Chapter 15

Querying RDF Data

Faisal Alkhateeb  
Yarmouk University, Jordan

Jérôme Euzenat  
INRIA & LIG, France

ABSTRACT

This chapter provides an introduction to the RDF language as well as surveys the languages that can be used for querying RDF graphs. Then it reviews some of the languages that can be used for querying RDF and provides a comparison between these query languages.

INTRODUCTION

The Resource Description Framework (RDF) (Manola & Miller, 2004) is the W3C recommended data model for the representation of information about resources on the Web. Nowadays, more resources are annotated via RDF due to its simple data model, formal semantics, and a sound and complete inference mechanism. A query language that provides a range of querying paradigms is therefore needed.

The SPARQL query language (Prud’hommeaux & Seaborne, 2008) is a W3C recommendation developed in order to query an RDF knowledge base. The heart of a SPARQL query, the graph pattern, is an RDF graph (and more precisely an RDF graph with variables). The maps that are used to compute answers to a graph pattern query in an RDF knowledge base are exploited by (Perez, Arenas, & Gutierrez, 2006) to define answers to the more complex, more expressive SPARQL queries (using, for example, disjunctions or functional constraints).

This chapter provides an introduction to RDF as well as surveys the languages that can be used for querying RDF graphs. In particular, it introduces the RDF language: its syntax and its semantics. Then it reviews some of the well known graph query languages used for querying structured or semi-structured data bases, presents some of RDF query languages, and details the SPARQL query language. It then presents some extensions of the SPARQL query language. Finally, it provides a comparison between these query languages.

DOI: 10.4018/978-1-61350-053-8.ch015
The RDF Language

The Resource description Framework (RDF) is a W3C standard language dedicated to the annotation of resources within the SemanticWeb (Manola & Miller, 2004). The atomic constructs of RDF are statements, which are triples (subject, predicate, object) consisting of the resource (the subject) being described, a property (the predicate), and a property value (the object).

For example, the assertion of the following RDF triples \{<book1 rdf:type publication>, <book1 title “Ontology Matching”>, <book1 author “Jérôme Euzenat”>, <book1 publisher “Springer”>\} means that “Jérôme Euzenat” is an author of a book titled “Ontology Matching” whose publisher is “Springer”.

A collection of RDF statements (RDF triples) can be intuitively understood as a directed labeled graph: resources are nodes and statements are arcs (from the subject node to the object node) connecting the nodes. The language is provided with a model-theoretic semantics [Hayes, 2004], that defines the notion of consequence (or entailment) between two RDF graphs, i.e., when an RDF graph is entailed by another one. Answers to an RDF query (the knowledge base and the query are RDF graphs) can be computed using a particular map (a mapping from terms of the query to terms of the knowledge base preserving constants), a graph homomorphism [Gutierrez, Hurtado & Mendelzon, 2004; Baget, 2005].

RDF Syntax

RDF can be expressed in a variety of formats including RDF/XML [Beckett, 2004], Turtle (Beckett, 2006), etc. We use here its abstract syntax (triple format), which is sufficient for our purpose. To define the syntax of RDF, we need to introduce the terminology over which RDF graphs are constructed.

RDF TERMINOLOGY

The RDF terminology \( T \) is the union of three pairwise disjoint infinite sets of terms [Hayes, 2004]: the set \( U \) of URIs, the set \( L \) of literals (itself partitioned into two sets, the set \( Lp \) of plain literals and the set \( Lt \) of typed literals), and the set \( B \) of variables. The set \( V = U \cup L \) of names is called the vocabulary. From now on, we use different notations for the elements of these sets: a variable will be prefixed by ? (like ?b1), a literal will be between quotation marks (like “27”), and the rest will be urirefs (like foaf:Person — foaf: is a namespace prefix used for representing personal information—ex:friend or simply friend).

The RDF terminology \( T \) is the union of three pairwise disjoint infinite sets of terms [Hayes, 2004]: the set \( U \) of URIs, the set \( L \) of literals (itself partitioned into two sets, the set \( Lp \) of plain literals and the set \( Lt \) of typed literals), and the set \( B \) of variables. The set \( V = U \cup L \) of names is called the vocabulary. From now on, we use different notations for the elements of these sets: a variable will be prefixed by ? (like ?b1), a literal will be between quotation marks (like “27”), and the rest will be urirefs (like foaf:Person — foaf: is a namespace prefix used for representing personal information—ex:friend or simply friend).

RDF Graphs as Triples

RDF graphs are usually constructed over the set of urirefs, blanks, and literals [Carroll & Klyne, 2004]. “Blanks” are vocabulary specific to RDF and SPARQL currently considers them as constant values without giving them their existential semantics [Hayes, 2004]. For the sake of simplicity, we do not use blanks in the definition of RDF syntax.

**Definition (RDF graph).** An RDF triple is an element of \( V \times V \times V \). An RDF graph is a finite set of RDF triples.