Representation of Complex Expressions in RDF

Sébastien Ferré

Representation of Complex Expressions in RDF

Sébastien Ferré

IRISA, Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France
Email: ferre@irisa.fr

Abstract. Complex expressions, as used in mathematics and logics, account for a large part of human knowledge. It is therefore desirable to allow for their representation and search in RDF. We propose an approach[1] that fulfills three objectives: (1) the accurate representation of expressions in standard RDF, so that expressive search is made possible, (2) the automated generation of human-readable labels for expressions, and (3) the compatibility with legacy data (e.g., OWL/RDF, SPIN).

1 Introduction

Complex expressions account for a large part of human knowledge. Common instances of expressions are mathematical equations, logical formulae, regular expressions, or parse trees of natural language sentences. In the domain of the Semantic Web, they can be OWL axioms, SWRL rules, or SPARQL queries. It is therefore desirable to allow for their representation in RDF so that they can be mixed with other kinds of knowledge. For example, it should be possible to describe a theorem by its author, its discovery date, its informal description as a text, and its formal description as a mathematical and logical expression, all in RDF. An expected advantage of the formal representation of expressions is the ability to search those expressions by their content: e.g., *mathematical search* [1]. For example, we want to retrieve all expressions that are an integral in some variable x and whose body contains the sub-expression x^2. Correct answers are $\int x^2 + 1\,dx$ and $\int y^2 - y\,dy$. This example exhibits two difficulties in expression search: (1) to take into account the nested structure of expressions (x^2 is in the scope of the integral), (2) to abstract over the name of bound variables (x is bound by the integral $\int\ dx$).

Textual search methods that work by linearizing expressions cannot correctly account for the above difficulties [1]. In the above example, a textual search would have false positives such as $\int 2x\,dx = x^2 + c$ (x^2 is not in the scope of $\int$), and would have false negatives such as $\int y^2 - y\,dy$ (y instead of x). On the contrary, structured query languages [1] correctly account for them by reasoning directly on the structure of expressions, and by using *jokers* as place-holders for variables and sub-expressions. However, those query languages are limited to mathematical expressions, and are not interoperable with Semantic Web languages.

[1] A long version of this paper is available at http://hal.inria.fr/hal-00812197.

A number of RDF vocabularies have been defined to represent complex expressions with blank nodes and *ad-hoc* classes and properties: e.g., OWL/RDF for OWL class expressions, SPIN for SPARQL queries, and Robbins' proposal [4] for MathML strict content. Each of them is good in isolation, but they kind of reinvent the wheel each time, and a semantic web tool would have to be configured for each of them in order to nicely render the different kinds of expressions. A number of proposals have been made to use RDF for Mathematical Knowledge Management (MKM) [3], but few of them allow for complete RDF representations. In addition to Robbins' proposal, N3 has a syntax for expressions but its goal is to extend RDF in a non-standard way to express *computations*, whereas we are interested in the representation of *syntax trees* in standard RDF.

We propose to represent expressions as RDF containers, using custom *constructors* in addition to `rdf:Seq`, `rdf:Bag`, `rdf:Alt`: e.g., `[ a math:Power; rdf:_1 _:x; rdf:_2 2]` represents the expression x^2, using constructor `math:Power`. With a simple syntactic extension of Turtle and SPARQL for containers, the formula $\int x^2 + 1\, dx$ can be concisely represented as `math:Integral(math:Plus(math:Power(_:x,2),1),_:x)`; and the query that retrieves integrals in x whose body contains x^2 can be represented as `SELECT ?e WHERE { ?e is math:Integral(...math:Power(?x,2)...,?x) }` By annotating constructors with notation expressions (e.g., `math:Plus expr:hasNotation expr:LeftAssociativeInfixOperator("+",math:PlusPriority)`), *human-readable labels* can be automatically generated for each blank node representing an expression: e.g., `"∫ x^2+1 dx"` for the above example. To allow for the generation of such labels for legacy data, patterns of *ad-hoc* classes and properties can be mapped to *implicit constructors*. For example, OWL existential restrictions `[a owl:Restriction; owl:onProperty ?p; owl:someValuesFrom ?c]` can be mapped to expressions `owl:Some(?p,?c)`, by defining the implicit constructor `owl:Some`: `owl:Some expr:hasImplicitClass owl:Restriction ; expr:hasImplicitProperties rdf:Seq(owl:onProperty,owl:someValuesFrom)`. From there, both DL and Manchester notations can be generated as labels.

The above ideas have been implemented in Sewelis [2], a Semantic Web tool for the guided exploration and edition of RDF graphs, and applied to math formulas and OWL axioms. Expressions are only displayed and edited through their human-readable form (generation of labels), and can be searched through expressive query-based faceted search [2].

References

1. Altamimi, M.E., Youssef, A.S.: A math query language with an expanded set of wildcards. Mathematics in Computer Science 2(2), 305–331 (2008)
2. Ferré, S., Hermann, A.: Semantic search: Reconciling expressive querying and exploratory search. In: Aroyo, L., Welty, C. (eds.) Int. Semantic Web Conf. pp. 177–192. LNCS 7031, Springer (2011)
3. Lange, C.: Ontologies and languages for representing mathematical knowledge on the semantic web. Semantic Web Journal, IOS Press (To appear)
4. Robbins, A.: Semantic MathML (2009), http://straymindcough.blogspot.fr/2009/06/