Datatypes for Lists and Maps in RDF Literals

Olaf Hartig1,2  Gregory Todd Williams1  Michael Schmidt1  Ora Lassila1  Carlos Manuel Lopez Enriquez1  Bryan Thompson1
1Amazon Neptune Team, Amazon Web Services, Seattle, WA, USA  2Linköping University, Linköping, Sweden

What is the Problem?

In contrast to many other popular data representation forms and their query languages, RDF and SPARQL lack built-in support for generic types of composite values such as lists and maps. Instead, RDF introduces so-called containers and collections, which allow users to model composite values through a dedicated vocabulary on top of the core data model. Drawbacks:

- verbose representation, blows up storage footprint
- cumbersome (even tricky) to query such containers & collections in SPARQL
- manipulation of such containers & collections in SPARQL even more complex

Our Proposal

Dedicated datatypes for capturing lists and maps as literals; and corresponding extensions of SPARQL.

- Value space: sequences/functions over RDF terms and null (with some limitations)
- Lexical forms: Turtle based, including shorthands for literals; shorthands for nesting; superset of JSON
- Various functions to operate on these literals in SPARQL expressions

```
SELECT ?s (cdt:get(?list,1) AS ?v)
WHERE { ?s :performance ?list .
FILTER( cdt:size(?list) > 10 )
BIND( 1 AS ?x )
BIND( cdt:List(?x, ?x+1, ?x+2) AS ?list )
}
```

```
SELECT (FOLD(?name ORDER BY ?name) AS ?list)
?p foaf:name ?name .}
```

```
SELECT * { ?s :performance ?list .
UNFOLD(?list AS ?elm, ?pos) }
ORDER BY ?s ?pos
```

Resources

- Formal specification of the approach
- Comprehensive test suite that covers all aspects of the specification
- Two complete open source implementations integrating support for the approach into the RDF programming frameworks Apache Jena (Java) and Attean (Perl)

Broad Range of Use Cases

- Augmenting entities in a Knowledge Graph directly with corresponding embedding vectors
- Maintaining and operating over lists of all kinds; e.g., public transport timetables, series of measurements
- More direct interoperability between RDF graphs and Property Graphs that contain composite values
- Integration with other data ecosystems; e.g., queries over JSON and CSV data expressed directly within SPARQL, creation of JSON and CSV from SPARQL
- SPARQL as a bidirectional mapping language to describe mappings between RDF, JSON, and CSV