologies and conceptual modeling formalisms in databases and software engineering (Calvanese et al., 2007). By (Straccia, 2006b) (resp., (Calvanese et al., 2007)), deciding whether a knowledge base in DL-Lite (resp., fuzzy DL-Lite) is satisfiable can be done in polynomial time, and conjunctive query processing from a knowledge base in DL-Lite (resp., fuzzy DL-Lite) can also be done in polynomial time in the data complexity.

We first recall DL-Lite and fuzzy DL-Lite. Let \( A, R_A, \) and \( I \) be pairwise disjoint sets of atomic concepts, abstract roles, and individuals, respectively. A basic concept in fuzzy DL-Lite is either an atomic concept from \( A \) or an existential restriction on roles \( \exists R. \top \) (abbreviated as \( \exists R \)), where \( R \in R_A \cup R_A^{-} \). A literal in DL-Lite is either a basic concept \( b \) or the negation of a basic concept \( \neg b \). Concepts in DL-Lite are defined by induction as follows. Every basic concept in DL-Lite is a concept in DL-Lite. If \( b \) is a basic concept in DL-Lite, and \( \phi_1 \) and \( \phi_2 \) are concepts in DL-Lite, then \( \neg b \) and \( \phi_1 \cap \phi_2 \) are also concepts in DL-Lite. An axiom in DL-Lite is either

1. a concept inclusion axiom \( b \sqsubseteq \phi \), where \( b \) is a basic concept in DL-Lite, and \( \phi \) is a concept in DL-Lite, or
2. a functionality axiom \( \text{funct} R \), where \( R \in R_A \cup R_A^{-} \), or
3. a concept assertion axiom \( b(a) \), where \( b \) is a basic concept in DL-Lite and \( a \in I \), or
4. a role assertion axiom \( R(a, c) \), where \( R \in R_A \) and \( a, c \in I \).

A fuzzy concept (resp., role) assertion axiom is of the form \( b(a) \geq n \) (resp., \( R(a, c) \geq n \)), where \( b(a) \) (resp., \( R(a, c) \)) is a concept (resp., role) assertion axiom in DL-Lite, and
A fuzzy axiom in DL-Lite is either a fuzzy concept assertion axiom or a fuzzy role assertion axiom. A fuzzy knowledge base in DL-Lite $L$ is a finite set of concept inclusion, functionality, fuzzy concept assertion, and fuzzy role assertion axioms in DL-Lite.

We next define a preparative transformation on certain fuzzy dl-programs. For the conjunction strategies of Gödel and Zadeh Logic, every knowledge base in fuzzy DL-Lite can be transformed into an equivalent one in fuzzy DL-Lite $\text{trans}(L)$ in which every concept inclusion axiom is of form $b \sqsubseteq \ell$, where $b$ (resp., $\ell$) is a basic concept (resp., literal) in DL-Lite. For fuzzy dl-programs $KB = (L, P)$ with $L$ in DL-Lite, we then define $\text{trans}(KB) = (L, \text{trans}(P))$ by $\text{trans}(P) = P \cup \{b'(X) \leftarrow b(X) \mid b \sqsubseteq b' \in \text{trans}(L), b'$ is a basic concept $\}$ $\cup \{\exists R(X) \leftarrow R(X, Y) \mid R \in R_A \cap \Phi \} \cup \{\exists R^-(Y) \leftarrow R(X, Y) \mid R \in R_A \cap \Phi \}$. Intuitively, we make explicit all the rule-based relationships between the predicates in $P$ that are implicitly encoded in $L$.

We now define the notion of local stratification for disjunctive fuzzy programs. A local stratification of a disjunctive fuzzy program $P$ is a mapping $\lambda: HB_{\Phi} \rightarrow \{0, 1, \ldots, k\}$ such that $\lambda(\alpha) \geq \lambda(\beta)$ (resp., $\lambda(\alpha) > \lambda(\beta)$) for each $r \in \text{ground}(P)$, $\alpha \in H(r)$, and $\beta \in B^+(r)$ (resp., $\beta \in B^-(r)$), where $k \geq 0$ is the length of $\lambda$. We say $P$ is locally stratified iff it has a stratification $\lambda$ of some length $k \geq 0$.

We are now ready to define fuzzy dl-programs in DL-Lite as follows. We say that a fuzzy dl-program $KB = (L, P)$ is defined in DL-Lite iff

1. $L$ is in fuzzy DL-Lite and interpreted relative to the conjunction strategies of Gödel or Zadeh Logic,
2. $\text{trans}(P)$ is normal and locally stratified, and
3. $KB$ is closed under $TV_n = \{0, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for some $n > 0$, where we assume a unary encoding of the numbers in $TV_n$.

Like for the crisp case (Lukasiewicz, 2007a), consistency checking and query processing for fuzzy dl-programs in DL-Lite are both data tractable. This result follows from the fact that such programs have either no or a unique answer set, which can be computed by a finite sequence of fixpoint iterations.

**Theorem 21** Let $KB$ be a fuzzy dl-program in DL-Lite. Then,

(a) deciding whether $KB$ has an answer set, and

(b) computing the truth value of a ground atom $a \in HB_{\Phi}$ in the answer set of $KB$

can both be done in polynomial time in the data complexity.

**Example 22 (Shopping Agent cont’d)** Consider the fuzzy dl-program $KB = (L, P)$, where $L$ is a finite set of concept and (abstract) role assertion axioms, with truth values from some $TV_n = \{0, \frac{1}{n}, \ldots, \frac{n}{n}\}$ with $n > 0$, and $P$ is the first rule of Example 7. Then, $KB$ is defined in DL-Lite, $KB$ is satisfiable, and the truth values of all $\text{query}(c)$ in the answer set of $KB$ can be computed in polynomial time in the data complexity.
Related Work

Most closely related to the presented approach are other integrations of rules and ontologies that allow for handling fuzzy vagueness. Also related are integrations of rules and ontologies that allow for handling probabilistic uncertainty.

The earliest works on fuzzy dl-programs are (Straccia, 2006b) and (Łukasiewicz, 2006a), which propose tightly coupled positive fuzzy dl-programs under the canonical least model semantics and loosely coupled normal fuzzy dl-programs under the answer set semantics, respectively. Moreover, (Łukasiewicz & Straccia, 2007b) presents an efficient top-k retrieval technique in this context, while (Venetis et al., 2007) studies a fuzzy extension of the mapping between ontologies and rules in (Grosof et al., 2003).

The earliest work on probabilistic dl-programs (Łukasiewicz, 2005) is based on loosely coupled normal dl-programs under the answer set and the well-founded semantics. Recent extensions include especially a tractable variant (Łukasiewicz, 2007b), particularly for probabilistic data integration, a unified framework for handling both fuzzy vagueness and probabilistic uncertainty (Łukasiewicz & Straccia, 2007a), and a tightly coupled disjunctive version (Cali & Łukasiewicz, 2007; Cali et al., 2008), particularly for representing and reasoning with mappings between ontologies. A related (less expressive) approach is (Predoiu & Stuckenschmidt, 2007), which is based on Bayesian logic programs, combining dl-programs with Bayesian networks.

Conclusion

We have presented tightly coupled fuzzy dl-programs under the answer set semantics, which generalize the tightly coupled disjunctive dl-programs in (Łukasiewicz, 2007a) by fuzzy vagueness in both the description logic and the logic program component. We have shown that the new formalism faithfully extends both fuzzy disjunctive programs and fuzzy description logics, and that under suitable assumptions, reasoning in the new formalism is decidable. We have presented a polynomial reduction for certain fuzzy dl-programs to tightly coupled disjunctive dl-programs, and we have analyzed the complexity of consistency checking and query processing for certain fuzzy dl-programs. Finally, we have also provided a special case of fuzzy dl-programs for which deciding consistency and query processing are both data tractable.

An interesting issue for future work is the implementation of the presented framework and its experimental testing along applications in practice.

Acknowledgments. This work was partially supported by the German Research Foundation (DFG) under the Heisenberg Programme. We thank the reviewers of this paper and its RR-2007 abstract for their useful comments, which helped to improve this work.

Appendix: Proofs

Proof of Theorem 12. (a) Let $I \subseteq HB_\Phi$ be any answer set of $KB$. That is, $I$ is a minimal model of $KB^I = (L, P^I)$. In particular, (i) $I \models L$ and (ii) $I \models r$ for every $r \in P^I$. 

18
Proof of Theorem 13. Observe first that \( I \subseteq HB_\Phi \) is a model of \( KB^I = (L, P^I) \) iff (i) \( I \models L \) and (ii) \( I \models r \) for every \( r \in \text{ground}(P) \). Since \( L = \emptyset \), this is equivalent to \( I \models r \) for every \( r \in P^I \). Thus, \( I \subseteq HB_\Phi \) is a minimal model of \( KB^I \) iff \( I \) is a minimal model of \( P^I \). That is, \( I \) is also a minimal model of \( KB^I \). But this contradicts \( I \) being a minimal model of \( KB^I \). In summary, this shows that \( I \) is a minimal model of \( KB \).

(b) Since \( KB^I = (L, \text{ground}(P)) \) for every positive fuzzy dl-program \( KB = (L, P) \), it follows that the set of all answer sets of \( KB \), that is, the set of all minimal models of \( KB^I \), coincides with the set of all minimal models of \( KB \).

Proof of Theorem 14. Observe first that, by Theorem 12, since \( P \) is positive, the set of all answer sets of \( KB \) is the set of all minimal models \( I \subseteq HB_\Phi \) of \( KB \). Observe then that for \( a \in HB_\Phi \), \( a \geq n \) is true in all minimal models \( I \subseteq HB_\Phi \) of \( KB \) iff \( a \geq n \) is true in all minimal models \( I \subseteq HB_\Phi \) of \( KB \). It thus remains to show that \( a \geq n \) is true in all minimal models \( I \subseteq HB_\Phi \) of \( KB \) iff \( a \geq n \) is true in all first-order models of \( L \cup \text{ground}(P) \):

\((\Rightarrow)\) Suppose \( a \geq n \) is true in all minimal models \( I \subseteq HB_\Phi \) of \( KB \). Let \( \mathcal{I} \) be any fuzzy first-order model of \( L \cup \text{ground}(P) \). Let \( I \subseteq HB_\Phi \) be defined by \( I(b) = I(b) \) for all \( b \in HB_\Phi \). Then, \( \mathcal{I} \) is a model of \( L^* = L \cup \{ a = I(a) \mid a \in HB_\Phi \} \), and thus \( L^* \) is satisfiable. Hence, \( I \) is a model of \( P \). Since \( \mathcal{I} \) is a model of \( \text{ground}(P) \), also \( I \) is a model of \( \text{ground}(P) \). In summary, \( I \) is a model of \( KB \). Hence, \( a \geq n \) is true in \( I \), and thus \( a \geq n \) is true in \( I \). Overall, \( a \geq n \) is true in all first-order models of \( L \cup \text{ground}(P) \).

\((\Leftarrow)\) Suppose \( a \geq n \) is true in all first-order models of \( L \cup \text{ground}(P) \). Let \( I \subseteq HB_\Phi \) be any model of \( KB \). Then, \( L^* = L \cup \{ a = I(a) \mid a \in HB_\Phi \} \) is satisfiable. Let \( \mathcal{I} \) be a first-order model of \( L^* \). Then, \( \mathcal{I} \) is a particular model of \( L \). Furthermore, since \( I \) is a model of \( \text{ground}(P) \), also \( \mathcal{I} \) is a model of \( \text{ground}(P) \). In summary, \( \mathcal{I} \) is a model of \( L \cup \text{ground}(P) \). It thus follows that \( a \geq n \) is true in \( \mathcal{I} \), and thus \( a \geq n \) is also true in \( I \). Overall, \( a \geq n \) is true in all models \( I \subseteq HB_\Phi \) of \( KB \).

Proof of Theorem 17. \((\Rightarrow)\) Let \( \mathcal{I} \) be a first-order model of \( L^* = L \cup \{ a = I(a) \mid a \in HB_\Phi \} \). Let the equivalence relation \( \sim \) on \( \Phi_\varepsilon \) be defined by \( c \sim d \) iff \( c^d = d^c \). Since \( \mathcal{I} \) is a model of \( L^* \), it follows that \( \sim \) is admissible with \( I \). Furthermore, it follows that \( \mathcal{I} \) is a model of \( L \cup \{ a = I(a) \mid a \in DL_\Phi \} \cup \{ c \neq c' \mid c \neq c' \} \).

\((\Leftarrow)\) Let \( \mathcal{I} \) be a model of \( L \cup \{ a = I(a) \mid a \in DL_\Phi \} \cup \{ c \neq c' \mid c \neq c' \} \) for some equivalence relation \( \sim \) on \( \Phi_\varepsilon \) admissible with \( I \). Thus, \( \mathcal{I} \) can be extended to a model \( \mathcal{I}' \) of \( L \cup \{ a = I(a) \mid a \in HB_\Phi \} \) by \( \mathcal{I}'(b) = I(b) \), for all \( b \in HB_\Phi - DL_\Phi \).

Proof of Theorem 19. Recall that every fuzzy interpretation \( I \subseteq HB_\Phi \) (to \( TV_n^+ \)) corresponds to the binary interpretation \( t(I) = \{ a^\alpha \mid a \in HB_\Phi, \alpha \in TV_n^+, I(a) \geq \alpha \} \).
Hence, every \( a^\alpha \) encodes that the truth value of \( a \) is at least \( \alpha \). Thus, the rules 
\[
p^\beta(x_1, \ldots, x_k) \leftarrow p^\alpha(x_1, \ldots, x_k)
\]
such that \( p \) is a \( k \)-ary predicate symbol from \( \Phi \), \( x_1, \ldots, x_k \) are distinct variables, \( \alpha \in TV_n^+ \{ 1/n \} \), and \( \beta = \alpha - 1/n \) encode all the logical relationships between the \( a^\alpha \)'s, while the other rules in \( t(P) \) encode the instances of every rule in \( P \) under every possible truth value combination of its body atoms. Then, for every fuzzy interpretation \( I \subseteq HB_\Phi \), (i) the Gelfond-Lifschitz transform of \( P \) relative to \( I \) directly corresponds to the Gelfond-Lifschitz transform of \( t(P) \) relative to \( t(I) \), and (ii) \( L \) directly corresponds to \( t(L) \) (Bobillo et al., 2008; Straccia, 2004). This then implies that the fuzzy interpretation \( I \subseteq HB_\Phi \) is an answer set of \( KB \) iff \( t(I) \) is an answer set of \( t(KB) \).

**Proof of Theorem 20.** As for the lower bounds, fuzzy dl-programs generalize tightly coupled disjunctive dl-programs, and consistency checking and cautious/brave reasoning in the latter are hard for \( \text{NEXP}^{\text{NP}} \) and \( \text{co-NEXP}^{\text{NP}}/\text{NEXP}^{\text{NP}} \) (Łukasiewicz, 2007a), respectively.

As for the upper bounds, to solve the three problems, we first transform \( KB = (L, P) \) into its crisp equivalent \( t(KB) = (t(L), t(P)) \), as described in the body of the paper. Note that both the crisp description logic component \( t(L) \) and the crisp logic program component \( t(P) \) have a polynomial size in \( KB \) and \( TV_n^+ \). By Theorem 19, \( KB \) has an answer set iff \( t(KB) \) has an answer set. As shown in (Łukasiewicz, 2007a), deciding the latter is in \( \text{NEXP}^{\text{NP}} \). By Theorem 19, \( a \geq v \) is true in every/some answer set of \( KB \) iff \( a^v \) is true in every/some answer set of \( t(KB) \). As shown in (Łukasiewicz, 2007a), deciding the latter is in \( \text{co-NEXP}^{\text{NP}}/\text{NEXP}^{\text{NP}} \).

**Proof of Theorem 21 (sketch).** Like for the crisp case (Łukasiewicz, 2007a), it can be shown that fuzzy dl-programs in \( DL-Lite \) have either no or a unique answer set, which can be computed by a finite sequence of fixpoint iterations, as usual. Hence, for such programs, consistency checking and query processing are both data tractable.

**References**


