Einführung in RDF Schema

Markus Stocker

4. Juni 2018

Rekapitulation

- Was ist ein *triple pattern*?
- Welche Resultatformate gibt es in SPARQL?
- Wozu verwendet man `LIMIT` und `OFFSET`?
- Was leistet SPARQL Update?
- Worum geht es bei der Abfragenoptimierung?
- In welchem Zusammenhang haben wir über Schema gesprochen?
- Welche Sprachen haben wir angeschaut?

Übersicht

- Was ist RDF Schema?
- Konzepte und Sprachkonstrukte
- Beispiele

Schema: Schon wieder?

- Tripel als fundamentale Einheit in RDF
- RDF Daten ergeben eine Tripelmenge
- Solche Mengen können sehr gross sein
- Milliarden, sogar Billionen oder mehr
- Solche Mengen sind unübersichtlich
- Worüber sagen die Tripel etwas aus?
- Eine Übersicht zu erhalten ist auf solchen Mengen schwierig
- Schema kommt dabei zur Hilfe

RDF Schema

- RDF Schema ermöglicht das Organisieren von Tripelmengen
- Es ist eine Sprache und wird meist mit RDFS gekürzt
- Version 1.0 seit 2004 eine W3C Empfehlung (*Recommendation*)
- Seit 2014 in der Version 1.1 (ebenfalls W3C Empfehlung)

RDF Schema

- Zentral ist die Gruppierung von Ressourcen
- Wie z.B. `(a,b,...)` sind Planeten; `(x,y,...)` sind Satelliten
- RDFS stellt Konstrukte zur Verfügung
- Die u.A. solche Gruppierungen ermöglichen
- Gruppen werden *Klassen* genannt
- Man kann mit RDFS also aussagen, dass `K` eine Klasse ist
- `K` steht hier für die Klasse der z.B. Planeten, Tiere, Studenten, ...

Ressourcen Gruppieren: Beispiel

```
ex:ghf5 rdfs:label "Moon" .
ex:12bv ex:radius "6371" .
ex:fg54 rdfs:label "Mars" .
ex:12bv ex:satellite ex:ghf5 .
ex:12bv rdfs:label "Earth" .
```

- Folgende Ressourcen: `ex:12bv, ex:fg54, ex:ghf5`
- Diese kann man gruppieren, und zwar
- `(ex:12bv, ex:fg54)` und `(ex:ghf5)`
- Man erhält also zwei Klassen

Klassen

- Klassen sind benannt, mittels URIs
- Klassenbildung führt somit zu neuen Terme
- Dies sind Elemente eines Vokabulars
- Ein Vokabular das beschreibt worüber eine Tripelmenge “spricht”

Klassen Benennen: Beispiel

```
ex:ghf5 rdfs:label "Moon" .
ex:12bv ex:radius "6371" .
ex:fg54 rdfs:label "Mars" .
ex:12bv ex:satellite ex:ghf5 .
ex:12bv rdfs:label "Earth" .
```

- Die Klasse der Planeten: `ex:Planet = (ex:12bv, ex:fg54)`
- Die Klasse der Satelliten: `ex:Satellite = (ex:ghf5)`

Klassen Spezifizieren

- URI alleine genügt nicht um auszusagen, dass ein Name eine Klasse ist
- Sind `ex:Planet`, `ex:fg54` Klassen? Unbestimmt
- Man muss Klassennamen explizit als Klassen definieren
- Dafür stellt RDFS Konstrukte zur Verfügung
- Insbesondere die "Klasse aller Klassen": `rdfs:Class`
- Diese "Metaklasse" ist Teil der RDFS Sprache

Klassen Spezifizieren: Beispiel

```
ex:Planet rdf:type rdfs:Class .
```

Klassen Instanzen

- Hat man Klassen definiert, können Instanzen erzeugt werden
- Instanzen sind Elemente einer Klasse
- Das Prädikat `rdf:type` definiert die Klasse einer Instanz
- So typisiert man Ressourcen
- Sprich, definiert Ressourcen als Elemente einer Menge

Klassen Instanzen: Beispiel

```
ex:Planet rdf:type rdfs:Class .
ex:12bv rdf:type ex:Planet .
```

- `rdfs:Class` ist die Klasse aller Klassen

Klassen Instanzen: Beispiel

```
ex:Planet rdf:type rdfs:Class .
ex:12bv rdf:type ex:Planet .
```

- `rdfs:Class` ist die Klasse aller Klassen
- `ex:Planet` ist eine Instanz der Klasse aller Klassen
- `ex:Planet` ist somit eine Klasse

Klassen Instanzen: Beispiel

```
ex:Planet rdf:type rdfs:Class .
ex:12bv rdf:type ex:Planet .
```

- `rdfs:Class` ist die Klasse aller Klassen
- `ex:Planet` ist eine Instanz der Klasse aller Klassen
- `ex:Planet` ist somit eine Klasse
- `ex:12bv` ist eine Instanz der Klasse `ex:Planet`

Die Metaklasse ist eine Klasse

```
rdfs:Class rdf:type rdfs:Class .
```

- Die Klasse aller Klassen (die Metaklasse) ist selbst eine Klasse!

RDF und RDFS Klassen (Auswahl)

- `rdfs:Resource`, die Klasse aller Ressourcen (Klassen, Instanzen, ...)
- `rdfs:Class`, die Klasse aller Klassen
- `rdfs:Literal`, die Klasse aller Literale (Werte)
- `rdfs:Datatype`, die Klasse aller Datentypen (Instanzen sind Klassen)
- `rdf:Property`, die Klasse aller Prädikate (Relationen)

Unterklassen (*sub classes*)

- Wir nun haben die Klasse `ex:Planet` definiert
- In unserem Sonnensystem werden Planeten in zwei Gruppen unterteilt
- Innere Planeten und äussere Planeten
- Innere Planeten sind näher an der Sonne, kleiner und steinig
- Äussere Planeten sind weiter entfernt, grösser und bestehen aus Gasen
- Innere Planeten: Merkur, Venus, Erde und Mars
- Äussere Planeten: Jupiter, Saturn, Uranus und Neptun
- Natürlich sind das alle Planeten

Unterklassen

```
ex:Planet rdf:type rdfs:Class .
```

Unterklassen

```
ex:Planet rdf:type rdfs:Class .

ex:InnerPlanet rdf:type rdfs:Class .
ex:OuterPlanet rdf:type rdfs:Class .
```

Unterklassen

```
ex:Planet rdf:type rdfs:Class .
ex:InnerPlanet rdf:type rdfs:Class .
ex:OuterPlanet rdf:type rdfs:Class .

ex:12bv rdf:type ex:InnerPlanet .
ex:fg54 rdf:type ex:InnerPlanet .
ex:rs01 rdf:type ex:OuterPlanet .
ex:3op4 rdf:type ex:OuterPlanet .
```

- Gut, aber es ist nun nicht klar, dass z.B. `ex:12bv` ein Planet ist
- Bekannt ist nur, dass `ex:12bv` ein innerer Planet ist

Unterklassen: rdfs:subClassOf

```
ex:Planet rdf:type rdfs:Class .
ex:InnerPlanet rdfs:subClassOf ex:Planet .
ex:OuterPlanet rdfs:subClassOf ex:Planet .

ex:12bv rdf:type ex:InnerPlanet .
ex:12bv rdfs:label "Earth" .
ex:12bv rdf:type ex:Planet .
```

- ex:InnerPlanet ist unterklasse der Klasse ex:Planet
- ex:OuterPlanet ist unterklasse der Klasse ex:Planet
- Beide sind somit Klassen
- Explizit bekannt ist, dass ex:12bv Instanz von ex:InnerPlanet ist
- Implizit ist, dass ex:12bv auch eine Instanz von ex:12bv ist
- Weil alle inneren Planeten auch Planeten sind

Unterklassen: `rdfs:subClassOf`

- `rdfs:subClassOf` ist ein Prädikat
- Es ist somit eine Instanz der Klasse `rdf:Property`
- Man kann damit Klassenhierarchien bilden
- Ein “super class of” Prädikat gibt es in RDFS nicht

Prädikate (*Property*)

- Prädikate sind Relationen zwischen Ressourcen
- In RDFS sind Prädikate ebenfalls Ressourcen
- Allerdings als eigene Klasse organisiert
- Nämlich die Klasse aller Prädikate, `rdf:Property`
- `rdf:Property` ist somit eine Klasse, kein Prädikat
- Es ist eine Instanz der Klasse `rdfs:Class`
- Instanzen der Klasse `rdf:Property` sind aber Prädikate
- Tripel Prädikate werden automatisch also solche Instanzen behandelt

Prädikate: Beispiel

```
ex:radius rdf:type rdf:Property .

ex:12bv ex:radius "6371" .
```

Unterprädikate (*sub properties*)

- RDFS erlaubt die Spezifikation von Unterprädikate
- Dies ermöglicht die Erstellung von Prädikathierarchien
- Man verwendet dazu das Prädikat `rdfs:subPropertyOf`

Unterprädikate: Beispiel

```
ex:radius rdf:type rdf:Property .
ex:physicalProperty rdf:type rdf:Property .

ex:radius rdfs:subPropertyOf ex:physicalProperty .

ex:12bv ex:radius "6371" .
ex:12bv ex:physicalProperty "6371" .
```

Prädikatrestriktionen (*property restrictions*)

- Ermöglichen Aussagen über die Ressourcen die ein Prädikat verbindet
- Insbesondere Aussagen über Klassenzugehörigkeit der Ressourcen
- Klassenzugehörigkeit der Subjekte und Objekte eines Prädikats
- Beispiel: Sind `A` und `B` verheiratet dann sind beide Personen
- Dazu verwendet man `rdfs:domain` und `rdfs:range`
- `rdfs:domain`: Klassenzugehörigkeit des Subjekts
- `rdfs:range`: Klassenzugehörigkeit des Objekts
- `rdfs:domain` und `rdfs:range` sind beides Prädikate

Prädikatrestriktionen: Beispiel

```
ex:satellite rdf:type rdf:Property .
ex:satellite rdfs:domain ex:Planet .
ex:satellite rdfs:range ex:Satellite .

ex:12bv ex:satellite ex:ghf5 .
ex:12bv rdf:type ex:Planet .
ex:ghf5 rdf:type ex:Satellite .
```

Zusammenfassung

- Zentral in RDFS ist das Konzept der Gruppierung
- Gruppierung von Ressourcen als benannte Klassen
- Ermöglicht die Erstellung von abstraktem Vokabular
- Dieses beschreibt Tripelmengen