What Formalism(s) for the Semantic Web?

Hassan Aït-Kaci

hassan.ait-kaci@univ-lyon1.fr

ANR Chair of Excellence
CEDAR Project
LIRIS
Université Claude Bernard Lyon 1
France

Keynote

The Eleventh International Conference on Concept Lattices and Their Applications (CLA 2014)

Extended Abstract

The world is changing. The World Wide Web is changing. It started out as a set of purely notational conventions for interconnecting information over the Internet. The focus of information processing has now shifted from local disconnected disc-bound silos to Internet-wide interconnected clouds. The nature of information has also evolved. From raw uniform data, it has now taken the shape of semi-structured data and meaning-carrying so-called "Knowledge Bases." While it was sufficient to process raw data with structure-aware querying, it has now become necessary to process knowledge with contents-aware reasoning. Computing must therefore adapt from dealing with mere *explicit* data to inferring *implicit* knowledge. How to represent such knowledge and how inference therefrom can be made effective (whether reasoning or learning) is thus a central challenge among the many now facing the world wide web.

So called "ontologies" are being specified and meant to encode formally encyclopedic as well as domain-specific knowledge. One early (still on-going) such effort has been the Cyc system. It is a knowledge-representation system (using LISP syntax) that makes use of a set of varied reasoning methods, altogether dubbed "commonsense." A more recent formalism issued of Description Logic (DL)—*viz.*, the Web Ontology Language (OWL)—has been adopted as a W3C recommendation. It encodes knowledge using a specific standardized (XML, RDF) syntax. Its constructs are given a model-theoretic semantics which is usually realized operationally using tableau-based reasoning.[1] The point is that OWL is clearly designed for a specific logic and reasoning method. Saying that OWL is the most adequate interchange formalism for Knowledge Representation (KR) and automated reasoning (AR) is akin to saying that English is the best designed human language for facilitating information interchange among humans—notwithstanding the fact that it was simply imposed by the most recent pervasive ruling power, just as Latin was Europe's *Lingua Franca* for centuries.

Thus, it is fair to ask one's self a simple question: "***Is there, indeed, a single most adequate knowledge representation and reasoning method that can be such a norm?***"

I personally do not think so. In this regard, I share the general philosophy of Doug Lenat, Cyc's designer—although not the haphazard approach he has chosen to follow.[2]

If one ponders what characterizes an ontology making up a knowledge base, some specific traits most commonly appear. For example, it is universally acknowledged that, rather than being a general set of arbitrary formal logical statements describing some generic properties of "the world," a formal knowledge base is generally organized as a concept-oriented information structure. This is as important a change of perspective, just as object-oriented programming was with respect to traditional method-oriented programming. Thus, some notion of property "inheritance" among partially-ordered "concepts" (with an "*is-a*" relation) is a characterisc aspect of KR formalisms. In such a system, a concept has a straightforward semantics: its denotes of set of elements (its "instances") and the "is-a" relation denotes set inclusion. Properties attached to a concept denote information pertaining to all instances of this concept. All properties verified by a concept are therefore *inherited* by all its subconcepts.

Sharing this simple characteristic, formal KR formalisms have emerged from symbolic mathematics that offer means to reason with conceptual information, depending on mathematical apparatus formalizing inheritance and the nature of properties attached to concepts. In Description Logic, properties are called "roles" and denote binary relations among concepts. On the other hand, Formal Concept Analysis (FCA) uses an algebraic approach whereby an "is-

---

[1] Use of tableau methods is the case of the most prominent SW reasoner [6,5,7]. Systems using alternative reasoning methods must first translate the DL-based syntax of OWL into their own logic or RDF query processing. This may be costly [9] and/or incomplete [8].

[2] However, I may stand corrected in the future since knowledge is somehow fundamentally haphazard. My own view is that, even for dealing with a heterogenous world, I would rather favor mathematically *formal* representation and reasoning methods dealing with uncertainty and approximate reasoning, whether probabilistic, fuzzy, or dealing with inconsistency (*e.g.*, rough sets, paraconsistency).

a" ordering is automatically derived from propositional properties encoding the concepts they are attached to as bit vectors. A concept is associated an attribute with a boolean marker (1 or "true") if it possesses it, and with a (0 or "false") otherwise. The bit vectors are simply the rows of the "property matrix" relatings concepts to their attributes. This simple and powerful method, originally proposed by Rudolf Wille, has a dual interpretation when matching attributes with concepts possessing them. Thus, dually, it views attributes also as partially ordered bit vectors (as the columns of the binary matrix). An elegant Galois-connnection ensues that enables simple extraction of conceptual taxonomies (and their dual attribute-ordered taxonomies) from simple facts. Variations such as Relational Concept Analysis (RCA) offer more expressive, and thus more sophisticated, knowledge while preserving the essential algebraic properties of FCA. It has also been shown how DL-based reasoning (*e.g.*, OWL) can be enhanced with FCA.[3]

Yet another formalism for taxonomic attributed knowledge, which I will present in more detail in this presentation, is the Order-Sorted Feature ($\mathcal{OSF}$) constraint formalism. This approach proposes to see everything as an order-sorted labelled graph. Sorts are set-denoting and partially ordered with an inclusion-denoting "is-a" relation, and so form a conceptual taxonomy. Attributes, called "*features*," are function-denoting symbols labelling directed edges between sort-labelled nodes. Such $\mathcal{OSF}$ graphs are a straighforward generalization of algebraic First-Order Terms (FOTs) as used in Logic Programming (LP) and Functional Programming (FP). Like FOTs, they form a lattice structure with $\mathcal{OSF}$ graph *matching* as the partial ordering, $\mathcal{OSF}$ graph unification as infimum (denoting set intersection), and $\mathcal{OSF}$ graph generalization as supremum.[4] Both operations are very efficient. These lattice-theoretic properties are preserved when one endows a concept in a taxonomy with additional order-sorted relational and functional constraints (using logical conjunction for unification and disjunction for generalization of the attached constraints). These constraints are inherited down the conceptual taxonomy in such a way as to be incrementally enforceable as a concept becomes gradually refined.

The $\mathcal{OSF}$ system has been the basis of Constraint Logic Programming for KR and ontological reasoning (*viz.*, $\mathcal{LIFE}$) [2,1]. As importantly, $\mathcal{OSF}$ graph-constraint technology has been at work with great success in two essential areas of AI: NLP and Machine Learning:

- it has been a major paradigm in the field of Natural Language Processing (NLP) for a long time; notably, in so-called "Head-driven Phrase Stucture

---

[3] http://ijcai-11.iiia.csic.es/files/proceedings/T13-ijcai11Tutorial.pdf

[4] This supremum operation, however, does not (always) denote set union—as for FOT subsumption, it is is *not* modular (and hence neither is it distributive).

Grammar" (HPSG) and Unification Grammar (UG) technology [4]. This is indeed not surprising given the ease with which feature structure unification enables combining both syntactic and semantic information in a clean, declarative, and efficient way.[5]

- Similarly, while most of the attention in the $\mathcal{OSF}$ literature has been devoted to unification, its dual—namely, generalization—is just as simple to use, and computes the most specific $\mathcal{OSF}$ term that subsumes two given terms [3]. This operation is central in Machine Learning and with it, $\mathcal{OSF}$ technology lends itself to be combined with popular Data Mining techniques such as Support Vector Machines using frequency or probabilistic information.

In this presentation, I will give a rapid overview of the essential $\mathcal{OSF}$ formalism for knowledge representation along its reasoning method which is best formalized as order-sorted constraint-driven inference. I will also illustrate its operational efficieny and scalability in comparison with those of prominent DL-based reasoners used for the Semantic Web.

The contribution of this talk to answering the question in its title is that the Semantic Web effort should not impose *a priori* putting all our eggs in one single (untested) basket. Rather, along with DL, other viable alternatives such as the FCA and $\mathcal{OSF}$ formalisms, and surely others, should be combined for realizing a truly *semantic* web.

References

1. Aït-Kaci, H. Data models as constraint systems—a key to the Semantic Web. *Constraint Processing Letters 1* (November 2007), 33–88. (online).
2. Aït-Kaci, H., and Podelski, A. Towards a meaning of LIFE. *Journal of Logic Programming 16*, 3-4 (1993), 195–234. (online).
3. Aït-Kaci, H., and Sasaki, Y. An axiomatic approach to feature term generalization. In *Proceedings of European Conference on Machine Learning (ECML 2001)* (Freiburg, Germany, 2001), L. D. Raedt and P. Flach, Eds., LNAI 2167, Springer-Verlag, pp. 1–12. (online).
4. Carpenter, B. Typed feature structures: A generalization of first-order terms. In *Proceedings of the 1991 International Symposium on Logic Programming* (Cambridge, MA, USA, 1991), V. Saraswat and K. Ueda, Eds., MIT Press, pp. 187–201.
5. Motik, B., Shearer, R., and Horrocks, I. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research 36*, 1 (September 2009), 165–228. (online).
6. Shearer, R., Motik, B., and Horrocks, I. HermiT: A highly-efficient OWL reasoner. In *Proceedings of the 5th International Workshop on OWL Experiences and Directions* (Karlsruhe, Germany, October 2008), U. Sattler and C. Dolbear, Eds., OWLED'08, CEUR Workshop Proceedings. (online).

---

[5] `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.2021`

7. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics 5*, 2 (June 2007), 51–53. This is a summary; full paper: (online).
8. Stoilos, G., Cuenca Grau, B., and Horrocks, I. How incomplete is your semantic web reasoner? In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI 10)* (Atlanta, Georgia, USA, July 11–15, 2010), M. Fox and D. Poole, Eds., AAAI Publications, pp. 1431–1436. (online).
9. Thomas, E., Pan, J. Z., and Ren, Y. TrOWL: Tractable OWL 2 reasoning infrastructure. In *Proceedings of the 7th Extended Semantic Web Conference* (Heraklion, Greece, May-June 2010), L. Aroyo, G. Antoniou, E. Hyvnen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, Eds., ESWC'10, Springer-Verlag, pp. 431–435. (online).