A Fuzzy Semantics for Semantic Web Languages

Mauro Mazzieri[1] and Aldo Franco Dragoni[2]

[1] mauro.mazzieri@gmail.com
[2] Università Politecnica delle Marche
Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni (DEIT)
a.f.dragoni@univpm.it

Abstract. Although the model-theoretic semantics of the languages used in the Semantic Web are crisps, the need arise to extend them to represent fuzzy data, in the same way fuzzy logic extend first-order-logic. We will define a fuzzy counterpart of the RDF Model Theory for RDF (section 2) and RDF Schema (section 3). Last, we show how to implement the extended semantics in inference rules (section 4).

1 Knowledge representation on the web

The Semantic Web is an extension of the current web in which information is given well-defined meaning[1] by the use of knowledge representation (KR) languages.

The KR languages used (RDF, RDF Schema and OWL) have the characteristics that make them useful on the web[2]:

- the elements of the domain are represented by URI;
- there is no global coherence requirements, as local sources can make assertions independently without affecting each other's expressiveness.

The languages have the ability to describe, albeit not formally, much more than their semantics can express. Their model theory captures only a formal notion of meaning, captured by inference rules; the exact 'meaning' of a statement can depend on many factors, not all accessible to machine processing[3]. This feature can be useful to represent information from fields that require knowledge representation paradigms other than the FOL-like RDF Model Theory or the expressive Description Logic used by OWL. Amongst those paradigms there is fuzzy logic, to represent vague or ambiguous knowledge.

2 Fuzzy RDF

RDF has its own model-theoretic semantics, similar to that of first-order logic. To represent fuzzy data, we will define a syntactic and semantic extension of RDF, similar to the extension from first-order logic to fuzzy logic.

Even if fuzzy data can be simply seen as a juxtaposition of a triple and a number, the model-theoretic approach has well-known theoretical advantages.

We will try to be as plain as possible. Starting from RDF Syntax and RDF Model Theory, we will make as few changes as possible. In the rest of the paper, for the sake of brevity only the changes from RDF Semantics[3] are shown.

2.1 Syntax

The RDF syntax must be extended to add to the triple ⟨subject, predicate, object⟩ a value. Such a value can be taken as a real number in the interval [0, 1], but every bounded real interval will do.

This is not an extension from a 3-elements tuple to a 4-elements tuple as it may seem at a first glance. The added element has a syntactic nature different from the others: it is not an element of the domain of the discourse, but a property related to the formalism used by the language to represent uncertainty and vagueness.

The simple concrete syntax we define is as an extension of the EBNF of N-Triples as given in [4]. Our extension is given in table 1.

N-Triples is a line-based, plain text format for encoding an RDF Graph, used for expressing RDF test cases. A statement has the form `s p o.`, where `s`, `p` and `o` are respectively the subject, the predicate and the object of the statement. Our extended syntax add an optional prefix `n:` to a statement in N-triple notation, where n is a decimal number representing the fuzzy truth-value of the triple. The use of decimal numbers instead of real numbers is only a limitation of the syntax and does not undermine the discussion.

The term *triple*, used in the EBNF for N-Triple, is replaced with the more generic term *statement*. Triple and statement are often used in semantic web literature as a synonym, but we prefer to use the latter to avoid confusion between a plain RDF statement (made actually of three parts) and a fuzzy RDF statement (that, although is still a triple semantically, is made up of four elements).

The fuzzy *value* is defined as optional. This way, the syntax is backward-compatible; the intended semantics is that a statement with the form `s p o.` is equivalent to the statement `1: s p o.`. With such a (syntactic only) default, we could take an inference engine implementing fuzzy RDF, let it parse plain RDF statements, and get the same results of a conventional RDF inference engine. Furthermore, as it would be clear in the description of fuzzy RDF inference rules (section 4), even the complexity of the computation would be of the same order.

We will not give an abstract syntax, nor a RDF/XML based syntax, as they would not be useful. It can be shown that all "physical" data (i.e., data transmitted between host or processes) can be encoded using plain RDF reified statements. The extended syntax will be used only in the paper to write down the examples.

```
fuzzyNtripleDoc ::= line*
line              ::= ws* ( comment | statement )? eoln
comment           ::= '#' ( character − ( cr | lf ) )*
statement         ::= (value ws+)? subject ws+ predicate ws+ object ws* '.' ws*
value             ::= 1 | 0.[0–9]+
subject           ::= uriref | nodeID
predicate         ::= uriref
object            ::= uriref | nodeID | literal
uriref            ::= '<' absoluteURI '>'
nodeID            ::= '_:' name
literal           ::= langString | datatypeString
langString        ::= '"' string '"' ( '@' language )?
datatypeString    ::= '"' string '"' '^^' uriref
language          ::= [a–z]+ ('-' [a–z0–9]+ )*
                      encoding a language tag.
ws                ::= space | tab
eoln              ::= cr | lf | cr lf
space             ::= #x20 /* US-ASCII space - decimal 32 */
cr                ::= #xD /* US-ASCII carriage return - decimal 13 */
lf                ::= #xA /* US-ASCII line feed - decimal 10 */
tab               ::= #x9 /* US-ASCII horizontal tab - decimal 9 */
string            ::= character* (with escapes as defined in section Strings of [4])
name              ::= [A-Za-z][A–Za–z0–9]*
absoluteURI       ::= character+ (with escapes as defined in section URI References
                      of [4])
character         ::= [#x20–#x7E] /* US-ASCII space to decimal 126 */
```

Table 1. EBNF for Fuzzy N-Triples

2.2 Simple interpretation

The RDF Model Theory[3] is based on the concept of *extension*. An interpretation satisfies a triple `s p o.` if the couple formed by the interpretation of the subject and the interpretation of the object belongs to the extension of the interpretation of the property.

In this fuzzy counterpart, a couple ⟨subject, object⟩ has a membership degree to the extension of the predicate, given by the number prepended to the statement. The extension is not an ordinary set of couples anymore, but a fuzzy set of couples. In other words, a fuzzy RDF interpretation satisfies a statement `n: s p o.` if the membership degree of the couple, formed by the interpretation of the subject and the interpretation of the object, to the extension of the interpretation of the predicate, is greater or equal than n.

We have chosen not to make the mapping between vocabulary items and domain fuzzy. Instead, the membership of a resource to the domain is fuzzy. This is a step which poses some theoretical problems, in particular when we have to deal with properties in simple interpretations. In RDF interpretation, the property domain IP is a subset of the resource domain IR, so in fuzzy RDF

interpretations would be enough to make IP a fuzzy subset of IR; in simple interpretations, instead, there is no formal relation between IP and IR, so when the mapping IS from URI references to $(IR \cup IP)$ becomes fuzzy we need a further device. The chosen solution is to define a domain IDP for properties, so that IP is a fuzzy subset of IDP, and to modify the definition of IS to a mapping URI references $\in V \to (IR \cup IDP)$. RDF interpretations does not need IDP, as IP can be shown to be a fuzzy subset of IR.

Definition of a simple interpretation A *simple fuzzy interpretation I* of a vocabulary V is defined by:

1. A non empty set IR of resources, called the domain or universe of I
2. A non empty set IDP, called the property domain of I
3. A fuzzy subset IP of IDP, called the set of properties of I
4. A fuzzy mapping $IEXT : IP \to 2^{IR \times IR}$, i.e. the fuzzy set of pairs $\langle x, y \rangle$ with $x, y \in IR$.
5. A mapping IS from URI references $\in V \to (IR \cup IDP)$
6. A mapping IL from typed literals $\in V \to IR$
7. A distinguished subset $LV \subseteq IR$, called the set of literal values, which contains all the plain literals of V

The belonging of an element to the properties domain is strictly related to the use of such element as a property in a statement. Therefore, we have defined a membership degree to the property domain, intuitively related to the truth value of the statements in which the resource is used as a property.

2.3 Denotations for ground graphs

The next step is to define the semantic conditions an interpretation must satisfy in order to be a model for a graph. We state the semantic conditions that relate the membership degree of a couple ⟨subject, object⟩ to an extension and the truth of a fuzzy statement.

We will use the abbreviated Zadeh's notation $A(x) = n$, instead of $\mu_A(x) = n$, to state that the membership degree of the element x to the set A is equal to n [5].

Semantic conditions for ground graphs

- if E is a plain literal `aaa` $\in V$, then $I(E) =$ `aaa`
- if E is a plain literal `aaa@ttt` $\in V$, then $I(E) = \langle$`aaa`, `ttt`$\rangle$
- if E is a typed literal $\in V$, then $I(E) = IL(E)$
- if E is a URI reference $\in V$, then $I(E) = IS(E)$
- if E is a ground triple `n: s p o.`, then $I(E) =$ `true` if `s`, `p` and `o` $\in V$, $IP(I(\texttt{p})) \geq n$ and $IEXT(I(\texttt{p}))(\langle I(\texttt{s}), I(\texttt{o}) \rangle) \geq n$, otherwise $I(E) =$ `false`.
- if E is a ground RDF graph, than $I(E) =$ `false` if $I(E') =$ `false` for some triple $E' \in E$, otherwise $I(E) =$ `true`

Only the condition of truth and falsity of a ground statement in the interpretation is affected. The given formulation of the condition has as a consequence that a graph where the same statement appears more than once, with different membership degrees, is equivalent to a graph where the statement appears only once, with a membership degree equal to the maximum of the membership degrees.

Note that whether a statement is a model for a graph or not is not a fuzzy concept; it is either true or false. However, it could be interesting to compute the minimum and maximum membership degree to an extensions a couple must have in an interpretation to be a model of a given graph. This minimum degree has a role similar to the degree of truth of a statement in a knowledge base.

2.4 Simple entailment

The definition of simple interpretation is not affected. A set S of RDF graphs *(simply) entails* a graph E if every interpretation which satisfies every member of S also satisfies E.

Given a graph G and a triple $\langle s, p, o \rangle$, it could be interesting to compute the minimum and maximum value of n such that G entails `n: s p o.`. Those bounds must be taken in account when we compute the deductive closure of the graph, as it is not unique.

Section 2 of RDF Semantics [3] shows many lemmas that apply to simple interpretations. All of them retain their validity within fuzzy RDF Model Theory, making some adjustments in the proof of some of them. We will show these.

The *Empty Graph Lemma* can be shown using the same proof. The definition of an empty graph is the same as in plain RDF: an *empty graph* is a graph with no statements at all. It is important to note that an empty graph can not be defined as a graph with no not-zero-valued statements. Statements such as `0: s p o.`, although pretty useless, cannot be ignored, as the semantic requirement that `s`, `p` and `o` must belong to the graph's vocabulary still applies.

Subgraph Lemma, *Instance Lemma* and *Merging Lemma* retain both their validity and their proofs with the new semantics.

Interpolation Lemma, *Anonymity Lemma*, *Monotonicity Lemma* and *Compactness Lemma* make use in their proof of a way of constructing an interpretation of a graph using lexical items in the graph itself, the so called *Herbrand interpretation* [6]. To prove the lemmas, we need to construct a similar interpretation for a fuzzy graph.

The *(simple) Herbrand fuzzy interpretation* of G, written Herb(G), can be defined as follows.

- $LV_{\text{Herb}}(G)$ is the set of all plain literals in G;
- $IR_{\text{Herb}}(G)$ is the set of all names and blank nodes which occur in subject or object position of statements in G;
- $IDP_{\text{Herb}}(G)$ is the set of URI references which occur in the property position of statements in G;

- $IP_{\mathrm{Herb}(G)}(\mathtt{p})$ is the maximum of n for all statements in which $\mathtt{p}$ occur in property position;
- $IEXT_{\mathrm{Herb}(G)}(\langle \mathtt{s}, \mathtt{o}\rangle)$ is the maximum n for all the statements $\mathtt{n\colon s\ p\ o.}$ in G
- $IS_{\mathrm{Herb}}(G)$ and $IL_{\mathrm{Herb}}(G)$ are both identity mappings on the appropriate parts of the vocabulary of G.

Using this definition of Herbrand interpretation instead of that in Appendix A of [3], the proofs for cited lemmas still apply.

2.5 RDF Interpretation

RDF Semantic Conditions

- $IP(x) = IEXT(I(\mathtt{rdf\colon type}))(\langle x, I(\mathtt{rdf\colon Property}))$
- If "$\mathtt{xxx}$"$\wedge\wedge\mathtt{rdf\colon XMLLiteral} \in V$ and $\mathtt{xxx}$ is a well-typed XML literal string, then
  - $IL("\mathtt{xxx}" \wedge \wedge \mathtt{rdf\colon XMLLiteral})$ is the XML value of $\mathtt{xxx}$;
  - $IL("\mathtt{xxx}" \wedge \wedge \mathtt{rdf\colon XMLLiteral}) \in LV$;
  - $IEXT(I(\mathtt{rdf\colon type}))$
    $(\langle IL("\mathtt{xxx}" \wedge \wedge \mathtt{rdf\colon XMLLiteral}),$
    $I(\mathtt{rdf\colon XMLLiteral})\rangle) = 1$
- If "$\mathtt{xxx}$"$\wedge\wedge\mathtt{rdf\colon XMLLiteral} \in V$ and $\mathtt{xxx}$ is an ill-typed XML literal string, then
  - $IL("\mathtt{xxx}" \wedge \wedge \mathtt{rdf\colon XMLLiteral}) \notin LV$;
  - $IEXT(I(\mathtt{rdf\colon type}))$
    $(\langle IL("\mathtt{xxx}" \wedge \wedge \mathtt{rdf\colon XMLLiteral}),$
    $I(\mathtt{rdf\colon XMLLiteral})\rangle) = 0$

The first RDF semantic condition has the consequence that IP must be a subset of IR. Given such a fact, there is no more need of IDP, as it was for simple interpretation. IP can be directly defined as a fuzzy subset of IR.

The second and third conditions equal to see the well-formedness of an XML Literal as crisp truth-valued. We could conceive an external machinery that can be considered completely trustworthy as it classify an XML literal as well-formed or not.

RDF axiomatic triples By definition, we give axiomatic triples a unit truth value. Given the (syntactic) convention that a triple $\mathtt{s\ p\ o.}$ is equivalent to the fuzzy statement $\mathtt{1\colon s\ p\ o.}$, we can take the table of axiomatic triples of RDF in section 3.1 of [3] and copy it as-is as the table of axiomatic statements of fuzzy RDF.

3 Fuzzy RDF Schema

The path from RDF Schema to Fuzzy RDF Schema follows the same guidelines of the previous section.

The RDFS semantics is conveniently stated in terms of a new semantic construct: the *class* [3]. A class is a resource with a *class extension*, $ICEXT$, which represents a set of things in the universe which all have that class as the object of their `rdf:type` property. Thus, the definition of a *class* roots in the definition of *extension*.

In fuzzy RDF, extensions are fuzzy set of couples; in fuzzy RDFS, *class extensions* are fuzzy sets of domain's elements.

3.1 RDFS Interpretation

A RDFS interpretation define the domains for resources (IR), literals (IL) and literal values (LV) in terms of classes. In fuzzy RDFS they are fuzzy subdomains of IR.

We will give RDFS semantic conditions and axiomatic triples, then we will try to explain the more problematic definitions: domains/ranges (section 3.2) and subproperties/subclasses (section 3.3).

RDFS semantic conditions

- $ICEXT(y)(x) = IEXT(I(\mathtt{rdf:type}))(\langle x, y\rangle)$
  - $IC = ICEXT(I(\mathtt{rdfs:Class}))$
  - $IR = ICEXT(I(\mathtt{rdfs:Resource}))$
  - $IL = ICEXT(I(\mathtt{rdfs:Literal}))$
- $ICEXT(y)(u) \geq \min(IEXT(I(\mathrm{rdfs:domain}))(\langle x, y\rangle), IEXT(x)(\langle u, v\rangle))$
- $ICEXT(y)(u) \geq \min(IEXT(I(\mathrm{rdfs:range}))(\langle x, y\rangle), IEXT(x)(\langle u, v\rangle))$
- $IEXT(I(\mathtt{rdfs:subPropertyOf}))$ is transitive and reflexive on IP
- If $IEXT(\mathtt{rdfs:subPropertyOf})(\langle x, y\rangle) = n$, then $IP(x) \geq n$, $IP(y) \geq n$, $\min_{\langle a,b\rangle}\{1 - IEXT(x)(\langle a, b\rangle) + IEXT(y)(\langle a, b\rangle)\} \geq n$
- $IEXT(I(\mathtt{rdfs:subClassOf}))(\langle \mathrm{x}, I(\mathtt{rdfs:Resource})\rangle) = IC(x)$
- If $IEXT(\mathtt{rdfs:subClassOf})(\langle x, y\rangle) = n$, then $IC(x) \geq n$, $IC(y) \geq n$, $\min_a\{1 - IC(x)(a) + IC(y)(a)\} \geq n$.
- $IEXT(I(\mathtt{rdfs:subClassOf}))$ is transitive and reflexive on IC
- $IEXT(I(\mathtt{rdfs:subPropertyOf}))(\langle \mathrm{x}, I(\mathtt{rdfs:member})\rangle) = ICEXT(I(\mathtt{rdfs:ContainerMembershipProperty}))(x)$
- $ICEXT(I(\mathtt{rdfs:Datatype}))(x) = IEXT(I(\mathtt{rdfs:subClassOf}))(\langle x, I(\mathtt{rdfs:Literal})\rangle)$

RDFS axiomatic triples As for RDF axiomatic triples, fuzzy RDFS axioms are the same of plain RDFS, from section 4.2 of RDF Semantics [3].

3.2 Domains and ranges

The semantic condition on domains looks quite complicated. To explain it, we will proceed by grades.

In plain RDF Schema, if $\langle x, y \rangle \in IEXT(I(\texttt{rdfs:domain}))$ and $\langle u, v \rangle \in IEXT(x)$ then $u \in ICEXT(y)$.

In fuzzy set theory, let R be a fuzzy relation on $X \times Y$. Then the domain is defined as $\text{dom}(R)(x) = \sup_y R(x, y)$ [7], i.e. the least upper bound of $R(x, y)$ for all y.

In fuzzy RDFS, we have to deal both with a fuzzy notion of domain, and with a fuzzy assignment of a domain to a property.

Let consider a resource $\texttt{u}$ and a class $\texttt{y}$. For each property $\texttt{x}$, we take the minimum between $IEXT(I(\texttt{rdfs:domain}))(\langle x, y \rangle)$ and $IEXT(x)(\langle u, v \rangle)$. Then, following the original RDFS condition, $ICEXT(y)(u)$ must be greater or equal than this value.

The previous condition must hold for every property $\texttt{x}$, so it's equivalent to state that must be taken the maximum value.

The conditions for ranges are analogous.

3.3 Subproperties and subclasses

Subproperties and subclasses are fully analogous concepts. The set inclusion is between extensions for the former, between class extensions of the latter.

To define the semantics of $\texttt{subClassOf}$ and $\texttt{subPropertyOf}$, we need a relation of set inclusion between fuzzy sets that takes into account also the degree of the relation of inclusion itself. This relation must be transitive and reflexive.

Zadeh's definition of fuzzy subset [8][3] ($A \subseteq B \iff \forall x \in X \quad A(x) \leq B(x)$) is transitive and reflexive, but is not a fuzzy relation: either the set A is a subset of B, or not. What we need is instead a weaker fuzzy subset relation; a relation that reduces to the Zadeh's one when the subclass/subproperty relation has a unit truth value. It must also maintain the reflexivity and transitivity properties.

Dubois and Prade [7] define *weak inclusion* $\prec_\alpha$ as

$$ A \prec_\alpha B \iff x \in (\overline{A} \cup B)_\alpha \ \forall x \in X \ , $$

where α is a parameter and $(\cdot)_\alpha$ is the α-cut[4]. This relation is transitive only for $\alpha > \frac{1}{2}$.

Other definitions of weak inclusion make use of *inclusion grades*. An inclusion grade $I(A, B)$ is a scalar measure of the inclusion of the set A in the set B. In general, $A \subseteq_\alpha B$ iff $I(A, B) \geq \alpha$, where $\subseteq_\alpha$ denote a weak inclusion with inclusion grade α.

We have chosen to use the inclusion grade defined as [7]:

---

[3] Again, we use the abbreviation $A(x)$ for the membership function $\mu_A(x)$.

[4] The α-*cut* A_α of A is the set of all elements with a membership value to A greater than α, with $\alpha \in (0, 1]$ $\quad A_\alpha = \{x | A(x) \geq \alpha\}$

$$I(A, B) = \inf_{x \in X} \overline{(A \mid - \mid B)}(x)$$

where inf is the infimum and $\mid - \mid$ is the *bounded difference*[5].

When $A \subseteq B$, $I(A, B) = 1$ [7].

This inclusion grade could also be written as $I(A, B) = \inf_{x \in X}(1 - \max(0,\ A(x) - B(x))) = \inf_{x \in X} \min(1,\ (1 - A + B))$.

Furthermore, let's suppose that there is at least an x such that $A(x) > B(x)$. Then $I(A, B)$ could be written as $\inf_{x \in X}(1 - A + B)$. The semantic condition requires such measure to be greater than or equal to n, where n is the truth value of the statement. In this case the semantic condition reduces to

$$\inf_{x \in X}(1 - A + B) \geq n \ .$$

It could be interesting to ask how much this definition differs from the condition for classical fuzzy subsets, $A(x) \leq B(x)$.

If $A \subseteq B$, then $I(A, B) = 1$, so the semantic condition holds for any $n \in [0, 1]$.

Let's call $d(x)$ the difference $d(x) = A(x) - B(x)$, so that $1 - A + B = 1 - d$. We suppose that there is at least an x such that $A(x) > B(x)$, so $d(x)$ has at least a positive value. The semantic condition could then be written $\inf_{x \in X}(1 - d(x)) \geq n$. The maximum positive value of the difference d equal to $1 - n$.

As n is the truth value of the statement that asserts the relation of subproperty or subclass, and $1 - n$ represent the lack of truth of the same statement, we can conclude that the maximum allowable positive difference between $A(x)$ and $B(x)$ is equal to the lack of truth on the subproperty or subclass relation.

4   Fuzzy RDF entailment rules

RDF Model Theory's entailment rules [3] are all of the same form: add a statement to a graph when it contains triples conforming to a pattern. Each rule has only one or two antecedent statements and derive only one new inferred statement; either $P \vdash R$ or $P, Q \vdash R$.

Given the way fuzzy RDF semantics is defined, the corresponding inference rules for fuzzy RDF are analogous; only the fuzzy truth values of inferred statements must be computed. The simplest possible choice that respect the semantics is:

- With rules as $P \vdash Q$, having only one antecedent, the truth value of the consequent Q is taken to be the same of the antecedent P.
- With rules as $P, Q \vdash R$, the truth value of R is the minimum between the truth values of P and Q.

The inference rules for RDF/RDFS are shown in table 2. They were derived from the rules used by the Sesame[10] forward-chaining inferencer.

---

[5] $\forall x \in X, \quad (A \mid - \mid B)(x) = \max(0,\ A(x) - B(x))$ [9]

Sesame is a generic architecture for storing and querying RDF and RDF Schema. It makes use of a forward-chaining inferencer to compute and store the closure of its knowledge base whenever a transaction adds data to the repository[11]. Sesame applies RDF-MT inference rules in a optimized way, making use of the dependencies between them to eliminate most redundant inferencing steps.

To obtain a fuzzy RDF storage and inference tool it is only a matter of modify Sesame RDF-MT inferencer, making it compute the correct truth values for inferred statements, and to extend the underlying storage to make room for a truth value (i.e., a number) for each statement.

This shows how a simple proof-of-concept fuzzy RDF inferencer is easy to implement. The starting point is the code base of an inference engine that implements the RDF model theory.

It can be shown that an inference engine implementing such rules is correct: all its rules are *valid*, in the sense that a graph entails any larger graph that is obtained by applying the rules to the original graph. There is no formal proof that it is also complete, but there is not such a proof for plain RDF Model Theory inference rules either.

References

1. Hendler, J., Lassila, O., Berners-Lee, T.: The semantic web. Scientific American (2001) 28–37
2. Berners-Lee, T.: What the semantic web can represent. W3C design issues, World Wide Web Consortium (September 1998)
3. Hayes, P.: RDF Semantics. W3C recommendation, World Wide Web Consortium (10 February 2004)
4. Grant, J., Beckett, D.: RDF test cases. W3C recommendation, World Wide Web Consortium (2004)
   http://www.w3.org/TR/rdf-testcases/.
5. Zadeh, L.A.: A fuzzy set theoretic interpretation of linguistic hedges. Journal of Cybernetics **2** (1972) 4–34
6. Goldfarb, W.D., ed.: Logical Writings of Jacques Herbrand. Harvard University Press, Cambridge (1971)
7. Dubuois, D., Prade, H.: Fuzzy sets and Systems. Academic Press, New York, NJ (1980)
8. Zadeh, L.A.: Fuzzy sets. Information and Control (1965) 338–353
9. Zadeh, L.A.: Calculus of Fuzzy Restrictions. In: Fuzzy Sets and Their Applications to Cognitive and Decision Processes. Academic Press, New York (1975) 1–39
10. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and quering RDF and RDF Schema. In Horrocks, I., Handler, J., eds.: Proceedings of the first International Semantic Web Conference (ISWC 2002), Sardinia, Italy (2002) 54–68
11. Broekstra, J., Kampman, A.: Inferencing and truth maintenance in RDF Schema: exploring a naive practical approach. In: Workshop on Practical and Scalable Semantic Systems (PSSS) 2003. Second International Semantic Web Conference (ISWC), Sanibel Island, Florida, USA (2003)

#	antecedents	consequent
1	iii: xxx aaa yyy	iii: aaa rdf:type rdf:Property
2.1	iii: xxx aaa yyy jjj: aaa rdfs:domain zzz	kkk: xxx rdf:type zzz where kkk = min(iii, jjj)
2.2	iii: aaa rdfs:domain zzz jjj: xxx aaa yyy	kkk: xxx rdf:type zzz where kkk = min(iii, jjj)
3.1	iii: xxx aaa uuu jjj: aaa rdfs:range zzz	kkk: uuu rdf:type zzz where kkk = min(iii, jjj)
3.2	iii: aaa rdfs:range zzz jjj: xxx aaa uuu	kkk: uuu rdf:type zzz where kkk=min(iii, jjj)
4a	iii: xxx aaa yyy	jjj: xxx rdf:type rdfs:Resource
4b	iii: xxx aaa uuu	iii: uuu rdf:type rdfs:Resource
5a.1	iii: aaa rdfs:subPropertyOf bbb jjj: bbb rdfs:subPropertyOf ccc	kkk: aaa rdfs:subPropertyOf ccc where kkk=min(iii, jjj)
5a.2	iii: bbb rdfs:subPropertyOf ccc jjj: aaa rdfs:subPropertyOf bbb	kkk: aaa rdfs:subPropertyOf ccc where kkk=min(iii, jjj)
5b	iii: xxx rdf:type rdf:Property	iii: xxx rdfs:subPropertyOf xxx *reflexivity of rdfs:subPropertyOf*
6.1	iii: xxx aaa yyy jjj: aaa rdfs:subPropertyOf bbb	kkk: xxx bbb yyy where kkk=min(iii, jjj)
6.2	iii: aaa rdfs:subPropertyOf bbb jjj: xxx aaa yyy	kkk: xxx bbb yyy where kkk=min(iii, jjj)
7a	iii: xxx rdf:type rdfs:Class	iii: xxx rdfs:subClassOf rdfs:Resource
7b	iii: xxx rdf:type rdfs:Class	iii: xxx rdfs:subClassOf xxx *reflexivity of rdfs:subClassOf*
8.1	iii: xxx rdfs:subClassOf yyy jjj: yyy rdfs:subClassOf zzz	kkk: xxx rdfs:subClassOf zzz where kkk=min(iii, jjj)
8.2	iii: yyy rdfs:subClassOf zzz jjj: xxx rdfs:subClassOf yyy	kkk: xxx rdfs:subClassOf zzz where kkk=min(iii, jjj)
9.1	iii: xxx rdfs:subClassOf yyy jjj: aaa rdf:type xxx	kkk: aaa rdf:type yyy where kkk=min(iii, jjj)
9.2	iii: aaa rdf:type xxx jjj: xxx rdfs:subClassOf yyy	kkk: aaa rdf:type yyy where kkk=min(iii, jjj)
10	iii: xxx rdf:type rdfs:ContainerMembershipProperty	iii: xxx rdfs:subPropertyOf rdfs:member
11	iii: xxx rdf:type rdfs:Datatype	jjj: xxx rdfs:subClassOf rdfs:Literal
X1	iii: xxx rdf:_* yyy	jjj: rdf:_* rdf:type rdfs:ContainerMembershipProperty *This is an extra rule for list membership* *properties (_1, _2, _3, ...). The RDF MT* *does not specify a production for this.*

Table 2. Fuzzy RDF inference rules