

Vorlesung

Cloud- und Web- Technologien

(CS3140)





Einführung in das Semantic Web, RDF und SPARQL

Professor Dr. rer. nat. habil. Sven Groppe

<https://www.ifis.uni-luebeck.de/index.php?id=groppe>

Chronologische Übersicht über die Themen

Nr Thema

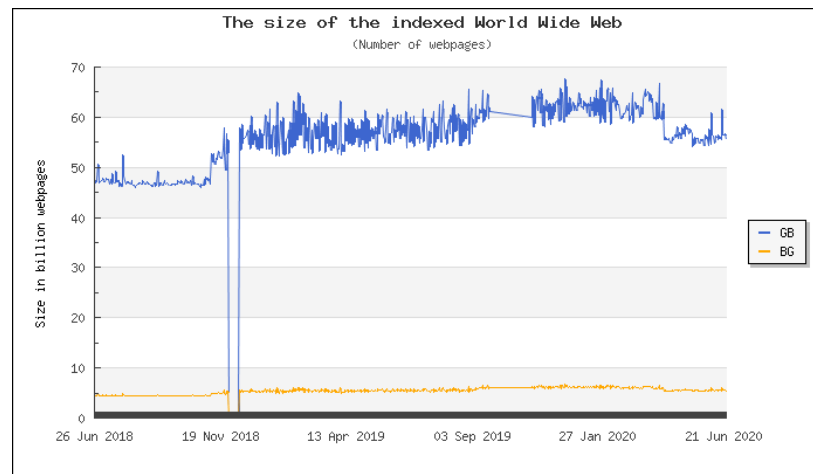
| | | |
|----|--|--|
| 1 | Einleitung | |
| 2 | Einführung in das Semantic Web, RDF und SPARQL |  Datenmodell |
| 3 | Die Semantic Web-Ontologiesprachen RDFS und OWL | |
| 4 | Multiplattform-Entwicklung mit Kotlin |  Multiplattform |
| 5 | Fortgeschrittene Themen mit Kotlin | |
| 6 | Einstieg in Cloud Computing, Hadoop |  Backend |
| 7 | Operatoren der relationalen Algebra in Hadoop | |
| 8 | Datenverarbeitung mit Pig | |
| 9 | Einführung in Spark und Flink | |
| 10 | Stromverarbeitung mit Flink | |
| 11 | Knotenzentrische Algorithmen mit Flink | |
| 12 | HTML und CSS |  Web |
| 13 | Browserprogrammierung mit JS/JQuery und Serverprogrammierung mit PHP Hypertext Preprocessor | |
| 14 | Zusammenfassung und Ausblick | |

Überblick über die Vorlesung

- Einführung in das Semantic Web
- Überblick über die Standards des Semantic Webs
- Das Datenmodell des Semantic Webs
- Die Anfragesprache des Semantic Webs

World Wide Web (WWW): Bestandsaufnahme

- meist verwendeter Dienst im Internet
- Ca. 50 Milliarden Dokumente (geschätzt)



GB: Sorted on Google and Bing **BG:** Sorted on Bing and Google

Source: www.worldwidewebsite.com

Besonderheiten des WWW

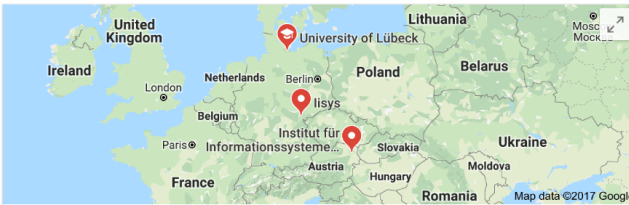
- Hypertext Transfer Protocol (HTTP)
 - Jeder Rechner kann weltweit Informationsquelle sein
 - Hypertext Markup Language (HTML), Extensible Markup Language (XML), Javascript
 - Standards für die Darstellung und Verknüpfung von Informationen / für die Code-Ausführung im Browser
 - Web-Services
 - Jeder Rechner kann weltweit einen Dienst anbieten
- ➔ **Metadaten für die Beschreibung von Semantik:**
stark wachsender Bedarf (und Präsenz)

Herausforderungen für das WWW

Institut für Informationssysteme

All Maps Shopping Images News More Settings Tools

About 455.000 results (0,47 seconds)



Rating

| | |
|---|--|
| University of Lübeck 3.7 ★★★★★ (21) · Universität Lübeck · 0451 31010 | WEBSITE DIRECTIONS |
| lisys 4.0 ★★★★★ (1) · Forschungszentrum Hof · 09281 4096112 | WEBSITE DIRECTIONS |
| Institut für Informationssysteme der TU Wien Wien, Austria · +43 1 5880118402 | WEBSITE DIRECTIONS |

IFIS Uni Lübeck: Startseite
<https://www.ifis.uni-luebeck.de/> [Translate this page](#)
 Willkommen. Das **Institut für Informationssysteme (IFIS)** ist eines der Institute für Informatik der Universität zu Lübeck. Es wurde 1993 als erstes Kerninformatik-Institut an der Universität zu Lübeck unter der Leitung von Prof. Dr. Volker Linnemann etabliert. Seit Oktober 2014 wird das Institut von Prof. Dr. Ralf Möller geleitet.

- **Information Retrieval**
 - Synonyme (zu wenige Resultate)
 - Homonyme (ungewollte Resultate)
 - Keine Kontexteingrenzung
- **Bedeutung von Ontologien**
 - Begriffsfindung/-abgrenzung
 - Kontextanalyse
 - Automatische Anfrageverfeinerung
 - dokumentenübergreifende Beantwortung von Anfragen
 - Stichworte: Textsynthese, Summarization

Herausforderungen für das WWW

Vertiefungen und Anwendungsfächer

Web and Data Science

Die kanonische Vertiefung im Bachelor

Anwendungen in Industrie, Medizin, Verwaltung und Wirtschaft erzeugen zunehmend größere Datenmengen, deren Auswertung ein großes Potential zur Optimierung von Prozessen bzw. Produkten und zur Akquise neuer Kunden bietet, um nur einige Möglichkeiten zu nennen.

Es hat sich im Sprachgebrauch der Begriff Data Science herausgebildet, um die Extraktion von Wissen aus großen Datenmengen zu charakterisieren, so dass hieraus möglicherweise Handlungsempfehlungen oder Hinweise zur Entscheidungsunterstützung systematisch abgeleitet werden können. Zusätzlich wird in Zukunft die Auswertung von vernetzten Strukturen, wie z.B. dem Web, weiter an Bedeutung gewinnen, so dass vielfältige neue Anwendungsfelder erschlossen werden können.

Mit Web und Data Science können Unternehmen effektiver arbeiten, es werden neuartige Behandlungsmethoden entwickelt oder Verwaltungsstrukturen besser organisiert, und es können wirtschaftliche Prozesse besser gelenkt werden. Das spannende Feld benötigt solide Kenntnisse in der Informatik, insbesondere spezielle Kenntnisse aus den Bereichen Algorithmik, Codierung, Datenmanagement (Big Data) und Datenauswertung (Data Mining) sowie aus dem Bereich des Maschinellen Lernens. Informatikkenntnisse werden in der Vertiefung mit relevanten mathematischen Methoden kombiniert, besonders aus den Bereichen Stochastik (Wahrscheinlichkeitstheorie und Statistik) sowie Optimierung.

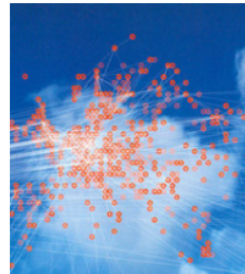
Aufbauend auf diesen Kenntnissen werden die Studierenden in die Lage versetzt, z.B. Klassifikatoren zu bestimmen oder Modelle automatisch aus Daten zu gewinnen, wobei für die zielgerichtete praktische Verwendung von Modellen verschiedene mathematische und informatische Modellierungstechniken systematisch kombiniert und Softwaretechniken zur Auswertung großer Datenmengen (z.B. Cloud Computing) effektiv eingesetzt werden müssen. Studierende der Vertiefung Web and Data Science erwerben Kenntnisse und Fähigkeiten, um moderne Werkzeuge der Datenanalyse anzuwenden und neuartige Werkzeuge für die Zukunft zu entwickeln. Lübeck gehört zu den Vorreiteruniversitäten, die ein solches Fach innerhalb des Informatikstudiengangs systematisch und strukturiert anbieten.

Inhalte

Die Bachelor-Veranstaltungen zu Web and Data Science wie *Codierung und Sicherheit*, *Numerik 1*, *Parallelverarbeitung*,

Berufsbild

Deutschlandweit und sogar weltweit werden in der Wirtschaft und Industrie händierend gut ausgebildete Absolventinnen und



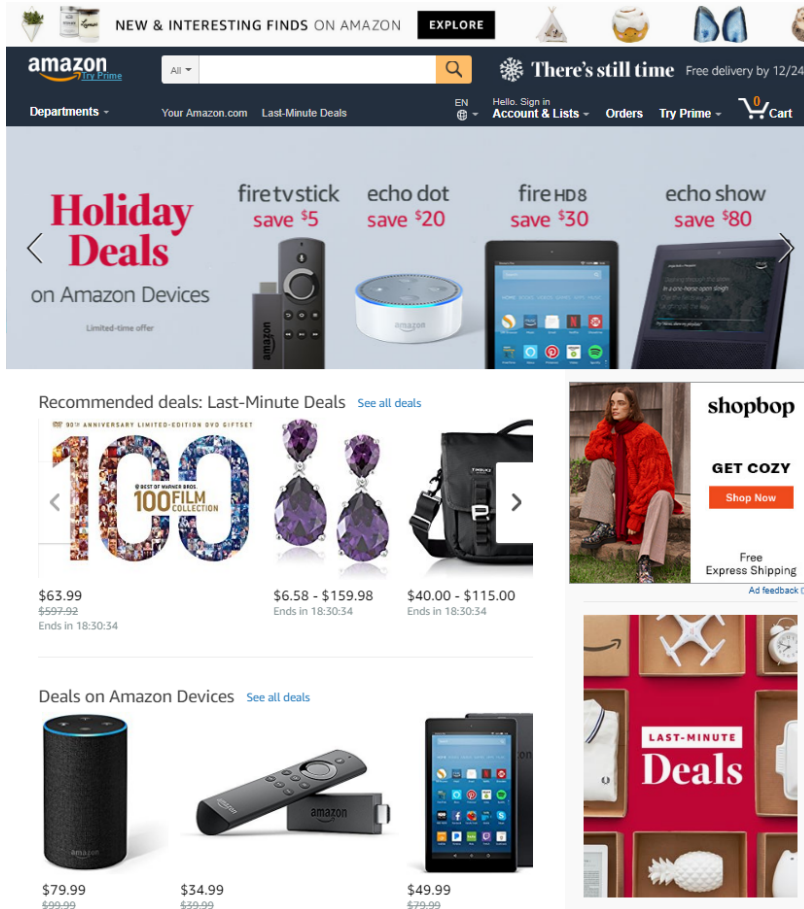
Persönliche Beratung durch:



Prof. Dr. rer. nat.
habil. Ralf Möller
Ansprechpartner Web
and Data Science

- Informationsextraktion
 - nur von Menschen (gut) durchführbar
 - Erkennung von Genre (Texttypen)
 - Informationsaggregation aus Bild und Text
 - Bewertung von Informationsqualität
- Rolle von Semantik
 - Hintergrund- und Weltwissen aus Ontologien
 - (einfache) Schlüsse folgern

Herausforderungen für das WWW



- **Personalisierung**
 - **Anpassung** von Inhalten und Darstellung
 - **Erkennung** von Verhaltensmustern
- **Rolle von Semantik**
 - **Verallgemeinerung und Spezialisierung** von Inhalten (basierend auf Ontologien)
 - **Abstimmen mit Gewohnheiten und Präferenzen** durch Deduktion

Vision des Semantic Webs von Tim Berners-Lee (1999)

“I have a dream for the Web ... and it has two parts.

1. **In the first part**, the Web becomes a much more powerful means for collaboration between people. I have always imagined the information space as something to which everyone has immediate and intuitive access, and not just to browse, but to create. [. . .] Furthermore, the dream of people-to-people communication through shared knowledge must be possible for groups of all sizes, interacting electronically with as much ease as they do now in person.
2. **In the second part** of the dream, collaborations extend to computers. Machines become capable of analyzing all the data on the Web—the content, links, and transactions between people and computers. A '**Semantic Web**', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy, and our daily lives will be handled by machines talking to machines, leaving humans to provide the inspiration and intuition.”

Vision des Semantic Webs nach Franczyk 2005

- Unterstützung des Menschen bei der Verarbeitung von Daten als Aufgabe der Informatik
 - Heutiges Web erfüllt diese Aufgabe nur bedingt
- Berners-Lees Zielvorstellung: Eine Art globale Datenbank durch Vereinigung der Eigenschaften
 - einer (Wissens-) Datenbank
(damit sind Daten durch Maschinen verarbeitbar)
 - mit denen des Webs (jeder kann zu jeder Zeit Daten
(und deren Verknüpfungen untereinander) erzeugen)

Vision des Semantic Webs nach dem W3C (2015)

'In addition to the classic “Web of documents” W3C is helping to **build a technology stack** to support a “Web of data,” the sort of data you find in databases. The **ultimate goal of the Web of data is to enable computers to do more useful work** and to develop systems that can support trusted interactions over the network. The term “**Semantic Web**” refers to **W3C’s vision of the Web of linked data**. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as **RDF, SPARQL, OWL, and SKOS**.'

(Source: <https://www.w3.org/standards/semanticweb/>)

Lösungsansatz des Semantic Webs

- Strukturierung der (Web-) Informationen zur Erleichterung der automatisierten Auswertung
- Miteinbeziehung der Bedeutung von Symbolen zur Präzisierung der maschinellen Verarbeitung
➔ Semantic Web

Vertiefungen und Anwendungsfächer

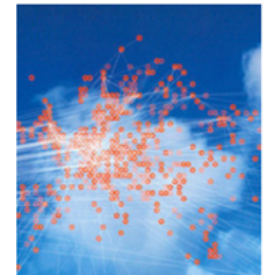
Web and Data Science

Die kanonische Vertiefung im Bachelor

Anwendungen in Industrie, Medizin, Verwaltung und Wirtschaft erzeugen zunehmend größere Datenmengen, deren Auswertung ein großes Potential zur Optimierung von Prozessen bzw. Produkten und zur Akquise neuer Kunden bietet, um nur einige Möglichkeiten zu nennen.

Es hat sich im Sprachgebrauch der Begriff Data Science herausgebildet, um die Extraktion von Wissen aus großen Datenmengen zu charakterisieren, so dass hieraus möglicherweise Handlungsempfehlungen oder Hinweise zur Entscheidungsunterstützung systematisch abgeleitet werden können. Zusätzlich wird in Zukunft die Auswertung von vernetzten Strukturen, wie z.B. dem Web, weiter an Bedeutung gewinnen, so dass vielfältige neue Anwendungsfelder erschlossen werden können.

Mit Web and Data Science können Unternehmen effektiver arbeiten, es werden neuartige Behandlungsmethoden entwickelt oder Verwaltungsstrukturen besser organisiert, und es können wirtschaftliche Prozesse besser gelenkt werden. Das spannende Feld benötigt solide Kenntnisse in der Informatik, insbesondere spezielle Kenntnisse aus den Bereichen Algorithmik, Codierung, Datenmanagement (Big Data) und Datenauswertung (Data Mining) sowie aus dem Bereich des Maschinellen Lernens. Informatikkenntnisse werden in der Vertiefung mit relevanten mathematischen Methoden kombiniert, besonders aus den Bereichen Stochastik (Wahrscheinlichkeitstheorie und Statistik) sowie Optimierung.



Persönliche Beratung durch:



picture name

Instanz:

:RalfMöller

Ontologie:

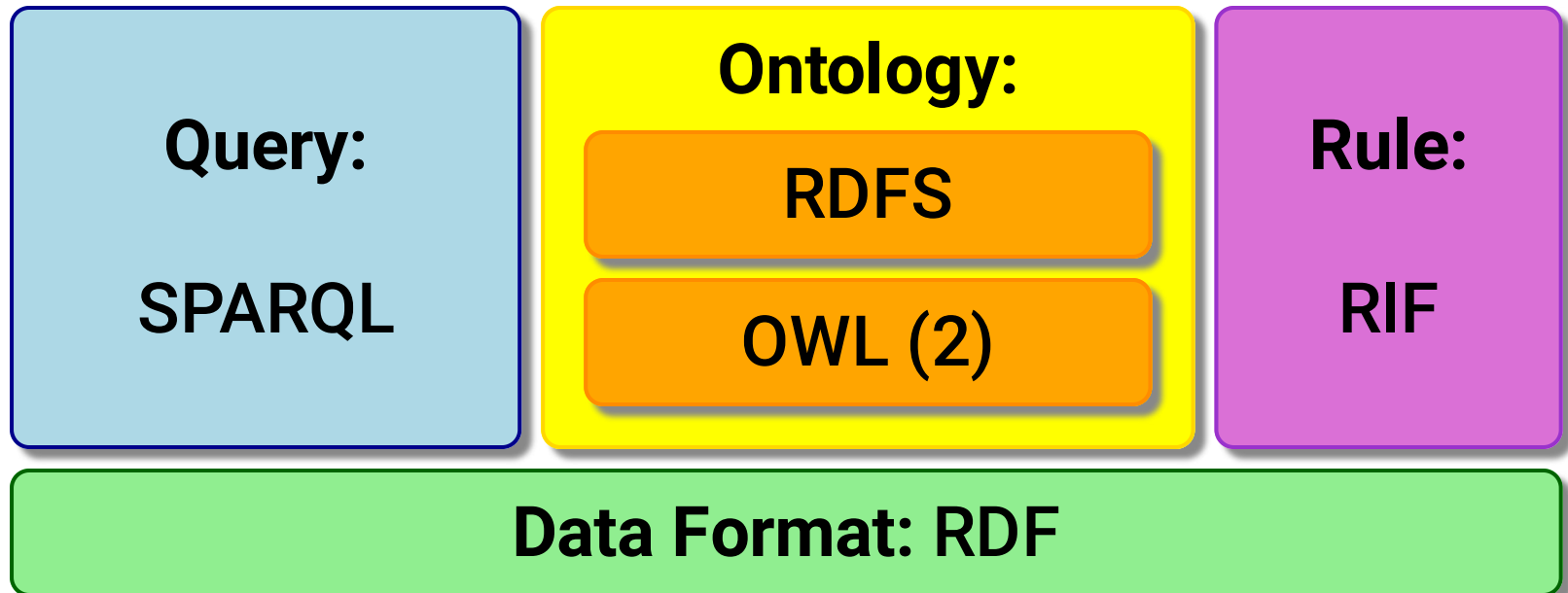
type

:Professor

sub-
ClassOf

:Person

Semantic Web Spezifikationen (Kern)



Offene Welt Annahme (Open world assumption (OWA))

- Geschlossene Welt Annahme
(Closed World Assumption (CWA))
in Datenbanken:

*"Die Datenbank enthält alles und
alles, was nicht in der Datenbank ist,
wird als falsch angenommen!"*

- Offener Kontext wie das Web
➔ CWA Annahme falsch!

OWA - Beispiel

- Datenquelle 1 enthält:

"Es gibt einen Flug um 14:00"

"Es gibt einen Flug um 15:00"

- Meine Anfrage:

"Gibt es einen Flug um 17:00?"

- CWA Antwort: **Nein!**
- OWA Antwort: **unbekannt!**

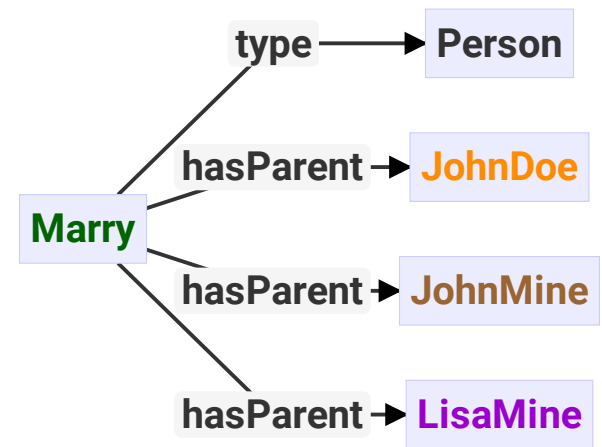
d.h. es könnte eine andere Datenquelle 2 existieren, die enthält, dass es einen Flug um 17:00 gibt!

Datenquelle 2 ist evtl. nur nicht momentan erreichbar...

Keine eindeutigen Namen/Schlüssel (No unique name assumption)

Beispiel:

Ein Kind hat 2 Elternteile (in DL:
 $\text{Person} \sqsubseteq \leq 2 \text{ hasParent. Person}$),
aber wir haben scheinbar
widersprüchliche Fakten
(auf der rechten Seite).



Es gibt **keine** eindeutigen Namen/Schlüssel

➔ **JohnDoe**, **JohnMine** und **LisaMine** sind

nicht unbedingt verschiedene Objekte (hier Personen)

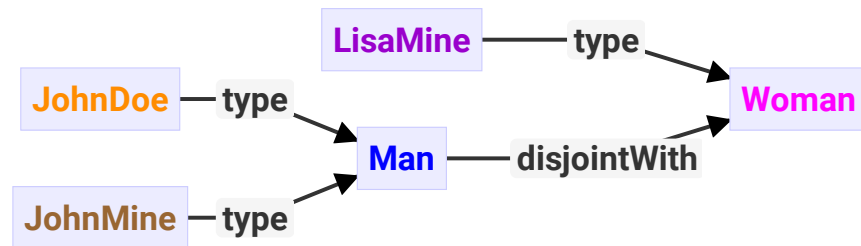
Keine eindeutigen Namen/Schlüssel (No unique name assumption)

4 verschiedene
Möglichkeiten:

1. $\text{JohnDoe} \equiv \text{JohnMine}$
2. $\text{JohnDoe} \equiv \text{LisaMine}$
3. $\text{JohnMine} \equiv \text{LisaMine}$
4. $\text{JohnDoe} \equiv \text{JohnMine} \equiv \text{LisaMine}$

Nur **1.** für Menschen intuitiv!

Hinzufügen folgender Fakten und
Axiome:

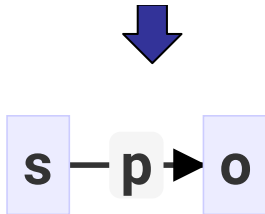


➡ automatische Herleitung der
ersten Möglichkeit!

Resource Description Framework (RDF)

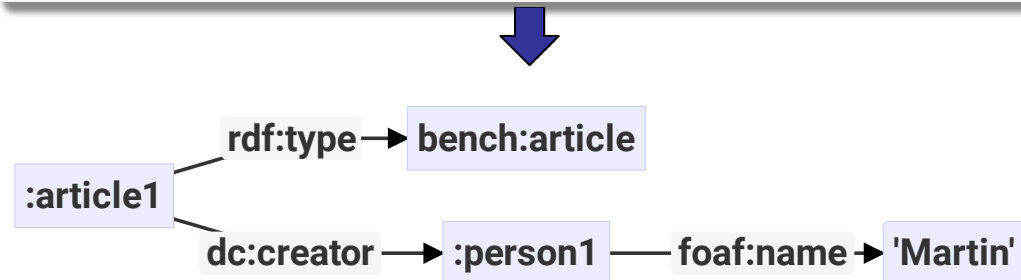
- **Datenformat** des Semantic Webs des World Wide Web Consortiums (W3C) (1999)
- RDF Daten

– **Menge**
von Tripeln
(s, p, o)



– Visualisierung als (RDF) **Graph**

```
:article1 rdf:type    bench:article.  
:article1 dc:creator  :person1.  
:person1  foaf:name   'Martin'.
```



Tripel - Definitionsbereich

- Grundbausteine

| Grundbaustein | Symbol | Bedeutung | Beispiele (in N3, Turtle, NTriples, ...) |
|--|----------|--|--|
| Inter-nationalized Resource Identifier (IRI) | <i>I</i> | (benannter) Bezeichner nicht unbedingt reale Objektadresse wie bei URL | Langform: <http://www.w3.org/2001/XMLSchema#integer> Kurzform: @prefix xsd: <http://www.w3.org/2001/XMLSchema#> xsd:integer |
| Blank Node | <i>B</i> | "für etwas ohne Namen " (anonymer) Bezeichner ohne Verwendung einer expliziten IRI | [], oder _:name, wobei name ein lokaler Bezeichner ist, der nicht persistent ist und nur während des Imports der gerade importierten Quelle gilt (zum Zweck des Mehrfachverwendens eines Blank Nodes in dieser Quelle). ➔ Blank Nodes mit demselben lokalen Bezeichner in unterschiedlichen Quellen sind verschiedene Blank Nodes! |
| Literal | <i>L</i> | Wert | Einfach: "hallo" Mit Datentyp typisiert: "1"^^xsd:integer Mit Sprachinfo: "Grill"^^<http://www.ex.org/myDataType> "hallo"@DE |

- Tripel $t \in \{I \cup B\} \times I \times \{I \cup B \cup L\}$

Syntaxzucker 1/2

- Objekt-Listen

```
_:b rdf:type bench:Journal .  
_:b rdf:type bench:Publication .  
_:b rdf:type bench:Media .
```



```
_:b rdf:type bench:Journal ,  
      bench:Publication ,  
      bench:Media .
```

- Prädikat-Objekt-Listen

```
_:b rdf:type bench:Journal .  
_:b dc:title "OJSW" .
```



```
_:b rdf:type bench:Journal ;  
    dc:title "OJSW" .
```

- Mischung

```
_:b rdf:type bench:Journal .  
_:b rdf:type bench:Publication .  
_:b dc:title "OJSW" .  
_:b dc:title "Op. J. of Semantic Web" .
```



```
_:b rdf:type bench:Journal ,  
      bench:Publication ;  
    dc:title "OJSW" ,  
             "Op. J. of Semantic Web" .
```

Syntaxzucker 2/2

- Blank Node-Darstellung

`[] p o .``[p o] .`

- Ausnutzung dieser Klammerung:

```
:bookshop1 v:sells [ rdf:type v:book;  
                     v:author "Herbert",  
                             "Josef" ]  
v:price "10"^^xsd:int.
```



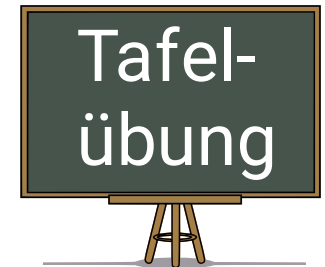
```
:bookshop1 v:sells _:b.  
_:b rdf:type v:book;  
    v:author "Herbert",  
            "Josef" ;  
v:price "10"^^xsd:int.
```

_:b nirgendwo anders verwendet...

RDF Collections

- Wie Listen als Tripel darstellen?

N3/Turtle: `("Friday" "Saturday" "Sunday")`



SPARQL Protocol And RDF Query Language (SPARQL)

Rekursives Acronym

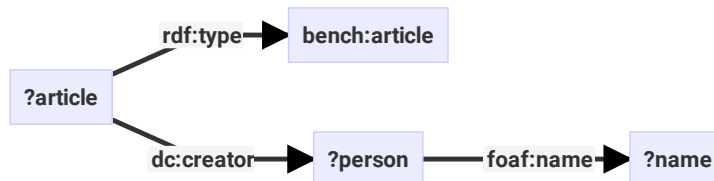
- (Graph) Anfragesprache für RDF Daten
- W3C Empfehlung seit Januar 2008
- Genauso mächtig wie Relationale Algebra
- Mit SPARQL 1.1 (März 2013) viele weitere Features von SQL übernommen
- Zusätzlich Protokoll standardisiert, um Anfragen an entfernte SPARQL Anfrageserver (sogenannte SPARQL Endpoints) zu stellen

Beispiel einer SPARQL-Anfrage

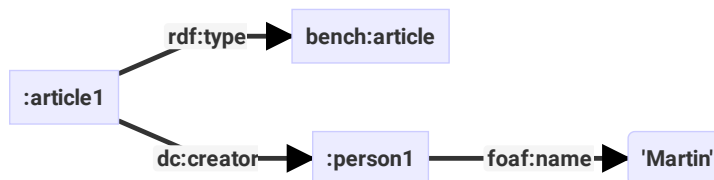
```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
...
SELECT ?article ?person ?name
WHERE {
  ?article rdf:type bench:article .
  ?article dc:creator ?person .
  ?person foaf:name ?name .
}
```




Anfragegraph:



Daten:



Resultat:



| ?article | ?person | ?name |
|-----------|----------|----------|
| :article1 | :person1 | 'Martin' |

Anfragegraph und Lösungen

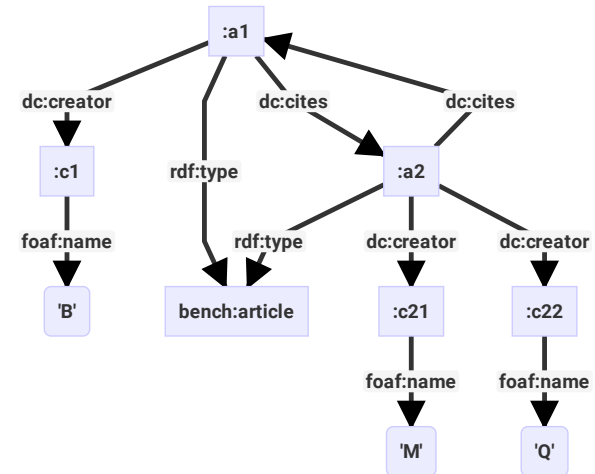


- Bestimme den **Anfragegraphen**, zeichne alle **Matchings** in den RDF-Graphen ein und gebe das **Ergebnis** der SPARQL-Anfrage an

Anfrage:

```
PREFIX ...  
SELECT *  
WHERE {  
  ?citingAuthor foaf:name ?nameOfCitingAuthor .  
  ?citedAuthor foaf:name ?nameOfCitedAuthor .  
  ?citingArticle rdf:type bench:Article .  
  ?citedArticle rdf:type bench:Article .  
  ?citingArticle dc:cites ?citedArticle .  
  ?citingArticle dc:creator ?citingAuthor .  
  ?citedArticle dc:creator ?citedAuthor .  
}
```

RDF Graph:



SPARQL Anfragearten

- **SELECT**
 - Gibt alle oder eine Teilmenge der Variablen (und deren gebundenen Werte) einer Anfrage zurück
- **CONSTRUCT**
 - Gibt RDF Daten zurück, die ermittelt worden sind auf Grund der Substitution von Variablen in Tripel Templates
- **DESCRIBE**
 - Gibt RDF Daten zurück, die die gefundenen Resultate beschreiben
- **ASK**
 - Gibt wahr/falsch zurück basierend auf einen Match des Anfragemusters

SPARQL Anfrage - Union Operator

- \cong relationaler Op. der äußeren Vereinigung \uplus
 - Wie Vereinigung \cup , aber auch mit inkompatiblen Schemata
 - Schema des Ergebnisses ist Vereinigung der Attributmengen
 - Fehlende Werte werden mit Nullwerten \perp ergänzt

Daten:

```
@prefix dc10: <http://purl.org/dc/elements/1.0/>.
@prefix dc11: <http://purl.org/dc/elements/1.1/>.
:book1 dc10:title 'Semantic Web Starter'.
:book1 dc11:title 'Advanced Semantic Web'.
```



Anfrage:

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT ?title1 ?title2
WHERE { { ?book dc10:title ?title1 }
        UNION
        { ?book dc11:title ?title2 }
}
```



Resultat:

| ?title1 | ?title2 |
|------------------------|-------------------------|
| 'Semantic Web Starter' | |
| | 'Advanced Semantic Web' |

SPARQL Anfrage - Optional

Daten:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
_:a rdf:type foaf:Person.
_:a foaf:name 'Alice'.
_:a foaf:mbox <mailto:alice@example.com>.
_:a foaf:mbox <mailto:alice@work.example>.
_:b rdf:type foaf:Person.
_:b foaf:name 'Bob'.
```



Anfrage:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
  ?x foaf:name ?name .
  OPTIONAL {
    ?x foaf:mbox ?mbox .
  }
}
```



Resultat:

| ?name | ?mbox |
|---------|-----------------------------|
| 'Alice' | <mailto:alice@example.com> |
| 'Alice' | <mailto:alice@work.example> |
| 'Bob' | |

\cong relationaler Op. des
links-äußeren Joins \bowtie

$$R \bowtie S \equiv (R - (R \ltimes S)) \uplus (R \ltimes S)$$

mit Semi-Join

$$R \ltimes S \equiv R \bowtie \pi_{Attr_R \cap Attr_S}(S)$$

Sortierte Resultate - ORDER BY

- Beispiel 1:

```
PREFIX      : <http://example.org/ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name WHERE {
    ?x foaf:name ?name ; :empId ?emp.
}
ORDER BY ?name
```

- Beispiel 2:

```
... ORDER BY DESC(?emp)
```

- Beispiel 3:

```
... ORDER BY ASC(?name) DESC(?emp)
```

- \cong relationaler Op. der
Sortierung $\tau_{A_1, \dots, A_n}(R)$
 - Ergebnis ist keine (Multi-)Menge, sondern Liste
 - In DBMS wird häufig auch zwischendurch sortiert (oder auf vorsortierte Eingaben aus einem Index zugegriffen), um optimierte Algorithmen anzuwenden, die sortierte Eingaben erfordern

FILTER-Klausel und Builtin-Funktionen

- **regex** gleicht reguläre Ausdrücke ab...
(äquivalent zur XPath built-in Funktion **fn:matches**)

Beispiel:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { ?x dc:title ?title
        FILTER(regex(?title, "SPARQL")) }
```

- Auswahl weiterer Builtin-Funktionen

| Operator | Beschreibung | Typ des Operanden | Rückgabebetyp |
|---|------------------------------|-------------------|---------------|
| bound (A) | Ist die Variable A gebunden? | Variable | xsd:boolean |
| isIRI (A) oder isURI (A) | Ist A eine IRI/URI? | RDF Term | xsd:boolean |
| isBlank (A) | Ist A ein Blank Node? | RDF Term | xsd:boolean |
| isLiteral (A) | Ist A ein Literal? | RDF Term | xsd:boolean |

Built-In Funktionen – Logische Operatoren und Vergleiche

| Operator | Beschreibung | Typ des/der Operanden | Rückgabotyp |
|------------------------|---|-----------------------|-------------|
| ! A | Logisches NOT | xsd:boolean | xsd:boolean |
| A B | Logisches OR | xsd:boolean | xsd:boolean |
| A && B | Logisches AND | xsd:boolean | xsd:boolean |
| A = B | Gleichheit ¹ von A und B | RDF Term | xsd:boolean |
| A != B | Ungleichheit ¹ von A und B | RDF Term | xsd:boolean |
| sameTerm (A, B) | Sind A und B dieselben RDF Terme? Umgehen von Problematik ¹ ! | RDF Term | xsd:boolean |

¹ Schmeißen von Fehlern bei nicht unterstützten Datentypen und ungleichen lexikalischen Werten, z.B. bei

```
"iiii"^^my:romanNumeral = "iv"^^my:romanNumeral und
"iiii"^^my:romanNumeral != "iv"^^my:romanNumeral
```

XPath Tests und Arithmetik

- Tests

| Operator | Beschreibung | Typ der Operanden | Rückgabotyp |
|------------|-----------------------------|---------------------|-------------|
| $A < B$ | A kleiner als B | XML Schema Datentyp | xsd:boolean |
| $A \leq B$ | A kleiner als oder gleich B | XML Schema Datentyp | xsd:boolean |
| $A > B$ | A größer als B | XML Schema Datentyp | xsd:boolean |
| $A \geq B$ | A größer als oder gleich B | XML Schema Datentyp | xsd:boolean |

Zusätzlich bei $A=B$ und $A \neq B$ Interpretation der Operanden falls Verwendung von XML Schema Datentypen:

`"+2"^^xsd:integer = "2"^^xsd:integer`

- Arithmetik

| Operator | Beschreibung | Typ der Operanden | Rückgabotyp |
|----------|----------------------------|-------------------|-------------|
| $A + B$ | Summe von A und B | Numerisch | Numerisch |
| $A - B$ | Differenz von A und B | Numerisch | Numerisch |
| $A * B$ | Multiplikation von A und B | Numerisch | Numerisch |
| A / B | Division von A und B | Numerisch | Numerisch |

Weitere (hier nicht besprochene) Features von SPARQL

- SPARQL 1.0
 - Weitere Built-In Funktionen
 - **Construct-/Ask-** und **Describe**-Anfragen, **Benannte Graphen**
 - **LIMIT/OFFSET**
- SPARQL 1.1
 - **Minus, (Not) Exists, Gruppieren, Aggregationsfunktionen**
 - **Pfad**ausdrücke
 - **Unterabfragen, Select-Ausdrücke, explizite Zuweisungen**
 - Absetzen von **Unterabfragen** zu entfernten **SPARQL Endpoints**
 - **Updates**
 - Mächtige Built-In Funktionsbibliothek
- ➡ Veranstaltung **Semantic Web** (Vertiefungsmodul **Datenmanagement**) bzw. nachzulesen in SPARQL-Spezifikationen

Zusammenfassung

- **RDF als Datenformat für Semantic Web Daten**
 - Grundkonstrukt **Tripel** (s, p, o)
 - RDF Datensatz als **Menge von Tripeln**
- **SPARQL als RDF-Anfragesprache**
 - Ideal angepasste Anfragesprache für das Semantic Web und RDF
 - Grundkonstrukt **Tripelmuster** (s, p, o)
 - **Zusätzlich** zu Tripeln **Variablen** (Platzhalter) als Komponenten des Tripelmusters möglich
 - **Sehr mächtig**, insbesondere **SPARQL 1.1**