



RDF 1.1: On Semantics of RDF Datasets

W3C Working Group Note 25 February 2014

This version:

<http://www.w3.org/TR/2014/NOTE-rdf11-datasets-20140225/>

Latest published version:

<http://www.w3.org/TR/rdf11-datasets/>

Previous version:

<http://www.w3.org/TR/2013/WD-rdf11-datasets-20131217/>

Editor:

[Antoine Zimmermann](#), [École Nationale Supérieure des Mines de Saint-Étienne](#)

Please check the [errata](#) for any errors or issues reported since publication.

Copyright © 2014 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

RDF defines the concept of RDF datasets, a structure composed of a distinguished RDF graph and zero or more named graphs, being pairs comprising an IRI or blank node and an RDF graph. While RDF graphs have a formal model-theoretic semantics that determines what arrangements of the world make an RDF graph true, no agreed formal semantics exists for RDF datasets. This document presents some issues to be addressed when defining a formal semantics for datasets, as they have been discussed in the RDF 1.1 Working Group, and specify several semantics in terms of model theory, each corresponding to a certain design choice for RDF datasets.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

The document is part of the RDF 1.1 document suite. The RDF Working Group did not standardize the semantics of RDF datasets. This note discusses issues in specifying such semantics.

This document was published by the [RDF Working Group](#) as a Working Group Note. If you wish to make comments regarding this document, please send them to public-rdf-comments@w3.org ([subscribe](#), [archives](#)). All comments are welcome.

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

1. [Introduction](#)
2. [Existing Work](#)
 - 2.1 [The RDF semantics](#)
 - 2.2 [The Named Graphs paper](#)
 - 2.3 [The SPARQL specification](#)
3. [Formal definitions](#)
 - 3.1 [Named graphs have no meaning](#)
 - 3.1.1 [Formalization](#)
 - 3.1.2 [Examples of entailment and non-entailments](#)
 - 3.1.3 [Properties of this dataset semantics](#)
 - 3.2 [Default graph as union or as merge](#)
 - 3.2.1 [Formalization: first version](#)
 - 3.2.2 [Formalization: second version](#)
 - 3.2.3 [Examples](#)
 - 3.2.4 [Properties of this dataset semantics](#)
 - 3.3 [The graph name denotes the named graph or the graph](#)
 - 3.3.1 [Formalization](#)
 - 3.3.2 [Examples](#)
 - 3.3.3 [Properties of this dataset semantics](#)
 - 3.4 [Each named graph defines its own context](#)
 - 3.4.1 [Formalization](#)
 - 3.4.2 [Examples](#)
 - 3.4.3 [Properties of this dataset semantics](#)
 - 3.4.4 [Variants of this dataset semantics](#)
 - 3.5 [Named graph are in a particular relationship with what the graph name dereferences to](#)
 - 3.5.1 [Formalization](#)
 - 3.5.2 [Examples](#)
 - 3.5.3 [Properties of this dataset semantics](#)
 - 3.6 [Quad semantics](#)
 - 3.6.1 [Formalization](#)
 - 3.6.2 [Properties of this dataset semantics](#)
 - 3.7 [Quoted graphs](#)
 - 3.8 [Relationship with SPARQL entailment regime](#)
4. [Declaring the intended semantics](#)
 - A. [Acknowledgements](#)
 - B. [Changes since the first public working draft of 17 December 2013](#)
 - C. [References](#)
 - C.1 [Informative references](#)

1. Introduction

The [Resource Description Framework \(RDF\)](#) version 1.1 defines the concept of RDF datasets, a notion introduced first by the SPARQL specification [[RDF-SPARQL-QUERY](#)]. An RDF dataset is defined as a collection of [RDF graphs](#) where all but one are [named graphs](#) associated with an [IRI](#) or [blank node](#) (the [graph name](#)), and the unnamed default graph [[RDF11-CONCEPTS](#)]. Given that RDF is a data model equipped with a formal semantics [[RDF11-MT](#)], it is natural to try and define what the semantics of datasets should be.

The RDF Working Group was [chartered](#) to provide such semantics in its

recommendation:

Required features

- Standardize a model and semantics for multiple graphs and graphs stores [...]

However, discussions within the Working Group revealed that very different assumptions currently exist among practitioners, who are using RDF datasets with their own intuition of the meaning of datasets. Defining the semantics of RDF datasets requires an understanding of the two following issues:

- what the graph names (IRI or blank node) denote, or what are the constraints on what the names can possibly denote;
- how the triples in the named graph influence the meaning of the dataset.

Possible choices for the denotation of graph names are:

- it denotes the RDF graph in the (name,graph) pair;
- it denotes the pair itself;
- it denotes a supergraph of the graph inside the pair;
- it denotes a container for the RDF graph, that is, a mutable element;
- it denotes the information resource that can be obtained by dereferencing the graph name, when it is an IRI and if such resource exists;
- it denotes an arbitrary resource that is constrained to be in a special relationship (for instance, `ex:hasGraph` with the graph inside the pair);
- it denotes the deductive closure of the graph inside the pair;
- it denotes an arbitrary resource that is in a special relation with the deductive closure, or with a superset of the graph;
- it denotes an unconstrained resource;
- etc.

Even with an intuitive understanding of what the truth of an RDF dataset should be, the precise model-theoretic formalization can be subject to many variations.

Possible choices for the meaning of the triples in the named graphs include:

- all the triples in the named graphs and default graphs contribute to the truth of the dataset in the same way triples contribute to the truth of a single graph;
- the triples of the named graphs are considered part of the knowledge of the default graph;
- different named graphs indicate different “contexts”, or different “worlds”, and the triples inside a named graph are assumed to be true in the associated context only; in this case, the default graph can be interpreted as yet another context, or be considered as a “global context” which must hold in all contexts, or again as metadata about the contexts;
- the named graphs are considered as “hypothetical graphs” which bear the same consequences as their RDF graphs, but they do not participate in the truth of the dataset; this is similar to the “context” option above but it allows a graph to contain contradictions without making the dataset contradictory;
- the triples are merely quoted without any indication of what they mean; they do not participate in the truth of a dataset.

Depending on the assumptions taken with respect to these two issues, the formalization of the semantics of RDF datasets can vary very much.

In this Working Group Note, we examine the propositions that were given by Working Group members in the course of a one-year-and-a-half debate.

2. Existing Work

We first take a look at existing specifications that could shed a light on how the semantics of datasets should be defined. There are three important documents that closely relate to the issue:

- the RDF semantics, as standardized in 2004 [RDF-MT] and its revision in 2014 [RDF11-MT];
- the article *Named Graphs* by Carroll et al. [CARROLL-05], which first introduced the term “named graph” and contains a section on formal semantics;
- the SPARQL specification [SPARQL11-QUERY], which defines RDF datasets and how to query them.

2.1 The RDF semantics

As described in RDF 1.1 Semantics, a set of RDF graphs can be interpreted as either the union of the graphs or as their merge ([RDF11-MT], Technical note, Section 5.2).

So, a first intuition could be that an RDF dataset, being presented as a collection of graph, should mean exactly what the set of its named graphs and default graph means. However, this completely leaves out the potential meaning of graph names, which could be valuable indicators for the truth of a dataset.

Formally, the semantics of RDF defines a notion of interpretation for a set of triples (i.e., an RDF graph), which then can extend to a set of RDF graphs. A dataset is neither a set of triples nor a set of RDF graphs. It is a set of *pairs* (name,graph) together with a distinguished RDF graph and the RDF semantics does not itself specify a meaning for these pairs.

Conceptually, it is problematic since one of the reasons for separating triples into distinct (named) graphs is to avoid propagating the knowledge of one graph to the entire triple base. Sometimes, contradicting graphs need to coexist in a store. Sometimes named graphs are not endorsed by the system as a whole, they are merely quoted.

2.2 The Named Graphs paper

In Carroll et al. [CARROLL-05], a named graph is defined as a pair comprising an IRI and an RDF graph. The notion of RDF interpretation is extended to named graphs by saying that the graph IRI in the pair must denote the pair itself. This non-ambiguously answers the question of what the graph IRI denotes. This can then be used to define proper dataset semantics, as shown in Section 3.3. Note that it is deliberate that the graph IRI is forced to denote the pair rather than the RDF graph. This is done in order to differentiate two occurrences of the same RDF graph that could have been published at different times, or authored by different people. A simple reference to the RDF graph would simply identify a mathematical set, which is the same wherever it occurs.

2.3 The SPARQL specification

RDF 1.1 borrows the notion of RDF dataset from the SPARQL specification [SPARQL11-QUERY], with the notable different that RDF 1.1 allows graph names to be blank nodes. So, in order to understand the semantics of dataset, it is worthwhile looking at how SPARQL uses datasets. SPARQL defines what answers to queries posed against a dataset are, but it never defines the notions that are key to a model theoretic formal semantics: it neither presents interpretations nor entailment. Still, it is worth noticing that a ASK query that only contains a basic graph pattern without variables yields the same result as asking whether the RDF graph in the query is entailed by the default graph. Based on this observation, one may extrapolate that a ASK query containing no variables and only **GRAPH** graph patterns would yield the same result as dataset entailment.

This can be used as a guide for formalizing the semantics of datasets, as can be seen in Section 3.7.

3. Formal definitions

This section presents the different options proposed, together with their formal definitions. We include each time a discussion of the merits of the choice, and some properties.

Each subsection here describes the option informally, before presenting the formal definitions. As far as the formal part is concerned, one has to be familiar with the definitions given in RDF Semantics. We rely a lot on the notion of interpretation and entailment, which are key in model theory.

All proposed options share some commonalities:

- they behave identically on datasets that do not contain named graphs; precisely, entailment between datasets having no named graph is carried out in the same way as entailment between RDF graphs;
- they define notions of interpretation and entailment in function of the corresponding notions in RDF Semantics.

The first item above reflects the indication given in [RDF11-MT] (Section "RDF Datasets") with respect to dataset semantics: “a dataset **SHOULD** be understood to have at least the same content as its default graph”.

The dependency on RDF semantics is such that most of the dataset semantics below reuse RDF semantics as a black box. More precisely, it is not necessary to be specific about how truth of RDF graphs is defined as long as there is a notion of interpretation that determines the truth of a set of triples. In fact, RDF Semantics does not define a single formal semantics, but multiple ones, depending on what standard vocabularies are endorsed by an application (such as the RDF, RDFS, XSD vocabularies). Consequently, we parameterize most of the definitions below with an unspecified entailment regime *E*. RDF 1.1 defines the following entailment regimes: simple entailment, D-entailment, RDF-entailment, RDFS-entailment. Additionally, OWL defines two other entailment regimes, based on the OWL 2 direct semantics [OWL2-DIRECT-SEMANTICS] and the OWL 2 RDF-based semantics [OWL2-RDF-BASED-SEMANTICS].

For an entailment regime *E*, we will say *E*-interpretation, *E*-entailment, *E*-equivalence, *E*-consistency to describe the notions of interpretations, entailment, equivalence and consistency associated with the regime *E*. Similarly, we will use the terms dataset-interpretation, dataset-entailment, dataset-equivalence, dataset-consistency for the corresponding notions in dataset semantics.

This document provides examples in TriG [TRIG] and assumes that the following prefixes are defined:

Namespace prefixes and IRIs used in this document

Namespace prefix	Namespace IRI
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#
ex	http://example.org/voc#

3.1 Named graphs have no meaning

The simplest semantics defines an interpretation of a dataset as an RDF interpretation

of the default graph. The dataset is true, according to the interpretation, if and only if the default graph is true. In this case, any datasets that have equivalent default graphs are dataset-equivalent.

This means that the named graphs in a dataset are irrelevant to determining the truth of a dataset. Therefore, arbitrary modifications of the named graphs in a graph store always yield a logically equivalent dataset, according to this semantics.

3.1.1 Formalization

Considering an entailment regime E , a dataset-interpretation with respect to E is an E -interpretation. Given an interpretation I and a dataset D having default graph G and named graphs NG , $I(D)$ is true if and only if $I(G)$ is true.

3.1.2 Examples of entailment and non-entailments

Consider the following dataset:

EXAMPLE 1

```
{ ex:s ex:p ex:o . }
ex:g1 { ex:a ex:b ex:c }
```

does not dataset-entail:

EXAMPLE 2

```
{ ex:s ex:p ex:o .
  ex:a ex:b ex:c .}
```

but dataset-entails:

EXAMPLE 3

```
{ } # empty default graph
ex:g2 { ex:x ex:y ex:z }
```

Since graph names are not particularly constrained, one can use them in triples, for instance:

EXAMPLE 4

```
{ ex:g1 ex:author ex:Bob .
  ex:g1 ex:created "2013-09-17"^^xsd:date .}
ex:g1 { ex:a ex:b ex:c }
```

but it would dataset-entail:

EXAMPLE 5

```
{ ex:g1 ex:author ex:Bob .
  ex:g1 ex:created "2013-09-17"^^xsd:date .}
ex:g1 { ex:x ex:y ex:z }
```

3.1.3 Properties of this dataset semantics

Assuming this semantics is convenient since it merely ignores named graphs in a dataset for any reasoning task. As a result, datasets can be simply treated as regular RDF graphs by extracting the default graph. Named graphs can still be used to preserve useful information, but it bears no more meaning than a commentary in a program source code.

The obvious disadvantage is that, since named graphs are completely disregarded in terms of meaning, there is no guarantee that any information intended to be conveyed by the named graphs is preserved by inference.

3.2 Default graph as union or as merge

It is sometimes assumed that named graphs are simply a convenient way of sorting the triples but all the triples participate in a united knowledge base that takes the place of the default graph. More precisely, a dataset is considered to be true if all the triples in all the graphs, named or default, are true together. This description allows two formalizations of dataset semantics, depending on how blank nodes spanning several named graphs are treated. Indeed, if one blank node appears in several named graphs, it may be intentional, to indicate the existence of only one thing across the graphs, in which case union is appropriate. If the sharing of blank nodes is incidental, merge is also an applicable solution.

3.2.1 Formalization: first version

We define a dataset-interpretation with respect to an entailment regime E as an E -interpretation. Given a dataset-interpretation I and a dataset D having default graph G and named graphs NG , $I(D)$ is true if and only if $I(G)$ is true and for all ng in NG , $I(ng)$ is true.

This is equivalent to $I(D)$ is true if $I(H)$ is true where H is the [merge](#) of all the RDF graphs, named or default, appearing in D .

3.2.2 Formalization: second version

We define a dataset-interpretation with respect to an entailment regime E as an E -interpretation. Given a dataset-interpretation I and a dataset D having default graph G and named graphs NG , $I(D)$ is true if and only if $I(H)$ is true where H is the union of all the RDF graphs, named or default, appearing in D .

An alternative presentation of this variant is the following: define $I+A$ to be an extended interpretation which is like I except that it uses A to give the interpretation of blank nodes; define $blank(D)$ to be the set of blank nodes in D . Then $I(D)$ is true if and only if $[I+A](D)$ is true for some mapping A from $blank(D)$ to the set of resources in I .

3.2.3 Examples

Consider the following dataset:

EXAMPLE 6

```
{ ex:s ex:p ex:o . } # default graph
ex:g1 { ex:a ex:b ex:c }
```

dataset-entails:

EXAMPLE 7

```
{ ex:s ex:p ex:o . }
```

```
ex:a ex:b ex:c .}
```

If the entailment regime E is RDFS with the recognized datatype `xsd:integer`, then the following RDF dataset is RDFS-dataset-inconsistent:

EXAMPLE 8

```
{ } # empty default graph
ex:g1 { ex:age rdfs:range xsd:integer . }
ex:g2 { ex:bob ex:age "twenty" .}
```

3.2.4 Properties of this dataset semantics

This semantics allows one to partition the triples of an RDF graph into multiple named graphs for easier data management, yet retaining the meaning of the overall RDF graph. Note that this choice of semantics does not impact the way graph names are interpreted: it is possible to further constrain the graph names to denote the RDF graph associated with it, or other possible constraints. The possible interpretations of graph names, and their consequences, are presented in the next sections.

This semantics is implicitly assumed by existing graph store implementations. The OWLIM RDF database management system implements reasoning techniques over RDF datasets that materialize inferred statements into the database [[citation needed]]. This is done by taking the union of the graphs in the named graphs, applying standard entailment regimes over this RDF graph and putting the inferred triples into the default graph.

This dataset semantics makes all triples in the named graphs contribute to a global knowledge, thus making the whole dataset inconsistent whenever two graphs are mutually contradictory. In situations where named graphs are used to store RDF graphs obtained from various sources on the open Web, inconsistencies or contradictions can easily occur. Notably, Web crawlers of search engines harvest all RDF documents, and it is known as a fact that the Web contains documents serializing inconsistent RDF graphs as well as documents that are mutually contradicting yet consistent on their own. In this case, this semantics can be seen as problematic.

3.3 The graph name denotes the named graph or the graph

It is common to use the graph name as a way to identify the RDF graph inside the named graphs, or rather, to identify a particular occurrence of the graph. This allows one to describe the graph or the graph source in triples. For instance, one may want to say who the creator of a particular occurrence of a graph is. Assuming this semantics for graph names amounts to say that each named graph pair is an assertion that sets the [referent](#) of the graph name to be the associated graph or named graph pair.

Intuitively, this semantics can be seen as quoting the RDF graphs inside the named graphs. In this sense, `ex:alice {ex:bob ex:is ex:smart}` has to be understood as “Alice said: “Bob is smart”” which does not entail “Alice said: “Bob is intelligent”” because Alice did not use the word “intelligent”, even though “smart” and “intelligent” can be understood as equivalent. Note, however, that this analogy is only valid insofar as it can provide an intuition of this type of semantics, but the formalization does not actually refer to speech and the act of asserting.

3.3.1 Formalization

In order to be consistent with RDF model theory, blank nodes used as graph names

are treated like existential variables. Consequently, their semantics is formalized according to the same notation presented in [RDF11-MT]:

Suppose I is an interpretation and A is a mapping from a set of blank nodes to the universe IR of I . Define the mapping $[I+A]$ to be I on names, and A on blank nodes on the set: $[I+A](x)=I(x)$ when x is a name and $[I+A](x)=A(x)$ when x is a blank node; and extend this mapping to triples and RDF graphs using the rules given above for ground graphs.

A dataset-interpretation I with respect to an entailment regime E is an E -interpretation extended to named graphs and datasets as follows:

- if (n,g) is a named graph where the graph name is an IRI, then $I(n,g)$ is true if and only if $I(n) = (n,g)$.
- if D is a dataset comprising default graph DG and named graphs NG , then $I(D)$ is true if and only if there exists a mapping from blank nodes to the universe IR of I such that $[I+A](DG)$ is true and for all named graph (n,g) in NG , $[I+A](n) = (n,g)$.

3.3.2 Examples

Consider the following dataset:

EXAMPLE 9

```
{ } # empty default graph
ex:g1 { ex:a ex:b ex:c }
ex:g2 { ex:x ex:y ex:z }
```

dataset-entails:

EXAMPLE 10

```
{ }
_:b { ex:a ex:b ex:c }
ex:g2 { ex:x ex:y ex:z }
```

but does not dataset-entail:

EXAMPLE 11

```
{ }
ex:g1 { [] ex:b ex:c }
ex:g2 { ex:x ex:y ex:z }
```

nor:

EXAMPLE 12

```
{ }
ex:g1 { }
```

If the entailment regime E is RDFS with the recognized datatype `xsd:integer`, then the following RDF dataset is RDFS-dataset-inconsistent:

EXAMPLE 13

```
{ ex:age rdfs:range xsd:integer .
  ex:me ex:age ex:g1 . } # default graph
ex:g1 { ex:s ex:p ex:o }
```

The graph name can be used in triples to attached metadata (here `ex:hasNextVersion` is a custom term that does not enforce a formal constraint, so it is up to the implementation to decide how to treat it):

EXAMPLE 14

```
{ ex:g1 ex:published "2013-08-26"^^xsd:date .
  ex:g1 ex:hasNextVersion ex:g2 .}
ex:g1 { ex:s1 ex:p1 ex:o1 .
        ex:s2 ex:p2 ex:o2 }
ex:g2 { ex:s1 ex:p1 ex:o1 }
```

3.3.3 Properties of this dataset semantics

There are important implications with this semantics. In this case, a named graph pair can only entail itself or a graph that is structurally equivalent if the graph name is a blank node. Graph names have to be handled almost like literals. Unlike other IRIs or blank nodes, their denotation is strictly fixed, like literals are. This means that graph IRIs may possibly clash with constraints on datatypes, as in the example above.

A variant of this dataset semantics imposes that the graph name denotes the RDF graph itself, rather than the pair. This means that two occurrences of the same graph in different named graph pairs actually identify the same thing. Thus, the graph names associated with the same RDF graphs are interchangeable in any triple in this case.

3.4 Each named graph defines its own context

Named graphs in RDF datasets are sometimes used to delimit a context in which the triples of the named graphs are true. From the truth of these triples according to the graph semantics, follows the truth of the named graph pair. An example of such situation occurs when one wants to keep track of the evolution of facts with time. Another example is when one wants to allow different viewpoints to be expressed and reasoned with, without creating a conflict or inconsistency. By having inferences done at the named graph level, one can prevent for instance that triples coming from untrusted parties are influencing trusted knowledge. Yet it does not disallow reasoning with and drawing conclusions from untrusted information.

Intuitively, this semantics can be seen as interpreting the RDF graphs inside the named graphs. In this sense, `ex:alice {ex:bob ex:is ex:smart}` has to be understood as “Alice said that Bob is smart” which entails “Alice said that Bob is intelligent” because it is what Bob means, whether he used the term “smart”, “intelligent”, or “bright”. Neither sentence implies that Alice used these actual words.

3.4.1 Formalization

There are several possible formalizations of this leading to similar entailments. One way is to interpret the graph name as denoting a graph, and a named graph pair is true if this graph entails the graph inside the pair. In this case, a dataset-interpretation with respect to an entailment regime E is an E -interpretation such that:

- given a mapping A from blank nodes to the univers IR and a named graph pair $ng = (n, G)$, $[I+A](ng)$ is true if $[I+A](n)$ is an RDF graph and E -entails G ;
- for a dataset $D = (DG, NG)$, $I(D)$ is true if there exists a mapping A from blank nodes to the universe IR such that $[I+A](DG)$ is true and for all named graph ng

- in NG , $[I+A](ng)$ is true;
- $I(D)$ is false otherwise.

3.4.2 Examples

Consider the following dataset:

EXAMPLE 15

```
{ } # empty default graph
ex:g1 { ex:YoutubeEmployee rdfs:subClassOf ex:GoogleEmployee .
        ex:steveChen rdf:type ex:YoutubeEmployee . }
ex:g2 { ex:chadHurley rdf:type ex:YoutubeEmployee }
```

RDFS-dataset-entails:

EXAMPLE 16

```
{ }
ex:g1 { ex:steveChen rdf:type ex:GoogleEmployee }
```

but does not RDFS-dataset-entail:

EXAMPLE 17

```
{ }
ex:g2 { ex:chadHurley rdf:type ex:GoogleEmployee }
```

Graph names used in triples that express metadata do not necessarily generate inconsistency:

EXAMPLE 18

```
{ ex:g1 ex:validAfter "2006"^^xsd:gYear .
  ex:g1 ex:published "2013-08-26"^^xsd:date .
  ex:g2 ex:validAt "2005"^^xsd:gYear . }
ex:g1 { ex:YoutubeEmployee rdfs:subClassOf ex:GoogleEmployee .
        ex:steveChen rdf:type ex:YoutubeEmployee . }
ex:g2 { ex:chadHurley rdf:type ex:YoutubeEmployee }
```

(here, `ex:validAfter` and `ex:validAt` are custom terms that do not enforce a formal constraint, but may be used internally for, e.g., checking the temporal validity of triples in the named graph).

3.4.3 Properties of this dataset semantics

This semantics assumes that the truth of named graphs is preserved when replacing the RDF graphs inside named graphs with equivalent graphs. This means in particular, that one can normalize literals and still preserve the truth of a named graph. This means too that standard RDF inferences that can be drawn from the RDF graphs inside named graphs can be added to the graph associated with the graph name without impacting the truth of the RDF dataset.

While this semantics does not guarantee that reasoning with RDF datasets will preserve the exact triples of an original dataset, it is semantically valid to store both the original and any entailed datasets.

An example implementation of such a context-based semantics is Sindice [DELBRU-ET-AL-2008].

3.4.4 Variants of this dataset semantics

There are several variants of this type of dataset-semantics

- The default graph is interpreted as universal truth, that is, for a named graph (n, G) , $I(n)$ E -entails the default graph.
- The graph name does not denote an RDF graph but a resource associated with an RDF graph.
- Each named graph could be associated with a distinct E -interpretation and impose all interpretations to be true for their corresponding graph, in order for the dataset to be true.

3.5 Named graph are in a particular relationship with what the graph name dereferences to

In accordance with linked data principles, IRIs may be assumed to reference the document that is obtained by dereferencing it. If the document contains an RDF graph it can be assumed that the graph in the named graph is in a special relationship (such as, equals, entails) with this RDF graph.

In such case, the truth of an RDF dataset is dependent on the state of the Web, and the same dataset may entail different statements at different times.

3.5.1 Formalization

Let d be the function that maps an IRI to an RDF graph that can be obtained from dereferencing the IRI. For an IRI u , $d(u)$ is empty when dereferencing returns an error or a document that does not encode an RDF graph.

A dataset-interpretation I with respect to an entailment regime E is an E -interpretation such that:

- for a named graph pair $ng = (n, G)$, $I(ng)$ is true if $d(n)$ equals (respectively, is a subgraph of, is entailed by) G ;
- for a dataset $D = (DG, NG)$, $I(D)$ is true if $I(DG)$ is true and for all named graph ng in NG , $I(ng)$ is true;
- $I(D)$ is false otherwise.

3.5.2 Examples

Entailments in this semantics depend not only on the content of a dataset but also on the content of the Web and the ability of a reasoner to accept this content. Moreover, the entailments vary whether the considered relation is “equals”, or “subgraph of”, or “entailed by”.

For instance, if the reasoner is offline, then the dereferencing function d in the previous definition always return an empty graph. In this case, if the relation is “equals” or “subgraph of”, only empty named graphs can be true; if the relation is “entails by”, then only named graphs containing axiomatic triples are true. In general, if the relationship is “equals”, named graph do not provide extra entailments.

3.5.3 Properties of this dataset semantics

The distinguishing characteristic of this dataset semantics is the fact that a single RDF dataset can lead to different entailments, depending on the state of the Web. This can

be seen as a feature for systems that need to be in line with what is found online, but is a drawback for systems that must retain consistency even when they go offline.

3.6 Quad semantics

This approach consists in considering named graph as sets of quadruples, having the subject, predicate and object of the triples as first three components, and the graph IRI as the fourth element. Each quadruple is interpreted similarly to a triple in RDF, except that the relation that the predicate denotes is not indicating a binary relation but a ternary relation.

This semantics is extending the semantics of RDF rather than simply reusing it.

3.6.1 Formalization

A quad-interpretation is a tuple $(IR, IP, IEXT, IS, IL, LV)$ where IR , IP , IS , IL and LV are defined as in RDF and $IEXT$ is a mapping from IP into the powerset of $IR \times IR$ union $IR \times IR \times IR$.

Since this option modifies the notion of simple-interpretation, which is the basis for all E -interpretations in any entailment regime E , it is not clear how it can be extended to arbitrary entailment regimes. For instance, does the following quad set:

EXAMPLE 19

```
ex:a rdf:type ex:c ex:x .
ex:c rdfs:subClassOf ex:d ex:x .
```

RDFS-dataset-entails:

EXAMPLE 20

```
ex:a rdf:type ex:d ex:x .
```

3.6.2 Properties of this dataset semantics

With this semantics, all inferences that are valid with normal RDF triples are preserved, but it is necessary to extend RDFS in order to accommodate for ternary relations. There are several existing proposals that extend this quad semantics by dealing with a specific “dimension”, such as time, uncertainty, provenance. For instance, temporal RDF [TEMPORAL-RDF] uses the fourth element to denote a time frame and thus allow reasoning to be performed per time frame. Special semantic rules allow one to combine triples in overlapping time frames. Fuzzy RDF [FUZZY-RDF] extends the semantics to deal with uncertainty. stRDF [ST-RDF] extends temporal RDF to deal with spatial information. Annotated RDF [ANNOTATED-RDF] generalizes the previous proposals.

3.7 Quoted graphs

Quoted graphs are a way to associate information to a specific RDF graph without constraining the relationship between a graph name and the graph associated with it in a dataset. An RDF graph is “quoted” by using a literal having a lexical form that is a syntactic expression of the graph. For instance:

EXAMPLE 21

```
{ ex:g ex:quotes "ex:a ex:b []"^^ex:turtle . }
```

```
ex:g { ex:b rdf:type rdfex:Property .
      ex:a ex:b _:x . }
```

This technique allows one to assume a dataset semantics of contexts (as in Section 3.4) and still preserve an initial version of a graph. However, quoting big graphs may be cumbersome and would require a custom datatype to be recognized.

3.8 Relationship with SPARQL entailment regime

There is a strong relationship between SPARQL ASK queries with an entailment regime [[SPARQL11-ENTAILMENT](#)] and inferences in the regime. If an ASK query does not contain variables and its WHERE clause only contains a basic graph pattern, then the query can be seen as an RDF graph. If such a graph query Q returns **true** when issued against an RDF graph G with entailment regime E , then G E -entails Q . If it returns **false**, then G does not E -entail Q .

A dataset semantics can also be compared to what ASK queries return when they do not contain variables but may contain basic graph patterns or graph graph patterns. For instance, consider the dataset:

EXAMPLE 22

```
{ }
ex:g1 { ex:x rdf:type ex:c .
        ex:c rdfs:subClassOf ex:d . }
ex:g2 { ex:y rdf:type ex:c . }
```

Then the query:

EXAMPLE 23

```
ASK WHERE {
  GRAPH ex:g1 { ex:x rdf:type ex:d }
}
```

with RDFS entailment regime would answer **true**, but the query:

EXAMPLE 24

```
ASK WHERE {
  GRAPH ex:g1 { ex:x rdf:type ex:d }
  GRAPH ex:g2 { ex:y rdf:type ex:d }
}
```

would answer **false**.

This can lead to a classification of dataset semantics in terms of whether they are compatible with SPARQL ASK queries or not. It can be noted that a semantics where each named graph defines its own context is “SPARQL-ASK-compatible”, while a semantics where the graph name denotes the graph or named graph is not compatible in this sense.

4. Declaring the intended semantics

The RDF Working Group did not define a formal semantics for a multiple graph data model because none of the semantics presented before could obtained consensus. Choosing one or another of the propositions before would have gone against some

deployed implementations. Therefore, the Working Group discussed the possibility to define several semantics, among which an implementation could choose, and provide the means to declare which semantics is adopted.

This was not retained eventually, because of the lack of experience, so there is no definite option for this. Nonetheless, for completeness, we describe here possible solutions.

Using vocabularies

A dataset can be described in RDF using vocabularies like `void` [VOID] and the SPARQL service description vocabulary [SPARQL11-SERVICE-DESCRIPTION]. `VoiD` is used to describe how a collection of RDF triples is organized in a web site or across web sites, giving information about the size of the datasets, the location of the dump files, the IRI of the query endpoints, and so on. The notion of dataset in `void` is used as a more informal and broader concept than RDF dataset. However, an RDF dataset and the graphs in it can be describe as `void` datasets and the information can be completed with SPARQL service description

EXAMPLE 25

```
@prefix er: <http://www.w3.org/ns/entailment> .
@prefix sd: <http://www.w3.org/ns/sparql-service-description#> .
[] a sd:Dataset;
  sd:defaultEntailmentRegime er:RDF;
  sd:namedGraph [
    sd:name "http://example.com/ng1";
    sd:entailmentRegime er:RDFS
  ] .
```

A vocabulary specifically tailored for describing the intended dataset semantics could be defined in a future specification.

Using other mechanisms

Communication of the intended semantics could be performed in various ways, from having the author tell the consumers directly, to inventing a protocol for this. Use of the HTTP protocol and content negotiation could be a possible way too. Special syntactic markers in the concrete serialization of datasets could convey the intended meaning. All of those are solutions that do not follow current practices.

A. Acknowledgements

This document is the result of extensive discussions that involved many members of the RDF 1.1 Working Group. The editor especially acknowledges valuable contributions from Richard Cyganiak, Sandro Hawks, Pat Hayes, Ivan Herman, Peter F. Patel-Schneider, Guus Schreiber, and David Wood.

B. Changes since the first public working draft of 17 December 2013

- 2014-02-21: Fixed wrong style for some section titles.
- 2014-02-21: Defined prefixes and updated TriG examples.
- 2014-02-19: Updated references to [SPARQL11-QUERY].
- 2014-02-19: Removed Issue 2, adding support for blank nodes in the formalization of the semantics.
- 2014-02-19: Removed Issue 1 and added a sentence to mitigate the issue.
- 2014-02-19: Added an explanatory sentence to the formalization of Section 3.3.

C. References

C.1 Informative references

[ANNOTATED-RDF]

Octavian Udrea, Diego Reforgiato Recupero, V. S. Subrahmanian. *Annotated RDF*. In The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings. Springer, LNCS 4011, pp. 487-501

[CARROLL-05]

Jeremy J. Carroll, Christian Bizer, Patrick J. Hayes and Patrick Stickler. *Named Graphs, Provenance and Trust*. Proc. 14th Int. Conf. on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, pp. 613-622.

[DELBROU-ET-AL-2008]

Renaud Delbru, Axel Polleres, Giovanni Tummarello, Stefan Decker. *Context Dependent Reasoning for Semantic Documents in Sindice*. In Proceedings of the 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS). Karlsruhe, Germany, 2008.

[FUZZY-RDF]

Umberto Straccia. *A Minimal Deductive System for General Fuzzy RDF*. In Web Reasoning and Rule Systems, Third International Conference, RR 2009, Chantilly, VA, USA, October 25-26, 2009, Proceedings. Springer, LNCS 5837, pp. 166-181

[OWL2-DIRECT-SEMANTICS]

Boris Motik; Peter Patel-Schneider; Bernardo Cuenca Grau. *OWL 2 Web Ontology Language Direct Semantics (Second Edition)*. 11 December 2012. W3C Recommendation. URL: <http://www.w3.org/TR/owl2-direct-semantics/>

[OWL2-RDF-BASED-SEMANTICS]

Michael Schneider. *OWL 2 Web Ontology Language RDF-Based Semantics (Second Edition)*. 11 December 2012. W3C Recommendation. URL: <http://www.w3.org/TR/owl2-rdf-based-semantics/>

[RDF-MT]

Patrick Hayes. *RDF Semantics*. 10 February 2004. W3C Recommendation. URL: <http://www.w3.org/TR/rdf-mt/>

[RDF-SPARQL-QUERY]

Eric Prud'hommeaux; Andy Seaborne. *SPARQL Query Language for RDF*. 15 January 2008. W3C Recommendation. URL: <http://www.w3.org/TR/rdf-sparql-query/>

[RDF11-CONCEPTS]

Richard Cyganiak, David Wood, Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation, 25 February 2014. URL: <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>. The latest edition is available at <http://www.w3.org/TR/rdf11-concepts/>

[RDF11-MT]

Patrick J. Hayes, Peter F. Patel-Schneider. *RDF 1.1 Semantics*. W3C Recommendation, 25 February 2014. URL: <http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/>. The latest edition is available at <http://www.w3.org/TR/rdf11-mt/>

[SPARQL11-ENTAILMENT]

Birte Glimm; Chimezie Ogbuji. *SPARQL 1.1 Entailment Regimes*. 21 March 2013. W3C Recommendation. URL: <http://www.w3.org/TR/sparql11-entailment/>

[SPARQL11-QUERY]

Steven Harris; Andy Seaborne. *SPARQL 1.1 Query Language*. 21 March 2013. W3C Recommendation. URL: <http://www.w3.org/TR/sparql11-query/>

[SPARQL11-SERVICE-DESCRIPTION]

Gregory Williams. *SPARQL 1.1 Service Description*. 21 March 2013. W3C Recommendation. URL: <http://www.w3.org/TR/sparql11-service-description/>

[ST-RDF]

Manolis Koubarakis, Kostis Kyzirakos. *Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL*.

In The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I. Springer, LNCS 6088, pp. 425-439.

[TEMPORAL-RDF]

Claudio Gutiérrez, Carlos A. Hurtado, Alejandro A. Vaisman. *Temporal RDF*. In The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings. Springer, LNCS 3532, pp. 93-107

[TRIG]

Gavin Carothers, Andy Seaborne. *TriG: RDF Dataset Language*. W3C Recommendation, 25 February 2014. URL: <http://www.w3.org/TR/2014/REC-trig-20140225/>. The latest edition is available at <http://www.w3.org/TR/trig/>

[VOID]

Keith Alexander; Richard Cyganiak; Michael Hausenblas; Jun Zhao. *Describing Linked Datasets with the Void Vocabulary*. 3 March 2011. W3C Note. URL: <http://www.w3.org/TR/void/>