



Schemi RDF



L'estensione semantica di RDF

RDF schema

- RDF permette di creare un grafo per descrivere risorse, ma non fornisce nessuna informazione sulle entità descritte né sul vocabolario utilizzato per descriverle.
- La descrizione del vocabolario è affidata al livello successivo, Schemi RDF (RDFS).
- Attraverso il linguaggio degli schemi RDF è possibile descrivere un semplice vocabolario di tipi entità (classi) e relazioni tra di esse (proprietà)

Perché il focus sulle proprietà

Con RDF si può:

- Definire classi e proprietà
- Creare gerarchie di classi e proprietà
- Fare asserzioni sulle singole risorse è di pertinenza di RDF!

Si definisce *property-centric* perché il focus è sulle proprietà:

- le classi compaiono nella definizione delle proprietà, ma non viceversa
- questo rende possibile aggiungere proprietà senza modificare le classi (estensibilità)

Principali costrutti di RDF schema

Class	C (a resource) is an RDF class
Property	P (a resource) is an RDF property
Type	I (a resource) is an instance of C (a class)
Subclass	C1 (a class) is a subclass of C2 (a class)
Subproperty	P1 (a property) is a sub-property of P2 (a property)
Domain	domain of P (a property) is C (a class)
Range	range of P (a property) is C (a class)

Namespace

- Gli elementi dei linguaggi RDF e RDFS sono contraddistinti dai prefissi rdf: e rdfs: che li identificano come tali.
- Ogni serializzazione (tranne N-Triples) rappresenta i namespace e gli IRI secondo un formato specifico

Some example namespace prefixes and IRIs		
Namespace prefix	Namespace IRI	RDF vocabulary
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	The RDF built-in vocabulary [RDF11-SCHEMA]
rdfs	http://www.w3.org/2000/01/rdf-schema#	The RDF Schema vocabulary [RDF11-SCHEMA]
xsd	http://www.w3.org/2001/XMLSchema#	The RDF-compatible XSD types

Definire classi e sottoclassi in RDF

#vocabolario Animals

@prefix animals: <http://www.funny_animals.org#> .

definire la classe Film

animals:Film rdf:type rdfs:Class .

#definire la classe Animale

animals:Animale rdf:type rdfs:Class .

#definire la classe Quadrupede, sottoclasse di Animale

animals:Quadrupede rdf:type rdfs:Class ;

rdfs:subClass animals:Animale .

Formato RDF/XML

```
<!-- classe Animal -->
```

```
<rdfs:Class rdf:about="http://www.funny_animals.org#Animal"/>
```

```
<!-- classe Quadruped, sottoclasse di Animal -->
```

```
<rdfs:Class rdf:about="http://www.funny_animals.org#Quadruped">
```

```
  <rdfs:subClassOf rdf:resource="http://www.funny_animals.org#Animal"/>
```

```
</rdfs:Class>
```

Definire proprietà

```
#proprietà hasLegs  
animals:hasLegs rdf:type rdf:Property .
```

Definire proprietà (RDF/XML)

```
<rdf:Property rdf:about="http://www.funny_animals.org#hasLegs" />
```

```
<rdf:Property rdf:about="http://www.funny_animals.org#hasFourLegs">
    <rdfs:subPropertyOf rdf:resource="http://www.funny_animals.org#hasLegs"/>
</rdf:Property>
```

- La proprietà definita si chiama hasFourLegs (rdf:about), è una sottoproprietà di hasLegs (subPropertyOf)

Collegare le classi tramite le proprietà

- In RDFS, è possibile definire domain e range delle proprietà

La proprietà abita ha come dominio la classe Animal e come range la classe Habitat

```
animals:abita rdfs:domain animals:Animale ;  
          rdfs:range animals:Habitat .
```

La proprietà hasLegs ha come range la classe Leg

```
animals:hasLegs rdfs:range animals:Leg ;
```

Domain e range delle proprietà (RDFS)

```
<rdf:Property rdf:about="http://www.funny_animals.org#hasLegs">  
  <rdfs:domain rdf:resource="http://www.funny_animals.org#Animal"/>  
  <rdfs:range rdf:resource="http://www.funny_animals.org#Leg"/>  
</rdf:Property>
```

Individui e classi (RDF)

#Marty è una zebra

```
<http://www.moviedata.com/characters#Marty> rdf:type animals:Zebra .
```

a corrisponde a rdf:type in RDF

```
<http://www.moviedata.com/characters#Marty> a animals:Zebra .
```

#Melman è una giraffa e conosce Marty

```
<http://www.moviedata.com/characters#Melman> rdf:type animals:Giraffa ;  
foaf:knows animals:Marty .
```

Individui e classi (RDF/XML)

```
<rdf:Description rdf:about="http://www.funny_animals.org#Marty">  
    <rdf:type rdf:resource="http://www.funny_animals.org#Quadruped"/>  
</rdf:Description>
```

La risorsa Marty appartiene alla (ha come tipo la) classe Quadruped

Namespace in RDF/XML

```
<rdf:RDF  
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
    xmlns:animals="http://www.funny_animals.org#"  
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  
>  
  
...  
  
<rdfs:domain rdf:resource="..." />  
  
...  
  
<rdf:Description rdf:about="http://www.funny_animals.org#Marty">  
    <animals:countryOfOrigin rdf:resource="http://..."/>  
  
    ...  
  
</rdf:Description>  
  
...
```

Vocabolari RDF

- FOAF (Friend Of A Friend): vocabolario per descrivere reti sociali
- Dublin Core: un vocabolario di elementi (titolo, autore, ecc.) per descrivere risorse in termini editoriali.
- Schema.org: vocabolario nato su iniziativa di un consorzio di industrie per descrivere gli argomenti delle pagine web secondo uno schema semantico
- SKOS: W3C recommendation dal 2009, permette di descrivere e allineare terminologie diverse

Vocabolari e *Knowledge graphs*

- Vocabolario RDF
 - La definizione di un insieme di classi e proprietà costituisce un vocabolario
 - Il vocabolario viene pubblicato a un certo IRI perché possa essere riferito da tutti
- Uno o più vocabolari possono essere usati per descrivere i dati
 - La descrizione di risorse in riferimento a uno o più vocabolari (es. Monna Lisa), con proprietà che collegano tra di loro le risorse è denominata Knowledge Graph
- Classi e proprietà → Vocabolario
- Triple → Knowledge graph

Madagascar in RDF

- ✓ Classi: Film, Animale, Luogo, Bistecca, Leone, Zebra, Giraffa, Ippopotamo
- ✓ Proprietà: haPersonaggio, abita
- ✓ Dataset:
[<http://www.moviedata.com/...>](http://www.moviedata.com/...)
per IRI dei personaggi e luoghi del film Madagascar
- ✓ FOAF per relazioni tra personaggi
- ✓ Dublin Core per dati del film



Validatore RDF

- <https://www.w3.org/RDF/Validator/>

Triple dell'esempio:

Number	Subject	Predicate	Object
1	http://www.funny_animals.org#hasFourLegs	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
2	http://www.funny_animals.org#hasFourLegs	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.funny_animals.org#hasLegs
3	http://www.funny_animals.org#hasLegs	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
4	http://www.funny_animals.org#hasLegs	http://www.w3.org/2000/01/rdf-schema#domain	http://www.funny_animals.org#Animal
5	http://www.funny_animals.org#hasLegs	http://www.w3.org/2000/01/rdf-schema#range	http://www.funny_animals.org#Leg
6	http://www.funny_animals.org#Animal	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
7	http://www.funny_animals.org#Leg	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
8	http://www.funny_animals.org#Quadruped	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
9	http://www.funny_animals.org#Quadruped	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.funny_animals.org#Animal
10	http://www.funny_animals.org#Marty	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.funny_animals.org#Quadruped

Ragionamento RDF

DFS entailment patterns.		
	If S contains:	then S RDFS entails recognizing D:
<i>rdfs1</i>	any IRI aaa in D	aaa rdf:type rdfs:Datatype .
<i>rdfs2</i>	aaa rdfs:domain xxx . yyy aaa zzz .	yyy rdf:type xxx .
<i>rdfs3</i>	aaa rdfs:range xxx . yyy aaa zzz .	zzz rdf:type xxx .
<i>rdfs4a</i>	xxx aa yyy .	xxx rdf:type rdfs:Resource .
<i>rdfs4b</i>	xxx aaa yyy .	yyy rdf:type rdfs:Resource .
<i>rdfs5</i>	xxx rdfs:subPropertyOf yyy . yyy rdfs:subPropertyOf zzz .	xxx rdfs:subPropertyOf zzz .
<i>rdfs6</i>	xxx rdf:type rdf:Property .	xxx rdfs:subPropertyOf xxx .
<i>rdfs7</i>	aaa rdfs:subPropertyOf bbb . xxx aaa yyy .	xxx bbb yyy .
<i>rdfs8</i>	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf rdfs:Resource .
<i>rdfs9</i>	xxx rdfs:subClassOf yyy . zzz rdf:type xxx .	zzz rdf:type yyy .
<i>rdfs10</i>	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf xxx .
<i>rdfs11</i>	xxx rdfs:subClassOf yyy . yyy rdfs:subClassOf zzz .	xxx rdfs:subClassOf zzz .
<i>rdfs12</i>	xxx rdf:type rdfs:ContainerMembershipProperty .	xxx rdfs:subPropertyOf rdfs:member .
<i>rdfs13</i>	xxx rdf:type rdfs:Datatype .	xxx rdfs:subClassOf rdfs:Literal .

<https://www.w3.org/TR/rdf11-mt/> (RFD Semantics)

Inferenze su domain e range

	<i>S contains</i>	<i>S RDFS-entails</i>
rdfs2	aaa rdfs:domain xxx . yyy aaa zzz .	yyy rdf:type xxx .

Se

animals:abita rdfs:domain animals:Animale ;
 rdfs:range animals:Habitat .

e

md:Melman animals:abita animals:savana .

implica

md:Melman rdf:type animals:Animale .

Se Melman abita nella savana, allora Melman è un animale perché ad abitare nella savana sono gli animali

Inferenze su domain e range

	<i>S contains</i>	<i>S RDFS-entails</i>
rdfs3	aaa rdfs:range xxx . yyy aaa zzz .	zzz rdf:type xxx .

Se

animals:abita rdfs:domain animals:Animale ;
 rdfs:range animals:Habitat .

e

md:Marty animals:abita animals:savana .

implica

animals:savana rdf:type animals:Habitat .

Se Marty abita nella savana, allora la savana è un habitat perché il range di abitare appartiene alla classe Habitat

Transitività della relazione di sotto-proprietà

rdfs5	xxx rdfs:subPropertyOf yyy . yyy rdfs:subPropertyOf zzz .	xxx rdfs:subPropertyOf zzz .
--------------	--	--

Transitività della relazione di sotto-classe

rdfs11	xxx rdfs:subClassOf yyy . yyy rdfs:subClassOf zzz .	xxx rdfs:subClassOf zzz .
---------------	--	---

Se una risorsa (**zzz**) appartiene a una classe (**xxx**), appartiene anche alle sue classi sovraordinate (**yyy**)

rdfs9	xxx rdfs:subClassOf yyy . zzz rdf:type xxx .	zzz rdf:type yyy .
--------------	---	----------------------------------

Cosa manca al dizionario

- Non si può dire che Zebre e Leoni sono classi disgiunte
 - Ma solo che sono tipi specifici di animali
- Non si può dire che la relazione ‘conoscente di’ è transitiva
- Non si può dire che esistono individui
 - Esistono solo risorse, cioè IRI creati esternamente al linguaggio