Fuzzy Semantic Web Languages and Beyond

Umberto Straccia(✉)

ISTI - CNR, Pisa, Italy
straccia@isti.cnr.it

Abstract. The aim of this talk is to present the state of the art in representing and reasoning with fuzzy knowledge in Semantic Web Languages such as triple languages RDF/RDFS, conceptual languages of the OWL 2 family and rule languages. We further show how one may generalise them to so-called annotation domains, that cover also e.g. temporal and provenance extensions.

1 Introduction

Reasoning under fuzziness is growing in importance in Semantic Web research as recognised by a large number of research efforts in this direction [16,18]. *Semantic Web Languages* (SWL) are the languages used to provide a formal description of concepts, terms, and relationships within a given domain, among which the *OWL 2 family* of languages [10], *triple languages* RDF & RDFS [4] and *rule languages* (such as RuleML [6], Datalog$^{\pm}$ [5] and RIF [11]) are major players. While their syntactic specification is based on XML [22], their semantics is based on logical formalisms: briefly,

- RDFS is a logic having intensional semantics and the logical counterpart is ρdf;
- OWL 2 is a family of languages that relate to *Description Logics* (DLs);
- rule languages relate roughly to the *Logic Programming* (LP) paradigm, specifically *Datalog*;
- both OWL 2 and rule languages have an extensional semantics.

Fuzzyness. We recap that under *fuzziness* fall all those approaches in which statements (for example, "heavy rain") are true to some *degree*, which is taken from a truth space (usually $[0, 1]$). For instance, the grade of the sentence "heavy rain" may depend on the amount of rain is falling.[1] Often we may find rough definitions about rain types, such as:[2]

Rain. Falling drops of water larger than 0.5 mm in diameter. In forecasts, "rain" usually implies that the rain will fall steadily over a period of time;

[1] More concretely, the intensity of precipitation is expressed in terms of a precipitation rate R: volume flux of precipitation through a horizontal surface, i.e. $m^3/m^2s = ms^{-1}$. It is usually expressed in mm/h.

[2] http://usatoday30.usatoday.com/weather/wds8.htm.

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part I, LNAI 10350, pp. 3–8, 2017.
DOI: 10.1007/978-3-319-60042-0_1

Light rain. Rain falls at the rate of 2.6 mm or less an hour;
Moderate rain. Rain falls at the rate of 2.7 mm to 7.6 mm an hour;
Heavy rain. Rain falls at the rate of 7.7 mm an hour or more.

It is evident that such definitions are quite harsh and resemble a bivalent (two-valued) logic: e.g. a precipitation rate of 7.7 mm/h is a heavy rain, while a precipitation rate of 7.6 mm/h is just a moderate rain. This may be unsatisfactory, as quite naturally the more rain is falling, the more the sentence "heavy rain" is true and, vice-versa, the less rain is falling the less the sentence is true. A more fine grained way to define the various types of rains is illustrated in Fig. 1.

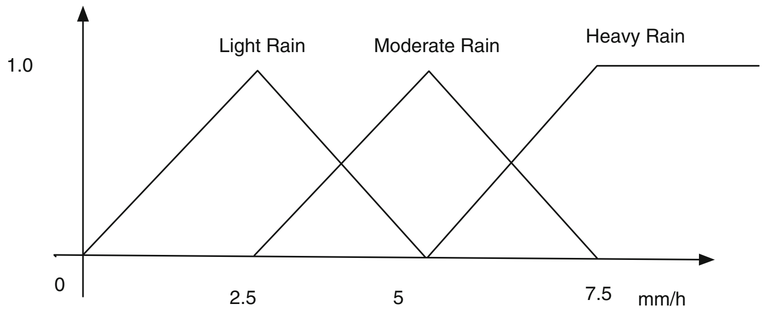

Fig. 1. Light, Moderate and Heavy Rain.

Light rain, moderate rain and heavy rain are called *Fuzzy Sets* in the literature and are characterised by the fact that membership is a matter of degree. Of course, the definition of fuzzy sets is frequently context dependent and subjective: e.g. the definition of heavy rain is quite different from heavy person and the latter may be defined differently among human beings.

From a logical point of view, a propositional interpretation maps a statement ϕ to a truth degree in $[0, 1]$, i.e. $\mathcal{I}(\phi) \in [0, 1]$. Fuzzy statements are truth-functional, that is, the degree of truth of every statement can be calculated from the degrees of truth of its constituents. For the sake of illustrative purpose, an example of truth functional interpretation of propositional statements is as follows:

$$\begin{aligned}\mathcal{I}(\phi \wedge \psi) &= \min(\mathcal{I}(\phi), \mathcal{I}(\psi))\\ \mathcal{I}(\phi \vee \psi) &= \max(\mathcal{I}(\phi), \mathcal{I}(\psi))\\ \mathcal{I}(\neg\phi) &= 1 - \mathcal{I}(\phi)\ .\end{aligned}$$

Fuzzy statements have the form $\langle \phi, r \rangle$, where $r \in [0, 1]$, which encodes that the degree of truth of ϕ is *greater or equal* r, i.e. fuzzy interpretation $\mathcal{I}$ *satisfies* a fuzzy statement $\langle \phi, r \rangle$, or $\mathcal{I}$ is a *model* of $\langle \phi, r \rangle$, denoted $\mathcal{I} \models \langle \phi, r \rangle$, iff $\mathcal{I}(\phi) \geq r$. A *fuzzy knowledge base* is a set of fuzzy statements and an interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a knowledge base, denoted $\mathcal{I} \models \mathcal{K}$, iff it satisfies each element in it. The *best entailment degree* of ϕ w.r.t. $\mathcal{K}$ (denoted $bed(\mathcal{K}, \phi)$), i.e. $bed(\mathcal{K}, \phi) = \sup \{r \mid \mathcal{K} \models \langle \phi, r \rangle\}$.

Annotation Domains. We have seen that fuzzy statements extend statements with an *annotation* $r \in [0, 1]$. Interestingly, we may further generalise this by allowing a statement being annotated with a value λ taken from a so-called *annotation domain* [23], which allow to deal with several domains (such as, fuzzy, temporal, provenance) and their combination, in a uniform way. Formally, let us consider a non-empty set L. Elements in L are our annotation values. For example, in a fuzzy setting, $L = [0, 1]$, while in a typical temporal setting, L may be time points or time intervals. In the annotation framework, an interpretation will map statements to elements of the annotation domain. Now, an *annotation domain* is an idempotent, commutative semi-ring $D = \langle L, \oplus, \otimes, \bot, \top \rangle$, where $\oplus$ is $\top$-annihilating. That is, for $\lambda, \lambda_i \in L$

1. $\oplus$ is idempotent, commutative, associative;
2. $\otimes$ is commutative and associative;
3. $\bot \oplus \lambda = \lambda$, $\top \otimes \lambda = \lambda$, $\bot \otimes \lambda = \bot$, and $\top \oplus \lambda = \top$;
4. $\otimes$ is distributive over $\oplus$, i.e.$\lambda_1 \otimes (\lambda_2 \oplus \lambda_3) = (\lambda_1 \otimes \lambda_2) \oplus (\lambda_1 \otimes \lambda_3)$.

We refer the reader to [23] for more details about annotation domains.

Talk Overview. We present here some salient aspects in representing and reasoning with fuzzy knowledge in Semantic Web Languages (SWLs) such as *triple languages* [4] (see, e.g. [17]), *conceptual languages* [10] (see, e.g. [9]) and *rule languages* (see, e.g. [13,15]). We refer the reader to [18,19] for an extensive presentation concerning fuzziness and semantic web languages. We then further show how one may generalise them to so-called annotation domains, that cover also e.g. temporal and provenance extensions (see, e.g. [23]).

2 Fuzzy Logic and Semantic Web Languages

We have seen in the previous section how to "fuzzyfy" a classical language such as propositional logic and FOL, namely fuzzy statements are of the form $\langle \phi, r \rangle$, where ϕ is a statement and $r \in [0, 1]$. The natural extension to SWLs consists then in replacing ϕ with appropriate expressions belonging to the logical counterparts of SWLs, namely *ρdf*, DLs and LPs, as we will illustrate next.

2.1 Fuzzy RDFS

The basic ingredients of *RDF* are *triples* of the form (s, p, o), such as $(umberto, likes, tomato)$, stating that *subject* s has *property* p with *value* o. In *RDF Schema* (RDFS), which is an extension of RDF, additionally some special keywords may be used as properties to further improve the expressivity of the language. For instance we may also express that the class of 'tomatoes are a subclass of the class of vegetables', $(tomato, \mathsf{sc}, vegetables)$, while Zurich is an instance of the class of cities, $(zurich, \mathsf{type}, city)$.

In *Fuzzy RDFS* (see [17] and references therein), triples are annotated with a degree of truth in $[0, 1]$. For instance, "Rome is a big city to degree 0.8" can be represented with $\langle (Rome, \mathsf{type}, BigCity), 0.8 \rangle$. More formally, *fuzzy triples* are expressions of the form $\langle \tau, r \rangle$, where τ is a RDFS triple (the truth value r may be omitted and, in that case, the value $r = 1$ is assumed).

Annotation Domains and RDFS. The generalisation to annotation domains is conceptual easy, as now one may replace truth degrees with annotation terms taken from an appropriate domain. For further details see [23].

2.2 Fuzzy DLs

Description Logics (DLs) [1] are the logical counterpart of the family of OWL languages. So, to illustrate the basic concepts of fuzzy OWL, it suffices to show the fuzzy DL case (see [2,9,18], for a survey). We recap that the basic ingredients are the descriptions of classes, properties, and their instances, such as

- $a{:}C$, meaning that individual a is an instance of concept/class C (here C is seen as a unary predicate);
- $(a, b){:}R$, meaning that the pair of individuals $\langle a, b\rangle$ is an instance of the property/role R (here R is seen as a binary predicate);
- $C \sqsubseteq D$, meaning that the class C is a subclass of class D.

In general, fuzzy DLs allow expressions of the form $\langle a{:}C, r\rangle$, stating that a is an instance of concept/class C with degree at least r, i.e. the FOL formula $C(a)$ is true to degree at least r. Similarly, $\langle C_1 \sqsubseteq C_2, r\rangle$ states a vague subsumption relationships. Informally, $\langle C_1 \sqsubseteq C_2, r\rangle$ dictates that the FOL formula $\forall x.C_1(x) \rightarrow C_2(x)$ is true to degree at least r. Essentially, *fuzzy DLs* are then obtained by interpreting the statements as fuzzy FOL formulae and attaching a weight n to DL statements, thus, defining so *fuzzy DL statements*.

So far, several *fuzzy* variants of DLs have been proposed: they can be classified according to (see [2,18])

- the description logic resp. ontology language that they generalize;
- the allowed fuzzy constructs;
- the underlying fuzzy logic;
- their reasoning algorithms and computational complexity results.

Annotation Domains and OWL. The generalisation to annotation domains is conceptual easy, as now one may replace truth degrees with annotation terms taken from an appropriate domain (see, e.g. [3,14]).

2.3 Fuzzy Rule Languages

The foundation of the core part of rule languages is *Datalog* [21], i.e. a Logic Programming Language (LP) [7] without n-ary function symbols ($n \geq 1$). In LP, the management of imperfect information has attracted the attention of many researchers and numerous frameworks have been proposed. Addressing all of them is almost impossible, due to both the large number of works published in this field (early works date back to early 80-ties [12]) and the different approaches proposed (see, e.g. [16,18,19]).

Basically, a Datalog program $\mathcal{P}$ is made out by a set of rules and a set of facts. *Facts* are ground *atoms* of the form $P(\boldsymbol{c})$. On the other hand rules are of the form

$$A(\boldsymbol{x}) \leftarrow \exists \boldsymbol{y}.\varphi(\boldsymbol{x}, \boldsymbol{y}) \ ,$$

where $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of n-ary predicates. A *query* is a rule and the *answer set* of a query q w.r.t. a set $\mathcal{K}$ of facts and rules is the set of tuples $\boldsymbol{t}$ such that there exists $\boldsymbol{t}'$ such that the instantiation $\varphi(\boldsymbol{t}, \boldsymbol{t}')$ of the query body is true in *minimal model* of $\mathcal{K}$, which is guaranteed to exists.

In the *fuzzy* case, rules and facts are as for the crisp case, except that now a predicate is annotated. An example of fuzzy rule defining good hotels may be the following:

$$\begin{aligned} \langle GoodHotel(x), s\rangle \leftarrow Hotel(x), \langle Cheap(x), s_1\rangle, \langle CloseToVenue(x), s_2\rangle, \\ \langle Comfortable(x), s_3\rangle, s := 0.3 \cdot s_1 + 0.5 \cdot s_2 + 0.2 \cdot s_3 \end{aligned} \qquad (1)$$

A *fuzzy query* is a fuzzy rule and, informally, the *fuzzy answer set* is the ordered set of weighted tuples $\langle \boldsymbol{t}, s\rangle$ such that all the fuzzy atoms in the rule body are true in the minimal model and s is the result of the scoring function f applied to its arguments. The existence of a minimal is guaranteed if the scoring functions in the query and in the rule bodies are *monotone* [18].

A rising problem is the problem to compute the top-k ranked answers to a query, without computing the score of all answers. This allows to answer queries such as "find the top-k closest hotels to the conference location". Solutions to this problem can be found in e.g. [8,15,20].

Annotation Domains and Rule Languages. The generalisation of fuzzy rule languages to the case in which an annotation $r \in [0, 1]$ is replaced with an annotation value λ taken from an annotation domain is straightforward and proceeds as for the other SWLs.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Bobillo, F., Cerami, M., Esteva, F., García-Cerdaña, À., Peñaloza, R., Straccia, U.: Fuzzy description logics in the framework of mathematical fuzzy logic. In: Cintula, C.N.P., Fermüller, C. (eds.) Handbook of Mathematical Fuzzy Logic, Volume 3. Studies in Logic, Mathematical Logic and Foundations, vol. 58, pp. 1105–1181. College Publications (2015). Chapter 16
3. Borgwardt, S., Peñaloza, R.: Description logics over lattices with multi-valued ontologies. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-11), pp. 768–773 (2011)
4. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, W3C (2004). http://www.w3.org/TR/rdf-schema/

5. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. J. Web Seman. **14**, 57–83 (2012)
6. The rule markup initiative. http://ruleml.org/index.html
7. Lloyd, J.W.: Foundations of Logic Programming. Springer, Heidelberg (1987)
8. Lukasiewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the semantic web. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS, vol. 4772, pp. 16–30. Springer, Heidelberg (2007). doi:10.1007/978-3-540-75410-7_2
9. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Seman. **6**, 291–308 (2008)
10. OWL 2 Web Ontology Language Document Overview. W3C (2009). http://www.w3.org/TR/2009/REC-owl2-overview-20091027/
11. Rule Interchange Format (RIF). W3C (2011). http://www.w3.org/2001/sw/wiki/RIF
12. Shapiro, E.Y.: Logic programs with uncertainties: a tool for implementing rule-based systems. In: Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-83), pp. 529–532 (1983)
13. Straccia, U.: Query answering in normal logic programs under uncertainty. In: Godo, L. (ed.) ECSQARU 2005. LNCS, vol. 3571, pp. 687–700. Springer, Heidelberg (2005). doi:10.1007/11518655_58
14. Straccia, U.: Description logics over lattices. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **14**(1), 1–16 (2006)
15. Straccia, U.: Towards top-k query answering in deductive databases. In: Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics (SMC-06), pp. 4873–4879. IEEE (2006)
16. Straccia, U.: Managing uncertainty and vagueness in description logics, logic programs and description logic programs. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) Reasoning Web. LNCS, vol. 5224, pp. 54–103. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85658-0_2
17. Straccia, U.: A minimal deductive system for general fuzzy RDF. In: Polleres, A., Swift, T. (eds.) RR 2009. LNCS, vol. 5837, pp. 166–181. Springer, Heidelberg (2009). doi:10.1007/978-3-642-05082-4_12
18. Straccia, U.: Foundations of Fuzzy Logic and Semantic Web Languages. CRC Studies in Informatics Series. Chapman & Hall (2013)
19. Straccia, U., Bobillo, F.: From fuzzy to annotated semantic web languages. In: Pan, J.Z., Calvanese, D., Eiter, T., Horrocks, I., Kifer, M., Lin, F., Zhao, Y. (eds.) Reasoning Web 2016. LNCS, vol. 9885, pp. 203–240. Springer, Cham (2017). doi:10.1007/978-3-319-49493-7_6
20. Straccia, U., Madrid, N.: A top-k query answering procedure for fuzzy logic programming. Fuzzy Sets Syst. **205**, 1–29 (2012)
21. Ullman, J.D.: Principles of Database and Knowledge Base Systems, vol. 1, 2. Computer Science Press, Potomac, Maryland (1989)
22. XML: W3C. http://www.w3.org/XML/
23. Zimmermann, A., Lopes, N., Polleres, A., Straccia, U.: A general framework for representing, reasoning and querying with annotated semantic web data. J. Web Seman. **11**, 72–95 (2012)