

## 3. RDFn Graphs

An *RDFn graph* is made up of two sets:

- Set of [RDF triples](#)
- Set of [RDFn edges](#) (See section 3.1.2 below).

### 3.1.1 RDF Triples

See [RDF triples](#), but for literals, use the extended definition in [literals in RDF1.2 FPWD](#).

### 3.1.2 RDFn Edges

An *RDFn edge* is a 4-tuple that is defined as follows:

- If  $s$  is an [IRI](#) or a [blank node](#),  $p$  is an [IRI](#),  $o$  is an [IRI](#), a [blank node](#), or a [literal](#), and  $n$  is an [IRI](#) or a [blank node](#), then  $(s, p, o, n)$  is an RDFn edge.

"RDFn edge" is usually written "edge".

Given an *edge*  $(s, p, o, n)$ ,  $s$  is called the *subject* of the edge,  $p$  is called the *predicate* of the edge,  $o$  is called the *object* of the edge, and  $n$  is called the *name* of the edge.

The name of an *RDFn edge* may be used as the subject or object of an RDF Triple or another *RDFn edge*.

Note: If use of the name of an edge  $e_1$  as the subject and/or object of another edge  $e_2$  is considered a dependency of the edge  $e_1$  on the edge  $e_2$ , then it is not valid in RDFn to have a circular dependency, based on transitivity, where  $e_1$  depends on  $e_1$ .

Note: One could use RDF triples such as  $(n, \text{rdf:type}, C)$ , where  $C$  is a class from a set of classes  $\{\text{rdfn:Occurrence}, \text{rdfn:Quote}, \dots\}$  that are part of a class hierarchy with  $\text{rdfn:EdgeName}$  as the root class, to indicate that an RDFn edge  $(s, p, o, n)$  belongs to some special subcategory (with implications on entailment).

Note: Here we assume the presence of a functional dependency  $n \rightarrow (s, p, o)$  where  $(s, p, o, n)$  represents an RDFn edge (see 3.1.2). Such a functional dependency is not essential and can be removed if support for naming of edge-sets is deemed critical.

## 3.2 IRIs

See [IRIs](#).

## 3.3 Literals

See [Literals \(in RDF1.2 FPWD\)](#).

### 3.4 Blank Nodes

See [Blank Nodes](#).

### 3.5 Replacing Blank Nodes with IRIs

See [Replacing Blank Nodes with IRIs](#).

### 3.6 Graph Comparison

See [Graph Comparison](#).

## 4. RDF Datasets

See [RDF Datasets](#).

## Abstract Syntax (a concise description similar to [this](#) from DRAFT 2023.12.21)

```
graph ::= (triple | edge)*
triple ::= subject predicate object
edge ::= subject predicate object name
subject ::= iri | BlankNode
predicate ::= iri
object ::= iri | BlankNode | literal
name ::= iri | BlankNode
```

## Sample RDF Graph using Concrete Syntax (like N-Triple)

```
:liz :spouse :richard | :wed-1 . # edge 1 (independent of, and hence does not
create, any triple)
:wed-1 :start 1964 . # triple 1a (uses name of edge 1 as subject)
:wed-1 :end 1974 . # triple 1b (uses name of edge 1 as subject)

:liz :spouse :richard | :wed-2 . # edge 21 (independent of, and hence does
not create, any triple)
:wed-2 :start 1975 . # triple 2a (uses name of edge 2 as subject)
:wed-2 :end 1976 . # triple 2b (uses name of edge 2 as subject)
```