Tightly Integrated Fuzzy Description Logic Programs
Under the Answer Set Semantics for the Semantic Web

Thomas Lukasiewicz\textsuperscript{1,*} and Umberto Straccia\textsuperscript{2}

\textsuperscript{1} DIS, Universit\`a di Roma “La Sapienza”, Via Salaria 113, I-00198 Rome, Italy
\textsuperscript{2} ISTI-CNR, Via G. Moruzzi 1, I-56124 Pisa, Italy
lukasiewicz@dis.uniroma1.it
straccia@isti.cnr.it

Abstract. We present a novel approach to fuzzy dl-programs under the answer set semantics, which is a tight integration of fuzzy disjunctive programs under the answer set semantics with fuzzy description logics. From a different perspective, it is a generalization of tightly integrated 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 programs and fuzzy description logics, and that under suitable assumptions, reasoning in the new formalism is decidable. Furthermore, we present a polynomial reduction of certain fuzzy dl-programs to tightly integrated disjunctive dl-programs. We also provide a special case of fuzzy dl-programs for which deciding consistency and query processing have both a polynomial data complexity.

1 Introduction

The Semantic Web \cite{16} 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 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 \cite{29,11}, 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 \cite{11}. As shown in \cite{9}, ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic \textit{SHIQ(D)} (resp., \textit{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 [4,5,3]) or concepts and roles from $L$ as unary resp. binary predicates in a tight integration (see especially [21,22,16]).

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 [14,13] and [27,28,15], respectively.

In this paper, we continue this line of research. We present tightly integrated 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 [27,28] and on loosely integrated fuzzy normal dl-programs [15], to our knowledge, this is the first approach to tightly integrated fuzzy disjunctive dl-programs (with default negation in rule bodies).

The main contributions of this paper can be summarized as follows:

- We present a novel approach to fuzzy dl-programs, which is a tight integration of fuzzy disjunctive programs under the answer set semantics with fuzzy description logics. It is a generalization of the tightly integrated disjunctive dl-programs in [16] by fuzzy vagueness in both the description logic and the logic program 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. Furthermore, 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.
- In the extended report [17], we also present a polynomial reduction for certain fuzzy dl-programs to the tightly integrated disjunctive dl-programs in [16]. Furthermore, we delineate a special case of fuzzy dl-programs where deciding consistency and query processing have both a polynomial data complexity.

The rest of this paper is organized as follows. Section 2 recalls combination strategies and fuzzy description logics. Section 3 introduces the syntax of fuzzy dl-programs and defines their answer set semantics. In Section 4 we analyze some semantic properties of fuzzy dl-programs under the answer set semantics. Section 5 summarizes our main results and gives an outlook on future research. Note that further results and technical details are given in the extended report [17].
Table 1. 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 ⊗ b</td>
<td>min(a,b)</td>
<td>a · b</td>
<td>min(a,b)</td>
</tr>
<tr>
<td>a ⊕ b</td>
<td>max(a + b - 1, 0)</td>
<td>max(a,b)</td>
<td>a + b - a · b</td>
</tr>
<tr>
<td>a ▷ b</td>
<td>min(1 - a + b, 1)</td>
<td>min(1,b/a)</td>
<td>max(1 - a,b)</td>
</tr>
<tr>
<td>⊗ a</td>
<td>1 - a</td>
<td>1 if a = 0</td>
<td>1 if a = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 otherwise</td>
<td>0 otherwise</td>
</tr>
</tbody>
</table>

2 Preliminaries

In this section, we illustrate the notions of combination strategies and fuzzy description logics through some examples; more details are given in the extended report [17].

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 ⊗, ⊕, ▷, and ⊹, respectively, which are functions ⊗, ⊕, ▷, and ⊹: [0, 1] × [0, 1] → [0, 1] and ⊹: [0, 1] → [0, 1] that generalize the ordinary Boolean operators ∧, ∨, →, and ¬, respectively, to the set of truth values [0, 1]. As usual, we assume that combination strategies have some natural algebraic properties [17]. Note that conjunction and disjunction strategies are also called triangular norms and triangular co-norms [8], respectively.

Example 2.1. The combination strategies of various fuzzy logics are shown in Table 1.

Fuzzy Description Logics. We now illustrate fuzzy SHI,F(D) and fuzzy SHOI,N(D) [25,26] (see also [23]) through an example. There also exists an implementation of fuzzy SHI,F(D) (the fuzzyDL system; see http://gaia.isti.cnr.it/~straccia). Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between classes of individuals, respectively. A description logic knowledge base encodes in particular subset relationships between classes of individuals, subset relationships between binary relations between classes, the membership of individuals to classes, and the membership of pairs of individuals to binary relations between classes. In fuzzy description logics, these relationships and memberships then have a degree of truth in [0, 1].
Example 2.2 (Shopping Agent). The following axioms are an excerpt of the description logic knowledge base \( L \) that conceptualizes a car selling web site:

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

Eqs. 1-3 describe the concept taxonomy of the site, while Eq. 4 describes the datatype attributes of the cars sold in the site. Eqs. 5-6 describe the properties of some sold cars.

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:

\[
\begin{align*}
C & = (\exists \text{hasInvoice}.\text{LeqAbout22000}) \\
D & = (\exists \text{hasHorsePower}.\text{Around150}) ,
\end{align*}
\]

where \( \text{LeqAbout22000} = \text{L}(22000, 25000) \) and \( \text{Around150} = \text{Tri}(125, 150, 175) \) (see Fig. 1). The latter two equations define the fuzzy concepts of “at most about 22 000 €” and “around 150 HP”. 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.

The following fuzzy axioms are (tight) logical consequences of the above description logic knowledge base \( L \) (under the Zadeh semantics of the connectives):

\[
\begin{align*}
C(\text{MazdaMX5Miata}) & \geq 1.0 ; \\
C(\text{MitsubishiEclipseSpyder}) & \geq 0.32 ; \\
D(\text{MazdaMX5Miata}) & \geq 0.36 ; \\
D(\text{MitsubishiEclipseSpyder}) & \geq 0.56 .
\end{align*}
\]

3 Fuzzy Description Logic Programs

In this section, we present a tightly integrated approach to fuzzy disjunctive description logic programs (or simply fuzzy dl-programs) under the answer set semantics.
We extend the tightly integrated disjunctive description logic programs in [16], which have very nice features compared to other tightly integrated description logic programs; see [15] for more details and a comparison to related works in the literature. Observe that differently from [15] (in addition to being a tightly integrated approach to fuzzy dl-programs), the fuzzy dl-programs here are based on fuzzy description logics as in [26]. Furthermore, they 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 integrated 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 it 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 [15], 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 \( \Phi_c \) to denote the set of all constant symbols in \( \Phi \). We also assume pairwise disjoint (nonempty) denumerable sets \( A, R_A, R_D, I, \) and \( M \) of atomic concepts, abstract roles, datatype roles, individuals, and fuzzy modifiers, respectively; see [17]. We assume that \( \Phi_c \) is a subset of \( I \). This assumption guarantees that every ground atom constructed from atomic concepts, abstract roles, datatype roles, and constants in \( \Phi_c \) 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 \( \mathcal{X} \) be a set of variables. A term is either a variable from \( \mathcal{X} \) 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 \( \text{not} \ p \). A disjunctive fuzzy rule (or simply fuzzy rule) \( r \) is of the form

\[
a_1 \lor_{\oplus_1} \cdots \lor_{\oplus_{l-1}} a_l \leftarrow \otimes_0 b_1 \land_{\otimes_1} b_2 \land_{\otimes_2} \cdots \land_{\otimes_{k-1}} b_k \land_{\otimes_k} \text{not}_{\otimes_{k+1}} b_{k+1} \land_{\otimes_{k+2}} \cdots \land_{\otimes_{m-1}} \text{not}_{\otimes_m} b_m \geq v,
\]

where \( l \geq 1, m \geq k \geq 0, a_1, \ldots, a_l, b_k, \ldots, b_m \) are atoms, \( b_1, \ldots, b_k \) are either atoms or truth values from \([0, 1]\), \( \oplus_1, \ldots, \oplus_{l-1} \) are disjunction strategies, \( \otimes_0, \ldots, \otimes_{m-1} \) are conjunction strategies, \( \text{not}_{\otimes_{k+1}}, \ldots, \text{not}_{\otimes_m} \) are negation strategies, and \( v \in [0, 1] \). We refer to \( a_1 \lor_{\oplus_1} \cdots \lor_{\oplus_{l-1}} a_l \) as the head of \( r \), while the conjunction \( b_1 \land_{\otimes_0} \cdots \land_{\otimes_{m-1}} \text{not}_{\otimes_m} b_m \) 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 \( r \) in \( P \). We say \( P \) is a positive fuzzy program iff \( l = 1 \) and \( m = k \) for all fuzzy rules \( r \) 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 3.1 (Shopping Agent cont’d). A fuzzy dl-program \( KB = (L, P) \) is given by the fuzzy description logic knowledge base \( L \) in Example 2.2 and the set of fuzzy rules \( P \), which contains only the following fuzzy rule (where \( x \otimes y = \min(x, y) \)):

\[
\text{query}(x) \leftarrow \odot \text{SportyCar}(x) \land \odot \text{hasInvoice}(x, y_1) \land \odot \text{hasHorsePower}(x, y_2) \land \text{LeqAbout22000}(y_1) \land \text{Around150}(y_2) \geq 1.
\]

Informally, the predicate \( \text{query} \) collects all sports cars, and ranks them according to whether they cost at most around 22,000 € and have a maximum speed of around 300 km/h. Another fuzzy rule is given as follows (where \( \ominus x = 1 - x \) and \( \text{Around300} = \text{Tri}(250, 300, 350) \)):

\[
\text{query}'(x) \leftarrow \odot \text{SportyCar}(x) \land \odot \text{hasInvoice}(x, y_1) \land \odot \text{hasMaxSpeed}(x, y_2) \land \text{LeqAbout22000}(y_1) \land \text{Around300}(y_2) \geq 1.
\]

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. Another fuzzy rule involving also a disjunction in its head is given as follows (where \( x \oplus y = \max(x, y) \)):

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

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 integrated 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 integrated approach to fuzzy dl-programs under the answer set semantics in [13], 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. 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 \( \Phi \). We denote by \( \text{ground}(P) \) the set of all ground instances of rules in \( P \). The Herbrand base relative to \( \Phi \), denoted \( HB_{\Phi} \), 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
constant(c) ← and p(e) ← p(e), respectively. We denote by \( DL_f \) the set of all ground atoms in \( HB_f \) 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 \( I : HB_f \rightarrow [0, 1] \). We write \( HB_f \) to denote the interpretation \( I \) such that \( I(a) = 1 \) for all \( a \in HB_f \). For interpretations \( I \) and \( J \), we write \( I \subseteq J \) iff \( I(a) \leq J(a) \) for all \( a \in HB_f \), 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 \( a \in HB_f \). Observe that \( I \subseteq HB_f \) 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

\[
I(a_1) \oplus_1 \cdots \oplus_I I(a_l) \geq I(b_1) \otimes_1 \cdots \otimes_{k-1} I(b_k) \otimes_k \\
\otimes_{k+1} I(b_{k+1}) \otimes_{k+1} \cdots \otimes_{m-1} \otimes_{m} I(b_m) \otimes_0 v. \tag{8}
\]

Here, we implicitly assume that the disjunction strategies \( \oplus \) and conjunction strategies \( \otimes \) are evaluated from left to right. Notice also that the above definition implicitly assumes an implication strategy \( \supset \) that is defined by \( a \supset b = \sup \{ c \in [0, 1] \mid a \otimes c \leq b \} \) for all \( a, b \in [0, 1] \) (and thus for \( n, m \in [0, 1] \) and \( a = n \), it holds that \( a \supset b \geq m \) iff \( b \geq n \otimes m \), if we assume that the conjunction strategy \( \otimes \) is continuous). Observe that such a relationship between the implication strategy \( \supset \) and the conjunction strategy \( \otimes \) (including also the continuity of \( \otimes \)) holds in Łukasiewicz, Gödel, and Product Logic (see Table 1). We say that \( I \) is a model of a fuzzy program \( P \), denoted \( I \models P \), iff \( I \models r \) for all \( r \in gound(P) \). We say \( I \) is a model of a fuzzy description logic knowledge base \( L \), denoted \( I \models L \), iff \( L \cup \{ a = I(a) \mid a \in HB_f \} \) is satisfiable. An interpretation \( I \subseteq HB_f \) 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 \( I \subseteq HB_f \), 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 \( gound(P) \) by replacing all default-negated atoms \( not \circ_j b_j \) by the truth value \( \circ_j I(b_j) \). We are now ready to define the answer set semantics of fuzzy dl-programs as follows.

**Definition 3.1.** Let \( KB = (L, P) \) be a fuzzy dl-program. An interpretation \( I \subseteq HB_f \) 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.

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

**Definition 3.2.** Let \( KB = (L, P) \) be a fuzzy dl-program. Let \( a \in HB_f \) and \( n \in [0, 1] \). Then, \( a \geq n \) 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 3.2 (Shopping Agent cont’d).** Consider again the fuzzy dl-program \( KB = (L, P) \) of Example 3.1. The following holds for the answer set \( M \) of \( KB \):

\[
M(query(MazdaMX5Miata)) = 0.36; \quad M(query(MitsubishiEclipseSpyder)) = 0.32.
\]


4 Semantic Properties

In this section, we summarize some semantic properties (especially those relevant for the Semantic Web) 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 4.1.** 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 is that they are a faithful [18,19] 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 4.2.** 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 4.3.** 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 4.1.** 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 [10] 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.

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 \( I \) 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 4.4.** 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 \( L \cup \{a = I(a) \mid a \in DL_\Phi \} \cup \{c = c' \mid c \not\sim c' \} \) is satisfiable for some equivalence relation \( \sim \) on \( \Phi_c \) admissible with \( I \).

5 Summary and Outlook

We have presented an approach to tightly integrated fuzzy dl-programs under the answer set semantics, which generalizes the tightly integrated disjunctive dl-programs in [16] 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. Furthermore, in [17], we have presented a polynomial reduction for certain fuzzy dl-programs to tightly integrated disjunctive dl-programs. Finally, in [17], we have also provided a special case of fuzzy dl-programs for which deciding consistency and query processing have both a polynomial data complexity.

An interesting topic for future research is to analyze the computational complexity of the main reasoning problems in fuzzy dl-programs, and to implement the approach. Another interesting issue is to extend fuzzy dl-programs by classical negation.

Acknowledgments. This work has been partially supported by a Heisenberg Professorship of the German Research Foundation (DFG). We thank the reviewers for their constructive comments, which helped to improve this work.

References