A Fuzzy RDF Semantics to Represent Trust Metadata

Mauro Mazzieri
Università Politecnica delle Marche
mauro.mazzieri@gmail.com

Abstract

The need for fuzzy knowledge bases arises from many application fields, such as trust management.

To represent fuzzy data, a syntactic and semantic extension of RDF is proposed. The syntax adds to each triple a truth value. The semantic for Fuzzy RDF and Fuzzy RDF Schema allows to derive truth values for derived statements as well.

Fuzzy RDF/RDFS does not aim to become a standard extension to plain RDF/RDFS. It is conceived to be used only inside the applications. Fuzzy data can be transferred as standard RDF using reification. The extra semantic can be brought back using rules.

1. Introduction

Semantic Web Languages, such as RDF, RDF Schema and OWL, can be considered as "webized" knowledge representation languages. Among their characteristics, there is the ability to describe much more than their semantics can express. This feature is useful to represent informations from fields that require knowledge representation paradigms other than the FOL-like RDF Model Theory.

Section 2 shows some of the application fields where fuzzy data could be useful. Section 3 and section 4 respectively describe the syntax and semantic of RDF and RDF Schema. Section 5 explains how to translate from/to plain RDF. Section 6 gives the example of fuzzy trust metadata. Section 7 draws the conclusions and looks at future works.

2. The need for fuzziness

A fuzzy knowledge base is useful in many application fields. As it is known from the classical theory of fuzzy logic, natural language has a big deal of imprecision and vagueness; *meaning in a natural language is a matter of degree*[1].

In the semantic web, the need for fuzzy data arises at two different levels. First, there are data from everyday's life. There are many examples of vague classifications: *old* people (by age), *heavy* commodities (by weight), and so on. In the web's context, another example is the characterization of multimedia pieces: classification by genre, valuation of similarity among them.

The second level is more connected to the very own nature of the semantic web. The practical management of semantic knowledge bases needs itself classifications that are fuzzy by nature, such as:

- the classification of information sources as *trustworthy*;
- the classification of data as *reliable*.

Terms such as *trustworth* are fuzzy. It means that they cannot be sharply defined. However, as humans, we make sense out of this information, and use it in decision making[1].

The fuzzy concept of *trustworth* depends on a reputation value that must be collected with some means. The ongoing research on trust management in semantic web and peer-to-peer network has not given a definitive answer on how to do that. See section 6 for a short review of current research.

3. Fuzzy RDF

To fully represent fuzzy data, we need a syntactic and semantic extension of RDF. Even if fuzzy data can be simply seen as a juxtaposition of a triple and a number, a complete model theory has well-known theoretichal advantages. The formal semantic gives to fuzzy knowledge representation in the semantic web a theoretical substrate that makes possible, via the notion of entailment, to state whether an inference mechanism is valid or not, and to know whether a given (inferred) statement is a member of the deductive closure of a knowledge base or not.

The approach will be minimalistic. Starting from RDF Syntax and RDF Model Theory, the smaller set of changes will be made.

The definitions will be followed by an explanation of the choices made. Before proceeding further, a re-read of *RDF Semantics*[2] is strongly suggested.

3.1. Syntax

The syntax must be extended from a triple ⟨subject, predicate, object⟩ to a couple ⟨value, triple⟩, adding to the triple a "value".

Formally, a fuzzy RDF statement is a couple $\langle E, n\rangle$, where E is a RDF statement and n a real number in the interval $[0,1]$.

A more formal syntax will not be given. As noted in section 5, all data must be read and written as plain RDF reified statements.

The syntax used in the examples of the rest of the document is an extension of N-Triples[3]. The extension consists in a prefix `n:`, where n is a real number, to state the fuzzy truth-value of the triple.

For backward compatibility, a statement `s p o.` is considered to be equivalent to the fuzzy statement `1: s p o.`. With such a syntactic default, it can be shown that a fuzzy RDF inference engine is equivalent, both in its results and in its order of magnitude of performance indicators, to a conventional RDF inference engine.

From here, the term *triple*, often used in semantic web literature as a synonym of statement, will be systematically replaced with the latter. Actually a fuzzy RDF statement is not a triple anymore, as it is made up of four elements: the elements of the triple and the value.

3.2. Simple interpretation

RDF Model Theory[2] 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, the value that the extended syntax attaches to the triple represents the membership grade of the couple ⟨subject, object⟩ to the extension of the predicate; so, in fuzzy RDF the extension is not an ordinary set of couples anymore, but a fuzzy set of couples.

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.

The mapping between vocabulary items and domain is not 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, *IP* is a subset of *IR*, so in fuzzy RDF interpretations would be enough to make *IP* a fuzzy subset of *IR*. In simple interpretation, 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 a to modify the definition of *IS* to a mapping URI references $\in V \rightarrow (IR \cup IDP)$. 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 $IEXT : IP \rightarrow 2^{IR \times IR}$, i.e. the fuzzy set of pairs $\langle x, y\rangle$ with $x, y \in IR$.
5. A mapping *IS* from URI references $\in V \rightarrow (IR \cup IDP)$
6. A mapping *IL* from typed literals $\in V \rightarrow IR$
7. A distinguished subset $LV \subseteq IR$, called the set of literal values, which contains all the plain literals of V

Even in simple interpretation the two fundamental principles of fuzzy RDF are outstanding: a membership degree of the couple ⟨subject, object⟩ to the extension, and the membership degree of an element of the properties domain to the properties set.

3.3. Denotations for ground graphs

We will use an abbreviated notation $A(x) = n$ to state that the membership degree of the element x to the set A is equal to n. In Zadeh's notation, this would be written as $\mu_A(x) = n$.

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 $I(E) = \langle \texttt{aaa}, \texttt{ttt}\rangle$
- 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.

Whether a statement is a model for a graph or not is not a fuzzy concept. Anyway, it could be interesting to compute the minimum 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.

3.4. 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.

Section 2 of RDF Semantics[2] shows many lemmata that apply to simple interpretations. All of them retain its validity within fuzzy RDF Model Theory, making some adjustments in the proof of some of them.

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

The Subgraph Lemma, the Instance Lemma and the Merging Lemma maintain both their validity. Their proofs are not affected by the changed definitions.

The proof of the Interpolation Lemma, the Anonymity Lemma, the Monotonicity Lemma, and the Compactness Lemma, use a way of constructing an interpretation of a graph by using lexical items in the graph itself, the so called Herbrand interpretation. To prove the lemma, 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.

- $LV_{\mathrm{Herb}}(G)$ is the set of all plain literals in G;
- $IR_{\mathrm{Herb}}(G)$ is the set of all names and blank nodes which occur in subject or object position in a statement in G;
- $IDP_{\mathrm{Herb}}(G)$ 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 `p` occur in property position;
- $IEXT_{\mathtt{Herb}(G)}(\langle \mathtt{s}, \mathtt{o} \rangle)$ is the maximum n for all the statements `n: s p o.` in G
- $IS_{\mathtt{Herb}}(G)$ and $IL_{\mathtt{Herb}}(G)$ are both identity mappings on the appropriate parts of the vocabulary of G.

With the substitution of this definition of Hebrand interpretation to that in Appendix A of [2], the proofs for cited lemmata still apply.

3.5. RDF interpretation

RDF semantic conditions

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

The RDF semantic conditions have 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. The extern *oracle* can be consider as completely trustworthy as it classifies an XML literal as well-formed or not.

RDF axiomatic triples By definition, axiomatic triples are absolutely true, so their truth is equal to 1. Given the (syntactic) convention that a triple `s p o.` is equivalent to the fuzzy statement `1: s p o.`, we can take the table of axiomatic triples of RDF in section 3.1 of [2] and copy it as-is as the table of axiomatic statements of Fuzzy RDF.

4. 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, a "class'[2]. A class is a resource with a *class extension*, 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 set of domain's elements.

4.1. RDFS Interpretation

The domains *IR*, *IL* and *LV* are defined in term of classes. This make necessary to define non-fuzzy super-domains, of which *IR*, *IL* and *LV* are fuzzy subsets.

RDFS semantic conditions

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

RDFS axiomatic triples As for RDF axiomatic triples, fuzzy RDFS axioms are the same of plain RDFS, from section 4.2 of RDF Semantics[2].

4.2. Domains and ranges

The semantic condition on domains looks quite compicated. 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 $\text{dom}(R)(x) = \sup_y R(x,y)$[4].

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

Lets consider a resource `u` and a class `y`. For each property `x`, let's take the minimun 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 `x`, so it's equivalent to state that must be taken the maximum value.

The conditions for ranges are analogous.

4.3. Subproperties and subclasses

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

To define the semantics of `subClassOf` e `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.

The Zadeh's definition of fuzzy subset,

$$A \subseteq B \iff \forall x \in X \quad A(x) \leq B(x) ,$$

is transitive and reflexive, but is not a fuzzy relation: either the set A is a subset of B, or not. What we need instead is weaker fuzzy subset relation, that reduces to this when the subclass/subpropery relation has a unity truth value. It must also maintain the reflexivity and transitivity properties.

Dubois and Prade[4] defined *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. 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 $I(A,B) \geq \alpha$.

The chosen definition use the inclusion grade

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

Recalling that the *bounded difference* $|-|$ is defined as

$$\forall x \in X, \quad (A|-|B)(x) = \max(0, A(x) - B(x)) \ ,$$

and noting that $\inf_{x\in X}(1 - \max(0,A(x) - B(x))) = \min(1, \inf_{x\in X}(1-A+B))$, this inclusion grade could also be written as

$$I(A,B) = \min(1, \inf_{x\in X}(1-A+B)) \ .$$

The semantic condition require such measure to be greater then or equal to n. As n is lesser then or uqual to 1, the condition reduce to

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

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

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 maximun positive value of the difference d equal to $1-n$.

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

5. Traslating from/to plain RDF

Fuzzy RDF is not a W3C standard, nor it aims to become a standard. Its full use is limited to the application that generates, gets and uses fuzzy data.

Fuzzy RDF syntax could never be used outside the user applications. The semantics is usefult to give a teoretical treatment to local data processing involving fuzzy metadata, not to propose an extension to W3C standards.

To move around fuzzy statements, they must be encoded as reified triples. The resource representing the reification of the triple is related, by an appropriate property defined by the ontology of the application domain, to the fuzzy value. This transposition loses the extra semantic implication given by the fuzzy model theory, that could be brought back using rules. The rules could be embedded in the data, using the RDF-encoding of a rules language.

Given a Fuzzy RDF graph, its deductive closure can be computed in a two step procedure: compute the deductive closure of its standard-compliant RDF counterpart, then apply mechanical inference rules to carry over the fuzzy membership values.

The next logical step would be to encode those rule in a rule language and to make an explicit reference to that in the ontology: the intended meaning of the properties that relate a reified triple to his truth value is that those rules must be used.

6. Fuzzy trust metadata

Fuzzy RDF could be used to carry trust metadata. The membership value of a statement to a graph represents is trustworthiness. The usual procedure to give such a value to each statement is in three steps. First, the statement is associated to a source by means of digital signatures. Then, the trustworthiness of the source is valuated using an appropriate trust metric. Finally, the trust metadata of the source is stored in each statement coming from that source.

The two most used techniques to measure the trust of asource are the *web of trust* approach and the *reputation* approach.

The "Web of Trust" is one of the ultimate goals of the Semantic Web. The research use results from social networks in physical world: people tend to trust who directly know and who knows someone they know. There are models that allow users to describe beliefs about others[5], using a base ontology, for example FOAF[6]. To generate locally-calculated reputation ratings from this semantic web social network we need algorithms, such that described in [7].

The reputation approach usually considers only sources directly known. The confidence grade is a function of the percentage of positive judgements got. This approach is similar to the *voting* assumed by Zadeh as the background philosophy of fuzzy logic semantics.

Works from peer-to-peer file sharing network, such as [8], also suggest a voting-based approach. Each servent mantains an experience repository, associating with each servent the number of succesful an unsuccesful downloads experienced. Each peer can then evaluate the reputation of another one polling other peers. The semantics of the digital reputation must allow for easily and consistently updating them at each interation.

6.1. Implementation of a fuzzy RDF inferencer

As a proof of concept, a fuzzy RDF inferencer was implemented.

The starting point is the code base of Sesame[9], a generic architecture for storing and querying RDF and RDF Schema.

Sesame allows querying at the RDF Schema level. It makes use of forward-chaining inferencer to compute and store the closure of its knowledge base[10]. Whenever a transaction adds data to the repository, the inferencer algorithm applies RDF Model Theory entailment rules in a optimized way, making use of the dependencies between them to eliminate most redundant inferencing steps.

RDF Model Theory entailment rules[2] are all of the same form: add a statement to a graph when it contains triples conforming to a patttern. Each rules has only one or two antecedent statements and derive only one new inferred statement. Given the way fuzzy RDF semantic is defined, the corresponding “fuzzy” inference rules are analogous; the fuzzy truth value of inferred statements is simply the minimum of the truth values of the antecedents.

It can be shown that such inference engine 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.

To obtain a fuzzy RDF storage and inference tools was then only a matter of modify Sesame RDF-MT inferencers, making them compute the correct thuth values for inferred statements, and to extend the underlying storage to make room for a truth value (i.e., a number) for each statement. To make some tests, it was also coded a parser for the extended N-Triples syntax explained in section 3.1.

7. Conclusions and future works

Trust metadata are fuzzy by nature, and the semantic web is capable to represent fuzzy data. It follows from these two facts that the semantic web, if from the one hand needs trust metadada for it to work, on the other hand can manage those data like all other kinds of information.

Fuzzy RDF data could form part of a semantic-web-enabled application. It would operate as follows:

- Collect the data, e.g. from the web or from a peer-to-peer network;
- assign a trust valutation to data sources or to data themself, extracting them from RDF reification-encoded fuzzy metadata or using one of the algorithms cited in section 6;
- use a fuzzy RDF inferencer to compute a closure of the knowledge base that correctly assigns a trust value to inferred statements;
- use the statement collection and their trust values in a fuzzy-aware application, e.g. to decide which of the data deserve to be represented in the user interface, or to draw conclusions using a fuzzy reasoner;
- share the conclusions with the world, encoding them in plain RDF with the aid of reification.

The work on fuzzy data in the semantic web could be seen as a part of a greater scenario: the link between semantic web and classical artificial intelligence techniques. Altough semantic web is not conceived as another kind of AI knowledge representation language, in fact it *is* a knowledge representation language, capable to describe things beyond the limit of what its semantic can represent. RDF could be seen as a general data-encoding language, a level higher then XML, but with the same aim to be of widespread use in everyday applications. The encoding of web data in a knowledge representation language allow artificial intelligence techniques to be applied on the huge world wibe web information source.

References

[1] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*. Boca Raton, Florida: Chapman & Hall/CRC, second ed., 2000.

[2] P. Hayes, “RDF Semantics,” W3C recommendation, World Wide Web Consortium, 10 February 2004. `http://www.w3.org/TR/rdf-mt/`.

[3] J. Grant and D. Beckett, “RDF test cases,” W3C recommendation, World Wide Web Consortium, February, 10 2004. `http://www.w3.org/TR/rdf-testcases/`.

[4] D. Dubuois and H. Prade, *Fuzzy sets and Systems*. New York, NJ: Academic Press, 1980.

[5] J. Golbeck, B. Parsia, and J. Hendler, “Trust networks on the semantic web,” in *Proceedings of Cooperative Information Agents 2003*, (Helsinki, Finland), August 27–29, 2003. `http://www.mindswap.org/papers/CIA03.pdf`.

[6] E. Dumbill, “XML watch: Finding friends with XML and RDF,” tech. rep., IBM Developers Works, June 2002.

[7] J. Golbeck and J. Hendler, “Accuracy of metrics for inferring trust and reputation,” in *Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management*, (Northamptonshire, UK), October 5-8 2004.

[8] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, “Choosing reputable servents in a p2p network,” in *Proc. of the Eleventh International World Wide Web Conference*, (Honolulu, Hawaii, USA), May 7–11 2002.
`http://seclab.dti.unimi.it/Papers/www02.ps`.

[9] J. Broekstra, A. Kampamn, and F. van Harmelen, “Sesame — a generic architecture for storing and quering RDF and RDF Schema,” in *Proceedings of the first International Semantic Web Conference (ISWC 2002)* (I. Horrocks and

J. Handler, eds.), (Sardinia, Italy), pp. 54–68, June 9–12 2002. See also `http://sesame.aidministrator.nl/`.

[10] J. Broekstra and A. Kampman, “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), October 20–24 2003.