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

Thomas Lukasiewicz*
DIS, Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lukasiewicz@dis.uniroma1.it

Abstract

Vagueness and imprecision abound in multimedia information processing and retrieval. In this paper, we present an approach to fuzzy description logic programs under the answer set semantics for the Semantic Web, which is an integration of description logics with nonmonotonic logic programs under the answer set semantics (with default negation in rule bodies) that also allows for representing and reasoning with vagueness and imprecision. More concretely, we define a canonical semantics of positive and stratified fuzzy dl-programs in terms of a unique least model and iterative least models, respectively. We then define the answer set semantics of general fuzzy dl-programs, and show in particular that all answer sets of a fuzzy dl-program are minimal models, and that the answer set semantics of positive and stratified fuzzy dl-programs coincides with their canonical least model and iterative least model semantics, respectively. Furthermore, we also provide a characterization of the canonical semantics of positive and stratified fuzzy dl-programs in terms of a fixpoint and an iterative fixpoint semantics, respectively.

1. Introduction

The *Semantic Web* [1, 6] 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.

*Alternate address: Institut für Informationssysteme, TU Wien, Favoritenstr. 9-11, A-1040 Wien, Austria; lukasiewicz@kr.tuwien.ac.at.

The Semantic Web consists of several hierarchical layers, where the *Ontology layer*, in form of the *OWL Web Ontology Language* [32, 10] (recommended by the W3C), 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 [10]. As shown in [8], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$). On top of the Ontology layer, the *Rules*, *Logic*, and *Proof layers* of the Semantic Web will be developed next, which should offer sophisticated representation and reasoning capabilities. As a first effort in this direction, *RuleML* (Rule Markup Language) [2] is an XML-based markup language for rules and rule-based systems, whereas the OWL Rules Language [9] is a first proposal for extending OWL by Horn clause rules.

A key requirement of the layered architecture of the Semantic Web is to integrate the Rules and the Ontology layer. In particular, it is crucial to allow for building rules on top of ontologies, that is, for rule-based systems that use vocabulary from ontology knowledge bases. Another type of combination is to build ontologies on top of rules, which means that ontological definitions are supplemented by rules or imported from rules. Towards this goal, the works [4, 5] have proposed *description logic programs* (or simply *dl-programs*), which are of the form $KB = (L, P)$, where L is a knowledge base in a description logic and P is a finite set of *description logic rules* (or simply *dl-rules*). Such dl-rules are similar to usual rules in logic programs with negation as failure, but may also contain *queries to* L in their bodies, which are given by special atoms (on which possibly default negation may apply). Another important feature of dl-rules is that queries to L also allow for specifying an input from P, and thus for a *flow of information from* P *to* L, besides the flow of information from L to P, given by any query to L. Hence, description logic programs allow for building rules on top of ontologies, but also (to some extent) building ontologies on top of rules. In this way, additional knowledge (gained in the program) can be supplied to L before querying. The semantics of dl-programs was defined in [4] and [5] as an extension of the answer set semantics by Gelfond and Lifschitz [7] and the well-founded semantics by Van Gelder, Ross, and Schlipf [31], respectively, which are the two most widely used semantics for nonmonotonic logic programs. The description logic knowledge bases in dl-programs are specified in the well-known description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$.

In [16, 17], towards sophisticated representation and reasoning techniques that also allow for modeling probabilistic uncertainty in the Rules, Logic, and Proof layers of the Semantic Web, we have presented *probabilistic description logic programs* (or simply *probabilistic dl-programs*), which generalize dl-programs under the answer set and well-founded semantics by probabilistic uncertainty. They have been developed as a combination of dl-programs with Poole's independent choice logic (ICL) [22].

In this paper, we continue this line of research towards more sophisticated representation and reasoning techniques for the Semantic Web. Here, we present *fuzzy description logic programs* (or simply *fuzzy dl-programs*) *under the answer set semantics*, which generalize dl-programs under the answer set semantics by fuzzy imprecision and vagueness. Even though there has been previous work on positive fuzzy description logic programs by Straccia [28, 29], to our knowledge, this is the first approach to fuzzy description logic programs with default negation in rule bodies. Furthermore, differently from Straccia, we also allow for a flow of information from the logic program component to the description logic component of a fuzzy dl-program.

The main contributions of this paper are as follows:

- We introduce a simple fuzzy extension of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which allows for fuzzy concept and fuzzy role assertions, and which is intuitively based on a mapping to several layers of ordinary concepts and roles in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$.

- We introduce fuzzy dl-programs, which properly generalize dl-programs in [4] (where rule bodies may contain default-negated atoms) by fuzzy vagueness and imprecision. We define a natural semantics of positive and stratified fuzzy dl-programs in terms of a unique least model and iterative least models, respectively.

- We then define the answer set semantics of general fuzzy dl-programs. We also show that all answer sets of a fuzzy dl-program are minimal models, and that the answer set semantics of positive and stratified fuzzy dl-programs coincides with their canonical least model and iterative least model semantics, respectively.

- We also provide a characterization of the canonical semantics of positive and stratified fuzzy dl-programs in terms of a fixpoint and an iterative fixpoint semantics.

The rest of this paper is organized as follows. In Section 2, we recall the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. In Section 3, we define our fuzzy extension of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. Section 4 introduces fuzzy dl-programs, and defines the canonical semantics of positive and stratified fuzzy dl-programs, as well as the answer set semantics of general fuzzy dl-programs. In Section 5, we characterize the canonical models of positive and stratified fuzzy dl-programs in terms of a fixpoint and an iterative fixpoint semantics. Section 6 summarizes our main results and gives an outlook on future research.

2 $\mathcal{SHIF}(\mathrm{D})$ and $\mathcal{SHOIN}(\mathrm{D})$

In this section, we recall the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL, respectively. See especially [8] for further details. 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.

2.1 Syntax

We first describe the syntax of $\mathcal{SHOIN}(\mathbf{D})$. We assume a set of *elementary datatypes* and a set of *data values*. A *datatype* is either an elementary datatype or a set of data values (called *datatype oneOf*). A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype* (or *concrete*) *domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that associates with every elementary datatype a subset of $\Delta^{\mathbf{D}}$ and with every data value an element of $\Delta^{\mathbf{D}}$. The mapping $\cdot^{\mathbf{D}}$ is extended to all datatypes by $\{v_1, \ldots\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots\}$. Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ be nonempty finite and pairwise disjoint sets of *atomic concepts*, *abstract roles*, *datatype* (or *concrete*) *roles*, and *individuals*, respectively. We denote by $\mathbf{R}_A^-$ the set of inverses R^- of all abstract roles $R \in \mathbf{R}_A$.

A *role* is an element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Every $C \in \mathbf{A}$ is a concept, and if $o_1, \ldots, o_n \in \mathbf{I}$, then $\{o_1, \ldots, o_n\}$ is a concept (called *oneOf*). If C, C_1, and C_2 are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then also $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, and $\neg C$ are concepts (called *conjunction*, *disjunction*, and *negation*, respectively), as well as $\exists R.C$, $\forall R.C$, $\geq nR$, and $\leq nR$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. If D is a datatype and $U \in \mathbf{R}_D$, then $\exists U.D$, $\forall U.D$, $\geq nU$, and $\leq nU$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. We write $\top$ and $\bot$ to abbreviate $C \sqcup \neg C$ and $C \sqcap \neg C$, respectively, and we eliminate parentheses as usual.

An *axiom* is an expression of one of the following forms: (1) $C \sqsubseteq D$ (called *concept inclusion axiom*), where C and D are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $C(a)$ (called *concept assertion*), where C is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$) (called *role assertion*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (called *equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. A *knowledge base* L is a finite set of axioms. For decidability, number restrictions in L are restricted to simple abstract roles $R \in \mathbf{R}_A$ [11].

The syntax of $\mathcal{SHIF}(\mathbf{D})$ is as the above syntax of $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the *atleast* and *atmost* constructors limited to 0 and 1.

Example 2.1 An online store (such as *amazon.com*) may use a description logic knowledge base to classify and characterize its products. For example, suppose that (1) textbooks are books, (2) personal computers and laptops are mutually exclusive electronic products, (3) books and electronic products are mutually exclusive products, (4) objects on offer are products, (5) every product has at least one related product, (6) only products are related to each other, (7) *tb_ai* and *tb_lp* are textbooks, (8) which are related to each other, (9) *pc_ibm* and *pc_hp* are personal computers, (10) which are related to each other, and (11) *ibm* and *hp* are providers for *pc_ibm* and *pc_hp*, respectively. These relationships are expressed by the following description logic knowledge base L_1:

(1) $\mathit{Textbook} \sqsubseteq \mathit{Book}$;
(2) $\mathit{PC} \sqcup \mathit{Laptop} \sqsubseteq \mathit{Electronics}$; $\mathit{PC} \sqsubseteq \neg \mathit{Laptop}$;
(3) $\mathit{Book} \sqcup \mathit{Electronics} \sqsubseteq \mathit{Product}$; $\mathit{Book} \sqsubseteq \neg \mathit{Electronics}$;
(4) $\mathit{Offer} \sqsubseteq \mathit{Product}$;
(5) $\mathit{Product} \sqsubseteq\ \geq 1\, \mathit{related}$;
(6) $\geq 1\, \mathit{related} \sqcup \geq 1\, \mathit{related}^- \sqsubseteq \mathit{Product}$;
(7) $\mathit{Textbook}(\mathit{tb\_ai})$; $\mathit{Textbook}(\mathit{tb\_lp})$;
(8) $\mathit{related}(\mathit{tb\_ai}, \mathit{tb\_lp})$;
(9) $\mathit{PC}(\mathit{pc\_ibm})$; $\mathit{PC}(\mathit{pc\_hp})$;
(10) $\mathit{related}(\mathit{pc\_ibm}, \mathit{pc\_hp})$;
(11) $\mathit{provides}(\mathit{ibm}, \mathit{pc\_ibm})$; $\mathit{provides}(\mathit{hp}, \mathit{pc\_hp})$.

2.2 Semantics

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with respect to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (*abstract*) *domain* $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $C \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$). The mapping $\cdot^{\mathcal{I}}$ is extended to all concepts and roles as usual [8].

The *satisfaction* of a description logic axiom F in the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with respect to $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$, denoted $\mathcal{I} \models F$, is defined as follows: (1) $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; (2) $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$; (3) $\mathcal{I} \models \mathrm{Trans}(R)$ iff $R^{\mathcal{I}}$ is transitive; (4) $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$; (5) $\mathcal{I} \models R(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$; (6) $\mathcal{I} \models U(a, v)$ iff $(a^{\mathcal{I}}, v^{\mathbf{D}}) \in U^{\mathcal{I}}$; (7) $\mathcal{I} \models a = b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$; and (8) $\mathcal{I} \models a \neq b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The interpretation $\mathcal{I}$ *satisfies* the axiom F, or $\mathcal{I}$ is a *model* of F, iff $\mathcal{I} \models F$. The interpretation $\mathcal{I}$ *satisfies* a knowledge base L, or $\mathcal{I}$ is a *model* of L, denoted $\mathcal{I} \models L$, iff $\mathcal{I} \models F$ for all $F \in L$. We say that L is *satisfiable* (resp., *unsatisfiable*) iff L has a (resp., no) model. An axiom F is a *logical consequence* of L, denoted $L \models F$, iff every model of L satisfies F. A negated axiom $\neg F$ is a *logical consequence* of L, denoted $L \models \neg F$, iff every model of L does not satisfy F.

3 Fuzzy $\mathcal{SHIF}(\mathrm{D})$ and $\mathcal{SHOIN}(\mathrm{D})$

Even though the literature contains several previous approaches to fuzzy description logics [33, 30, 24, 25], only recently fuzzy description logics for the Semantic Web have been explored. In particular, recent work by Straccia introduces a fuzzy description logic with concrete domains (with reasoning techniques based on a mixture of completion rules and bounded mixed integer programming) [26] as well as a fuzzy extension of $\mathcal{SHOIN}(\mathbf{D})$ (without reasoning machinery) [27]. Closely related to the latter is the work by Stoilos et al. [23], which combines the description logic $\mathcal{SHIN}$ with fuzzy set theory for the Semantic Web.

For our combination of fuzzy description logics with fuzzy description logic programs under the answer set semantics, we use a simple fuzzy extension of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which allows only for fuzzy concept and fuzzy role assertions, and which is intuitively based on a mapping to several layers of ordinary concepts and roles in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$. As an important advantage, reasoning in this fuzzy extension can immediately be reduced to reasoning in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$, and thus directly be implemented on top of standard technology for reasoning in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$.

3.1. Syntax

We assume a set of *truth values* $TV = \{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$. A *fuzzy concept assertion* has the form $C(a) \geq v$, where C is a concept in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$, $a \in \mathbf{I}$, and $v \in TV$. Similarly, a *fuzzy role assertion* has the form $R(a,b) \geq v$ (resp., $U(a,s) \geq v$), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$), $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$, and s is a data value), and $v \in TV$. Informally, $C(a) \geq v$ (resp., $R(a,b) \geq v$ and $U(a,s) \geq v$) encodes that the truth value of $C(a)$ (resp., $R(a,b)$ and $U(a,s)$) is at least v. A *fuzzy description logic knowledge base* $KB = (L, F)$ consists of an ordinary description logic knowledge base L and a finite set of fuzzy concept and fuzzy role assertions F.

Example 3.1 A simple fuzzy description logic knowledge base $KB = (L, F)$ is given by L as in Example 2.1 and $F = \{\mathit{Inexpensive}(\mathit{pc\_ibm}) \geq 0.6, \mathit{Inexpensive}(\mathit{pc\_hp}) \geq 0.9\}$. Here, F encodes the different degrees of membership of PCs by IBM and HP to the fuzzy concept $\mathit{Inexpensive}$.

3.2. Semantics

We define the semantics of fuzzy description logic knowledge bases by a mapping to ordinary description logic knowledge bases in $\mathcal{SHIF}(\mathrm{D})$ resp. $\mathcal{SHOIN}(\mathrm{D})$. For $v \in TV$, a *v-layer* of an ordinary description logic knowledge base L, denoted L^v, is obtained from L by replacing every concept C and role R (resp., U) by C^v and R^v (resp., U^v). The *ordinary equivalent* to a finite set of fuzzy concept and fuzzy role assertions F, denoted $F^\star$, is obtained from F by replacing every $C(a) \geq v$ (resp., $R(a,b) \geq v$ and $U(a,s) \geq v$) by $C^v(a)$ (resp., $R^v(a,b)$ and $U^v(a,s)$). The *ordinary equivalent* to a fuzzy description logic knowledge base $KB = (L, F)$, denoted $KB^\star$, is defined as

$$\begin{aligned}
&\textstyle\bigcup_{v \in TV,\, v>0} L^v \cup F^\star \cup \\
&\{A^v \sqsubseteq A^{v'} \mid A \in \mathbf{A}, v \in TV, v > 2/n, v' = v{-}1/n\} \cup \\
&\{R^v \sqsubseteq R^{v'} \mid R \in \mathbf{R}_A, v \in TV, v > 2/n, v' = v{-}1/n\} \cup \\
&\{U^v \sqsubseteq U^{v'} \mid U \in \mathbf{R}_D, v \in TV, v > 2/n, v' = v{-}1/n\}\,.
\end{aligned}$$

A fuzzy description logic knowledge base $KB = (L, F)$ is *satisfiable* iff its ordinary equivalent is satisfiable. We say that F among $C(a) \geq v$, $R(a,b) \geq v$, and $U(a,s) \geq v$ is a *logical consequence* of KB, denoted $KB \models F$, iff $C^v(a)$, $R^v(a,b)$, and $U^v(a,s)$, respectively, are logical consequences of $KB^\star$. We say that $\neg F$ is a *logical consequence* of KB, denoted $KB \models \neg F$, iff $\neg C^v(a)$, $\neg R^v(a,b)$, and $\neg U^v(a,s)$, respectively, are logical consequences of $KB^\star$.

4. Fuzzy Description Logic Programs

In this section, we introduce fuzzy dl-programs. We first define negation and conjunction strategies. We then introduce the syntax of fuzzy dl-programs, and we finally define the semantics of positive, stratified, and general fuzzy dl-programs in terms of a least model semantics, an iterative least model semantics, and the answer set semantics.

4.1. Combination Strategies

We assume a set of *truth values* $TV = \{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$. We assume a set of *negation* and *conjunction strategies*, which are functions $\ominus\colon TV \to TV$ and $\otimes\colon TV \times TV \to TV$. For $v \in TV$, we call $\ominus v$ the *negation* of v. For $v_1, v_2 \in TV$, we call $v_1 \otimes v_2$ the *conjunction* of v_1 and v_2. As usual, we assume that the negation and conjunction strategies have some natural algebraic properties. In particular, we assume that every negation strategy $\ominus$ is *antitonic* (that is, $v_1 \leq v_2$ implies $\ominus v_1 \geq \ominus v_2$) and satisfies the properties that $\ominus 0 = 1$ and $\ominus 1 = 0$. Furthermore, we assume that every conjunction strategy $\otimes$ is *commutative* (that is, $v_1 \otimes v_2 = v_2 \otimes v_1$), *associative* (that is, $(v_1 \otimes v_2) \otimes v_3 = v_1 \otimes (v_2 \otimes v_3)$), *monotonic* (that is, $v_1 \leq v_1'$ and $v_2 \leq v_2'$ implies $v_1 \otimes v_2 \leq v_1' \otimes v_2'$), and satisfies the properties that $v \otimes 1 = v$ and $v \otimes 0 = 0$.

Example 4.1 An example of a negation strategy is given by $\ominus v = 1 - v$, while two examples of conjunction strategies are given by $v_1 \otimes v_2 = \min(v_1, v_2)$ and $v_1 \otimes v_2 = v_1 \cdot v_2$.

4.2. Syntax of Fuzzy DL-Programs

We assume a function-free first-order vocabulary Φ with nonempty finite sets of constant and predicate symbols, and a set $\mathcal{X}$ of variables. A *term* is a constant symbol from Φ or a variable from $\mathcal{X}$. If p is a predicate symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are terms, then $p(t_1, \ldots, t_k)$ is an *atom*. A *literal* is an atom a or a default-negated atom $\mathit{not}\ a$. A *normal fuzzy rule* r has the form

$$\begin{aligned}
a \leftarrow_{\otimes_0} b_1 \wedge_{\otimes_1} b_2 \wedge_{\otimes_2} \cdots \wedge_{\otimes_{k-1}} b_k \wedge_{\otimes_k} \\
\mathit{not}_{\ominus_{k+1}} b_{k+1} \wedge_{\otimes_{k+1}} \cdots \wedge_{\otimes_{m-1}} \mathit{not}_{\ominus_m} b_m \geq v,
\end{aligned} \tag{1}$$

where $m \geq k \geq 0$, $a, b_{k+1}, \ldots, b_m$ are atoms, $b_1, \ldots, b_k$ are either atoms or truth values from TV, $\otimes_0, \ldots, \otimes_{m-1}$ are conjunction strategies, $\ominus_{k+1}, \ldots, \ominus_m$ are negation strategies, and $v \in TV$. Observe here that $b_1, \ldots, b_k$ may also be truth values from TV, which will be very useful in the definition of the Gelfond-Lifschitz transformation for the answer set semantics. We refer to a as the *head* of r, denoted $H(r)$, while the conjunction $b_1 \wedge_{\otimes_1} \ldots \wedge_{\otimes_{m-1}} \mathit{not}_{\ominus_m} b_m$ is the *body* of r. We denote by $B(r)$ the set of body literals $B^+(r) \cup B^-(r)$, where $B^+(r) = \{b_1, \ldots, b_k\}$ and $B^-(r) = \{b_{k+1}, \ldots, b_m\}$. A *normal fuzzy program* P is a finite set of normal fuzzy rules. We say that P is *positive* iff no rule in P contains default-negated atoms.

Informally, a fuzzy dl-program consists of a fuzzy description logic knowledge base L and a generalized normal fuzzy program P, which may contain queries to L. In such a query, it is asked whether a certain description logic axiom or its negation logically follows from L or not. Formally, a *dl-query* $Q(\mathbf{t})$ is either

(a) of the form $C(t)$, where C is a concept and t is a term; or

(b) of the form $R(t_1, t_2)$, where R is a role and t_1, t_2 are terms.

A *dl-atom* has the form $DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{t})$, where each S_i is a concept or role, p_i is a unary resp. binary predicate symbol, $Q(\mathbf{t})$ is a dl-query, and $m \geq 0$. We call $p_1, \ldots, p_m$ its *input predicate symbols*. Intuitively, $\uplus$ increases S_i by the extension of p_i. A *fuzzy dl-rule* r is of the form (1), where any b_i in the body of r may be a dl-atom. A *fuzzy dl-program* $KB = (L, P)$ consists of a fuzzy description logic knowledge base L and a finite set of dl-rules P. We say $KB = (L, P)$ is *positive* iff P is positive. *Ground terms*, *atoms*, *literals*, etc., are defined as usual. The *Herbrand base* of P, denoted HB_P, is the set of all ground atoms with standard predicate symbols that occur in P and constant symbols in Φ. We denote by $\mathit{ground}(P)$ the set of all ground instances of fuzzy dl-rules in P relative to HB_P.

Example 4.2 In the running example, the following fuzzy dl-rules encode PCs that are not in the description logic knowledge base and say which of them are brand-new. Furthermore, they express that (i) electronic products that are not brand-new are on offer with degree of truth 1, (ii) a customer who needs a product on offer buys this product with degree of truth 0.7, and (iii) a customer who needs an inexpensive product buys this product with degree of truth 0.3:

$$pc(pc\_1) \geq 1; \quad pc(pc\_2) \geq 1; \quad pc(pc\_3) \geq 1;$$
$$brand\_new(pc\_1) \geq 1; \quad brand\_new(pc\_2) \geq 1;$$
$$\mathit{offer}(X) \leftarrow_{\otimes} DL[PC \uplus pc; \mathit{Electronics}](X) \wedge_{\otimes} \mathit{not}_{\ominus}\, brand\_new(X) \geq 1;$$
$$buy(C, X) \leftarrow_{\otimes} needs(C, X) \wedge_{\otimes} \mathit{offer}(X) \geq 0.7;$$
$$buy(C, X) \leftarrow_{\otimes} needs(C, X) \wedge_{\otimes} DL[\mathit{Inexpensive}](X) \geq 0.3.$$

4.3 Models of Fuzzy DL-Programs

We first define Herbrand interpretations and the truth of fuzzy dl-programs in Herbrand interpretations. In the sequel, let $KB = (L, P)$ be a fuzzy dl-program.

The *Herbrand base* of P, denoted HB_P, is the set of all ground atoms with a standard predicate symbol that occurs in P and constant symbols in Φ. An *interpretation I relative to P* is a mapping $I\colon HB_P \rightarrow TV$. We write $\boldsymbol{HB}_P$ to denote the interpretation I such that $I(a) = 1$ for all $a \in HB_P$. For interpretations I and J, we write $I \subseteq J$ iff $I(a) \leq J(a)$ for all $a \in HB_P$, 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 $a \in HB_P$. The truth value of $a \in HB_P$ under L, denoted $I_L(a)$, is defined as $I(a)$. The truth value of a ground dl-atom $a = DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{c})$ under L, denoted $I_L(a)$, is defined as the maximal truth value $v \in TV$ such that $L \cup \bigcup_{i=1}^{m} A_i(I) \models Q(\mathbf{c}) \geq v$, where

- $A_i(I) = \{S_i(\mathbf{e}) \geq I(p_i(\mathbf{e})) \mid I(p_i(\mathbf{e})) > 0\}$.

We say that I is a *model* of a ground fuzzy dl-rule r of the form (1) under L, denoted $I \models_L r$, iff

$$I_L(a) \geq v \otimes_0 I_L(b_1) \otimes_1 I_L(b_2) \otimes_2 \cdots \otimes_{k-1} I_L(b_k) \otimes_k \ominus_{k+1} I_L(b_{k+1}) \otimes_{k+1} \cdots \otimes_{m-1} \ominus_m I_L(b_m),$$

and of a fuzzy dl-program $KB = (L, P)$ denoted $I \models KB$, iff $I \models_L r$ for all $r \in \mathit{ground}(P)$.

4.4 Positive Fuzzy DL-Programs

We now define positive fuzzy dl-programs, which are informally fuzzy dl-programs without default negation. We show that they have a unique least model, which defines their canonical semantics. Formally, a fuzzy dl-program $KB = (L, P)$ is *positive* iff P is "not"-free.

For ordinary positive programs, as well as positive dl-programs KB, the intersection of two models of KB is also a model of KB. The following theorem shows that a similar result holds for positive fuzzy dl-programs KB.

Theorem 4.3 *Let $KB = (L, P)$ be a positive fuzzy dl-program. If the interpretations $I_1, I_2 \subseteq \boldsymbol{HB}_P$ are models of KB, then $I = I_1 \cap I_2$ is also a model of KB.*

Proof. We have to show that I is a model of every $r \in \mathit{ground}(P)$ under L. Consider any $r \in \mathit{ground}(P)$. Since I_j ($j \in \{1, 2\}$) is a model of KB, and thus of every $r \in \mathit{ground}(P)$ under L, the truth value of r's head under I_j and L is at least the truth value of r's body under I_j and L. Since r contains no default-negated atoms, every conjunction strategy in r is monotonic, and $I \subseteq I_j$, the truth value of r's body under I_j ($j \in \{1, 2\}$) and L is at least the truth value of r's body under I and L. Hence, the truth value of r's head under I_j ($j \in \{1, 2\}$) and L, and thus also under I and L, is at least the truth value of r's body under I and L. That is, I is a model of r under L. $\Box$

As an immediate corollary of this result, every positive fuzzy dl-program KB has a unique least model, denoted M_{KB}, which is contained in every model of KB.

Corollary 4.4 *Let $KB = (L, P)$ be a positive fuzzy dl-program. Then, a unique model $I \subseteq \boldsymbol{HB}_P$ of KB exists such that $I \subseteq J$ for all models $J \subseteq \boldsymbol{HB}_P$ of KB.*

4.5 Stratified Fuzzy DL-Programs

We next define stratified fuzzy dl-programs, which are informally composed of hierarchic layers of positive fuzzy dl-programs that are linked via default negation. Like for ordinary stratified programs, as well as stratified dl-programs, a minimal model can be defined by a number of iterative least models, which naturally describes the semantics of stratified fuzzy dl-programs.

For any fuzzy dl-program $KB = (L, P)$, let DL_P denote the set of all ground dl-atoms that occur in $ground(P)$. An *input atom* of $a \in DL_P$ is a ground atom with an input predicate of a and constant symbols in Φ.

A *stratification of* $KB = (L, P)$ (*with respect to* DL_P) is a mapping $\lambda\colon HB_P \cup DL_P \to \{0, 1, \ldots, k\}$ such that

(i) $\lambda(H(r)) \geq \lambda(a)$ (resp., $\lambda(H(r)) > \lambda(a)$) for each $r \in ground(P)$ and $a \in B^+(r)$ (resp., $a \in B^-(r)$), and

(ii) $\lambda(a) \geq \lambda(a')$ for each input atom a' of each $a \in DL_P$,

where $k \geq 0$ is the *length* of λ. For $i \in \{0, \ldots, k\}$, let

$$KB_i = (L, P_i) = (L, \{r \in ground(P) \mid \lambda(H(r)) = i\}),$$

and let HB_{P_i} (resp., $HB^{\star}_{P_i}$) be the set of all $a \in HB_P$ such that $\lambda(a) = i$ (resp., $\lambda(a) \leq i$).

A fuzzy dl-program $KB = (L, P)$ is *stratified* iff it has a stratification λ of some length $k \geq 0$. We define its iterative least models $M_i \subseteq \boldsymbol{HB}_P$ with $i \in \{0, \ldots, k\}$ as follows:

(i) M_0 is the least model of KB_0;

(ii) if $i > 0$, then M_i is the least model of KB_i such that $M_i|HB^{\star}_{P_{i-1}} = M_{i-1}|HB^{\star}_{P_{i-1}}$.

Then, M_{KB} denotes M_k. Observe that M_{KB} is well-defined, since it does not depend on a particular stratification λ (cf. Corollary 4.8). The following theorem shows that M_{KB} is in fact a minimal model of KB.

Theorem 4.5 *Let $KB = (L, P)$ be a stratified fuzzy dl-program. Then, M_{KB} is a minimal model of KB.*

Proof (sketch). The statement can be proved by induction along a stratification of KB. □

4.6 General Fuzzy DL-Programs

We now define the *answer set semantics* of general fuzzy dl-programs KB, which is reduced to the least model semantics of positive fuzzy dl-programs. We use a generalized transformation that removes all default-negated atoms. In the sequel, let $KB = (L, P)$ be a fuzzy dl-program.

The *fuzzy dl-transform* of P relative to L and an interpretation $I \subseteq \boldsymbol{HB}_P$, denoted P^I_L, is the set of all fuzzy dl-rules obtained from $ground(P)$ by replacing all default-negated atoms $not_{\ominus_j}\, a$ by the truth value $\ominus_j\, I_L(a)$.

Observe that (L, P^I_L) has no default-negated atoms anymore. Hence, (L, P^I_L) is a positive fuzzy dl-program, and by Corollary 4.4, has a least model.

Definition 1 Let $KB = (L, P)$ be a fuzzy dl-program. An *answer set* of KB is an interpretation $I \subseteq \boldsymbol{HB}_P$ such that I is the least model of (L, P^I_L).

The following result shows that, as desired, answer sets of a fuzzy dl-program KB are also minimal models of KB.

Theorem 4.6 *Let KB be a fuzzy dl-program, and let M be an answer set of KB. Then, M is a minimal model of KB.*

Proof. Let I be an answer set of $KB = (L, P)$. Since I is the least model of (L, P^I_L), it is immediate that I is also a model of KB. We now show that I is also a minimal model of KB. Towards a contradiction, suppose that there exists a model J of KB such that $J \subseteq I$. Then, since every conjunction strategy $\otimes$ in KB is monotonic, and every negation strategy $\ominus$ is antitonic, it follows that J is also a model of (L, P^I_L), which contradicts I being a minimal model of (L, P^I_L). Thus, I is a minimal model of KB. □

The next theorem shows that positive and stratified fuzzy dl-programs have at most one answer set, which coincides with the canonical minimal model M_{KB}.

Theorem 4.7 *Let KB be a* (*a*) *positive* (*resp.,* (*b*) *stratified*) *fuzzy dl-program. Then, M_{KB} is the only answer set of KB.*

Proof. (a) An answer set of KB is an interpretation $I \subseteq \boldsymbol{HB}_P$ such that I is the least model of (L, P^I_L). As KB is a positive dl-program, P^I_L coincides with $ground(P)$. Hence, $I \subseteq \boldsymbol{HB}_P$ is an answer set of KB iff $I = M_{KB}$.

(b) Let λ be a stratification of KB of length $k \geq 0$. Suppose that $I \subseteq \boldsymbol{HB}_P$ is an answer set of KB. That is, I is the least model of (L, P^I_L). Hence,

- $I|HB^{\star}_{P_0}$ is the least of all models $J \subseteq \boldsymbol{HB}^{\star}_{P_0}$ of $(L, {P_0}^I_L)$;
- if $i > 0$, then $I|HB^{\star}_{P_i}$ is the least among all models $J \subseteq \boldsymbol{HB}^{\star}_{P_i}$ of $(L, {P_i}^I_L)$ with $J|HB^{\star}_{P_{i-1}} = I|HB^{\star}_{P_{i-1}}$.

It thus follows that:

- $I|HB^{\star}_{P_0}$ is the least of all models $J \subseteq \boldsymbol{HB}^{\star}_{P_0}$ of KB_0;
- if $i > 0$, then $I|HB^{\star}_{P_i}$ is the least among all models $J \subseteq \boldsymbol{HB}^{\star}_{P_i}$ of KB_i with $J|HB^{\star}_{P_{i-1}} = I|HB^{\star}_{P_{i-1}}$.

Hence, KB is consistent, and $I = M_{KB}$. Since the above line of argumentation also holds in the converse direction, it follows that $I \subseteq \boldsymbol{HB}_P$ is an answer set of KB iff KB is consistent and $I = M_{KB}$. □

Since the answer sets of a stratified fuzzy dl-program KB are independent of the stratification λ of KB, we thus obtain that M_{KB} is independent of λ.

Corollary 4.8 *Let KB be a stratified fuzzy dl-program. Then, M_{KB} does not depend on the stratification of KB.*

5 Fixpoint Semantics

In this section, we give fixpoint characterizations for the answer set of positive and stratified fuzzy dl-programs, and we show how to compute it by finite fixpoint iterations.

The answer set of an ordinary positive resp. stratified normal logic program KB, as well as of a positive resp. stratified dl-program KB has a well-known fixpoint characterization in terms of an immediate consequence operator T_{KB}, which generalizes to fuzzy dl-programs. This can be exploited for a bottom-up computation of the answer set of a positive resp. stratified fuzzy dl-program.

For a fuzzy dl-program $KB = (L, P)$, we define the operator T_{KB} on the subsets of $\boldsymbol{HB}_P$ as follows. For every $I \subseteq \boldsymbol{HB}_P$ and $a \in HB_P$, let $T_{KB}(I)(a)$ be defined as the maximum of v subject to $r \in ground(P)$, $H(r) = a$, and v being the truth value of r's body under I and L. Note that if there is no such rule r, then $T_{KB}(I)(a) = 0$.

The following lemma shows that, if KB is positive, then T_{KB} is monotonic, which follows from the fact that each conjunction strategy in $ground(P)$ is monotonic.

Lemma 5.1 *For any positive fuzzy dl-program $KB = (L, P)$, the operator T_{KB} is monotonic (that is, $I \subseteq I' \subseteq \boldsymbol{HB}_P$ implies $T_{KB}(I) \subseteq T_{KB}(I')$).*

Proof. Let $I \subseteq I' \subseteq \boldsymbol{HB}_P$. Consider any $r \in ground(P)$. Then, since every conjunction strategy $\otimes$ in r is monotonic, it follows that the truth value of r's body under I' and L is at least the truth value of r's body under I and L. This shows that $T_{KB}(I) \subseteq T_{KB}(I')$. □

Since every monotonic operator has a least fixpoint, also T_{KB} has one, denoted $lfp(T_{KB})$. Moreover, $lfp(T_{KB})$ can be computed by finite fixpoint iteration (given finiteness of TV, P, and the number of constant symbols in Φ).

For every $I \subseteq \boldsymbol{HB}_P$, we define $T^i_{KB}(I) = I$, if $i = 0$, and $T^i_{KB}(I) = T_{KB}(T^{i-1}_{KB}(I))$, if $i > 0$.

Theorem 5.2 *For every positive dl-program $KB = (L, P)$, it holds that $lfp(T_{KB}) = M_{KB}$. Furthermore,*

$$lfp(T_{KB}) = \bigcup_{i=0}^{n} T^i_{KB}(\emptyset) = T^n_{KB}(\emptyset), \text{ for some } n \geq 0.$$

We finally describe a fixpoint iteration for stratified dl-programs. Using Theorem 5.2, we can characterize the strong answer set M_{KB} of a stratified dl-program KB as follows. Let $\widehat{T}^i_{KB}(I) = T^i_{KB}(I) \cup I$, for all $i \geq 0$.

Theorem 5.3 *Let $KB = (L, P)$ be a fuzzy dl-program with stratification λ of length $k \geq 0$. Let $M_i \subseteq \boldsymbol{HB}_P$, $i \in \{-1, 0, \ldots, k\}$, be defined by $M_{-1} = \emptyset$, and $M_i = \widehat{T}^{n_i}_{KB_i}(M_{i-1})$ for $i \geq 0$, where $n_i \geq 0$ such that $\widehat{T}^{n_i}_{KB_i}(M_{i-1}) = \widehat{T}^{n_i+1}_{KB_i}(M_{i-1})$. Then, $M_k = M_{KB}$.*

6. Summary and Outlook

We have first defined a simple fuzzy extension of the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. We have then presented fuzzy dl-programs. We have defined the answer set semantics of general fuzzy dl-programs, and shown that it coincides with the canonical semantics of positive and stratified fuzzy dl-programs, which is given by a unique least model and an iterative least model semantics, respectively. We have also given a characterization of the canonical semantics of positive and stratified fuzzy dl-programs in terms of a fixpoint and an iterative fixpoint semantics.

An interesting topic for future research is to analyze the computational complexity of this approach. It appears that fuzzy dl-programs under the answer set semantics have the same complexity characterization as non-fuzzy dl-programs under the answer set semantics [4], when unary number encoding for truth values is used. Furthermore, it would be interesting to provide an implementation for fuzzy dl-programs under the answer set semantics, which seems to be possible by a reduction to dl-programs under the answer set semantics (along the lines already described in [15] for many-valued disjunctive logics programs). Finally, another topic for future research is to integrate more expressive fuzzy description logics into fuzzy dl-programs.

Acknowledgments. This work was supported by a Heisenberg Professorship of the German Research Foundation.

References

[1] T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, 1999.

[2] H. Boley, S. Tabet, and G. Wagner. Design rationale for RuleML: A markup language for Semantic Web rules. In *Proceedings SWWS-2001*, pp. 381–401, 2001.

[3] A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. In *Proceedings ICLP-1997*, pp. 391–405, 1997.

[4] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. In *Proceedings KR-2004*, pp. 141–151, 2004. Extended Report RR-1843-03-13, Institut für Informationssysteme, TU Wien, 2003.

[5] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Well-founded semantics for description logic programs in the Semantic Web. In *Proceedings RuleML-2004*, *LNCS* 3323, pp. 81–97, 2004.

[6] D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.

[7] M. Gelfond and V. Lifschitz. Classical negation in logic programs and deductive databases. *New Generation Computing*, 17:365–387, 1991.

[8] I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proceedings ISWC-2003*, *LNCS* 2870, pp. 17–29, 2003.

[9] I. Horrocks and P. F. Patel-Schneider. A proposal for an OWL Rules Language. In *Proceedings WWW-2004*, pp. 723–731, 2004.

[10] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. Web Semantics*, 1(1):7–26, 2003.

[11] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings LPAR-1999*, *LNCS* 1705, pp. 161–180, 1999.

[12] T. Lukasiewicz. Many-valued first-order logics with probabilistic semantics. In *Proceedings CSL-1998*, *LNCS* 1584, pp. 415–429, 1999.

[13] T. Lukasiewicz. Probabilistic and truth-functional many-valued logic programming. In *Proceedings ISMVL-1999*, pp. 236–241, 1999.

[14] T. Lukasiewicz. Many-valued disjunctive logic programs with probabilistic semantics. In *Proceedings LPNMR-1999*, *LNAI* 1730, pp. 277-289, 1999.

[15] T. Lukasiewicz. Fixpoint characterizations for many-valued disjunctive logic programs with probabilistic semantics. In *Proceedings LPNMR-2001*, *LNAI* 2173, pp. 336-350, 2001.

[16] T. Lukasiewicz. Probabilistic description logic programs. In *Proceedings ECSQARU-2005*, *LNCS* 3571, pp. 737–749. Springer, 2005. Extended version accepted for publication in *International Journal of Approximate Reasoning*.

[17] T. Lukasiewicz. Stratified probabilistic description logic programs. In *Proceedings URSW-2005*, pp. 87–97, 2005.

[18] T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Trans. Computat. Logic*, 2(3):289–337, 2001.

[19] C. Mateis. *A Quantitative Extension of Disjunctive Logic Programming*. Doctoral Dissertation, TU Wien, 1998.

[20] R. T. Ng and V. S. Subrahmanian. A semantical framework for supporting subjective and conditional probabilities in deductive databases. *J. Autom. Reasoning*, 10(2):191–235, 1993.

[21] R. T. Ng and V. S. Subrahmanian. Stable semantics for probabilistic deductive databases. *Inf. Comput.*, 110:42–83, 1994.

[22] D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artif. Intell.*, 94:7–56, 1997.

[23] G. Stoilos, G. B. Stamou, V. Tzouvaras, J. Z. Pan, and I. Horrocks. The fuzzy description logic $f\text{-}\mathcal{SHIN}$. In *Proceedings URSW-2005*, pp. 67–76, 2005.

[24] U. Straccia. A fuzzy description logic. In *Proceedings AAAI-1998*, pp. 594–599, 1998.

[25] U. Straccia. Reasoning within fuzzy description logics. *J. Artif. Intell. Res.*, 14:137–166, 2001.

[26] U. Straccia. Description logics with fuzzy concrete domains. In *Proceedings UAI-2005*, pp. 559–567. AUAI Press, 2005.

[27] U. Straccia. Towards a fuzzy description logic for the Semantic Web (preliminary report). In *Proceedings ESWC-2005*, *LNCS* 3532, pp. 167–181, 2005.

[28] U. Straccia. Fuzzy description logic programs. In *Proceedings IPMU-2006*, 2006.

[29] U. Straccia. Uncertainty and description logic programs over lattices. In Elie Sanchez, editor, *Capturing Intelligence: Fuzzy Logic and the Semantic Web*. Elsevier, 2006.

[30] C. B. Tresp and R. Molitor. A description logic for vague knowledge. In *Proceedings ECAI-1998*, pp. 361–365, 1998.

[31] A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *J. ACM*, 38(3):620–650, 1991.

[32] W3C. OWL web ontology language overview, 2004. W3C Recommendation (10 Feb. 2004). Available at www.w3.org/TR/2004/REC-owl-features-20040210/.

[33] J. Yen. Generalizing term subsumption languages to fuzzy logic. In *Proceedings IJCAI-1991*, pp. 472–477, 1991.

[34] M. H. van Emden. Quantitative deduction and its fixpoint theory. *J. Logic Program.*, 3(1):37–53, 1986.