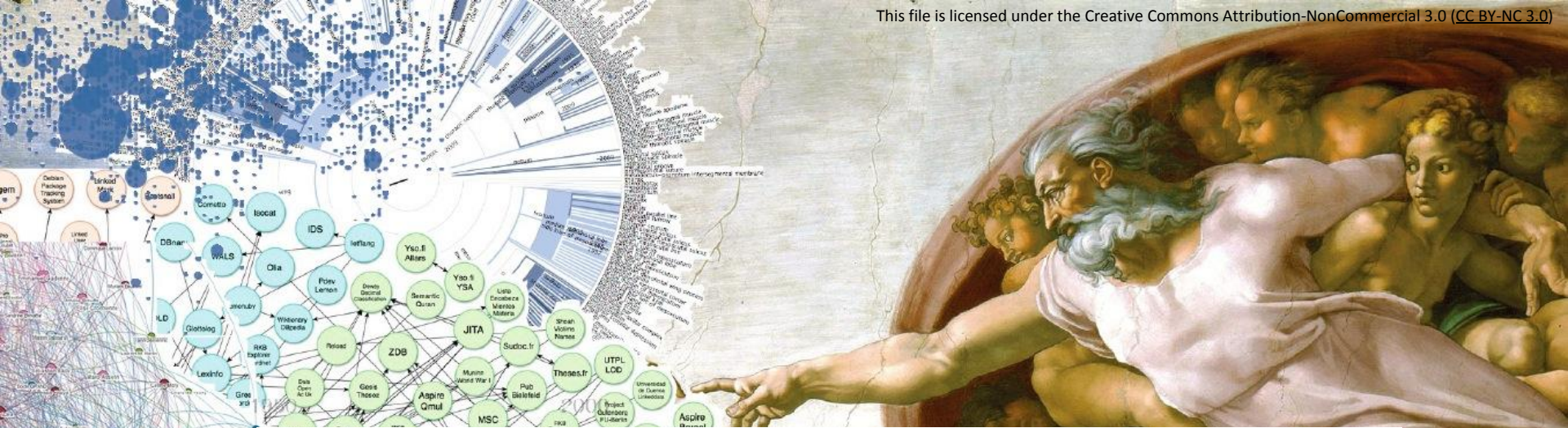




Information Service Engineering

Lecture 4 - Linked Data Engineering II

4.3 Web Ontology Language OWL II



FIZ Karlsruhe

Leibniz Institute for Information Infrastructure



Hasso
Plattner
Institut



Karlsruher Institut für Technologie

Prof. Dr. Harald Sack & Dr. Maria Koutraki
FIZ Karlsruhe

Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Spring 2018

Information Service Engineering

Lecture 4: Linked Data Engineering II

4.1 Linked Data Principles

4.2 Web Ontology Language OWL

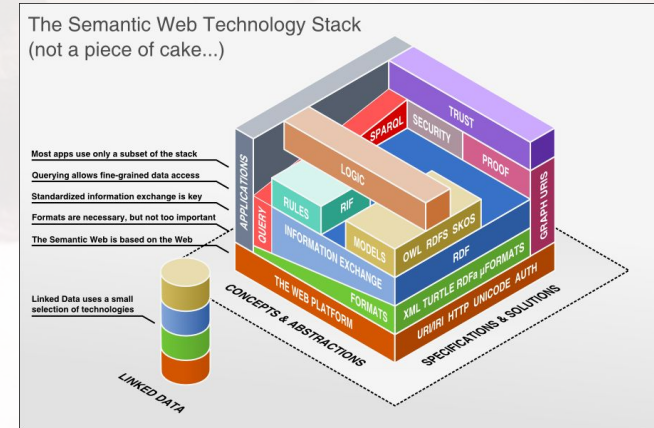
4.3 Web Ontology Language OWL II

4.4 How to Query RDF with SPARQL

4.5 Complex Queries with SPARQL

4.6 More Complex SPARQL Queries

4.7 Linked Data Knowledge Graphs



OWL Complex Classes - Nominals

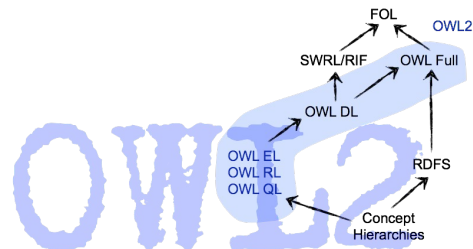
Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Novel a owl:Class .
:AnimalFarm a :Novel .
:NineteenEightyfour a :Novel .

:NovelsInStore a owl:Class ;
  owl:oneOf
    ( :AnimalFarm
      :NineteenEightyfour ) .
```

$\text{NovelsInStore} \sqsubseteq \{\text{AnimalFarm}, \text{NineteenEightyfour}\}$

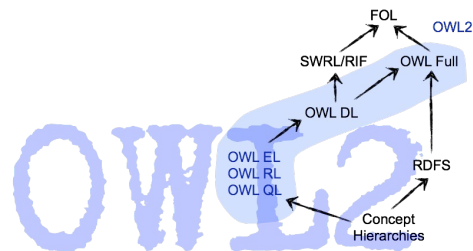
- *There are only two novels available in the store.*



OWL Logical Class Constructors

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

- logical AND (conjunction): `owl:intersectionOf` $\sqcap$
- logical OR (disjunction): `owl:unionOf` $\sqcup$
- logical negation: `owl:complementOf` $\neg$
- Logical constructors are applied to **create complex class descriptions** from atomic classes.



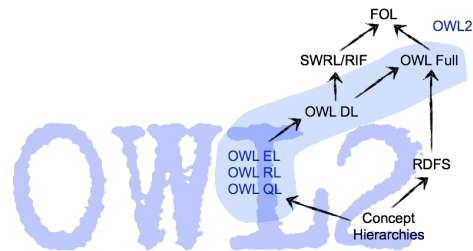
OWL Logical Class Constructors - INTERSECTION

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Book a owl:Class .  
:ThingsInStore a owl:Class .  
:BooksInStore a owl:Class ;  
    owl:intersectionOf (:ThingsInStore :Book) .
```

$\text{BooksInStore} \equiv \text{ThingsInStore} \sqcap \text{Books}$

- The class `:BooksInStore` results from the intersection of all individuals of the classes `:ThingsInStore` and `:Book`



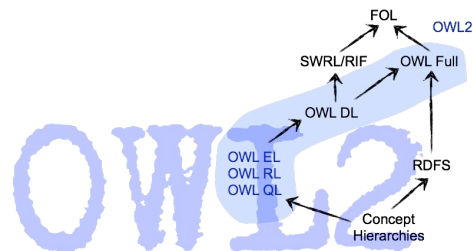
OWL Logical Class Constructors - UNION

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Book a owl:Class ;  
    owl:equivalentClass [  
        owl:unionOf ( :Novel  
                        :Poetry  
                        :NonFiction )  
    ] .
```

Book $\equiv$ Novel $\sqcup$ Poetry $\sqcup$ NonFiction

- Novels, poetry, and non-fiction are also books



OWL Logical Class Constructors - NEGATION

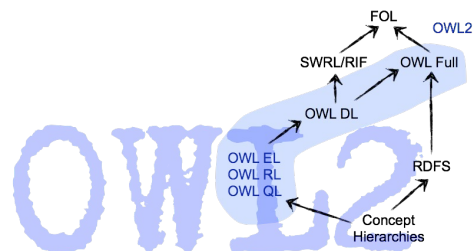
Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Book a owl:Class ;  
    rdfs:subClassOf [  
        owl:complementOf :Writer  
    ] .
```

Book $\sqsubseteq$ $\neg$ Writer

- semantically equivalent assertion:

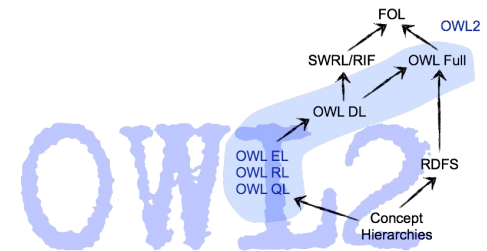
```
:Book a owl:Class ;  
    owl:disjointWith :Writer .
```



OWL Property Restrictions

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

- OWL property restrictions are used to describe complex classes via properties
- restrictions on values:
 - `owl:hasValue`
 - `owl:allValuesFrom`
 - `owl:someValuesFrom`
- restrictions on cardinality:
 - `owl:cardinality`
 - `owl:minCardinality`
 - `owl:maxCardinality`



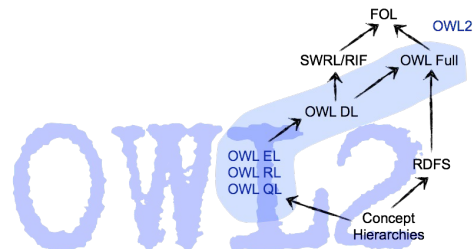
OWL Property Restrictions with CONSTANTS

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:OrwellsBooks a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :author ;  
      owl:hasValue :GeorgeOrwell ] .
```

OrwellsBooks $\sqsubseteq$ author.(GeorgeOrwell)

- Class :OrwellsBooks is described via fixed value assignment (=constant) of the individual :GeorgeOrwell to the property :author



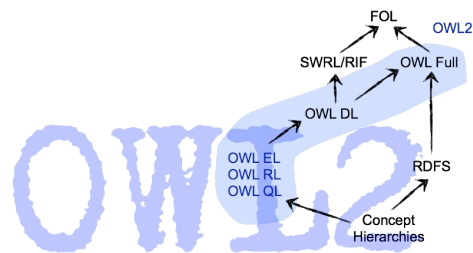
OWL Property Restrictions with STRICT BINDING

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Poetry a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :author ;  
      owl:allValuesFrom :Poet ] .
```

Poetry $\sqsubseteq$ $\forall \text{author.Poet}$

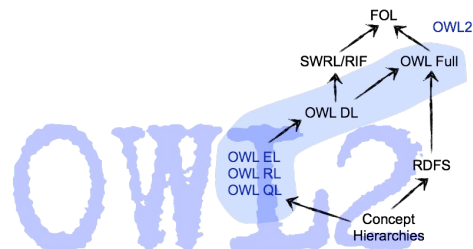
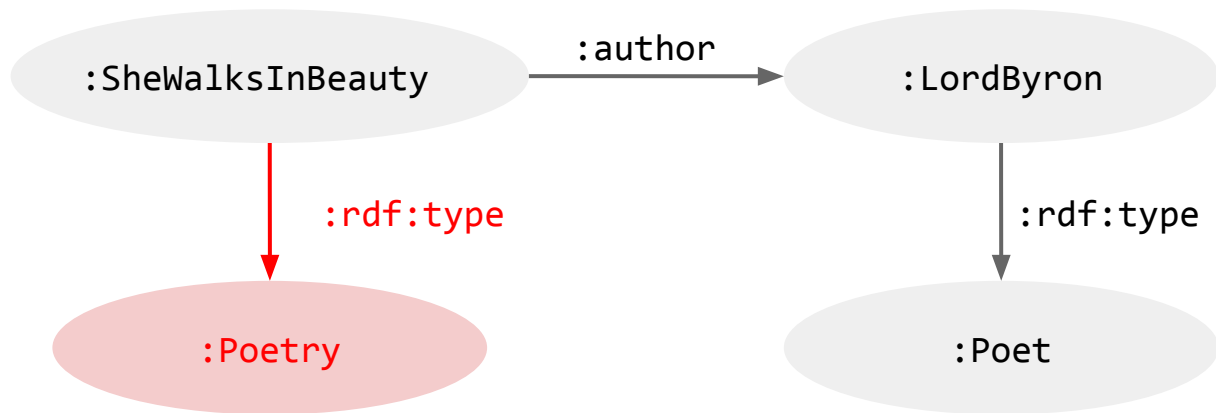
- **owl:allValuesFrom**
fixes all instances of a specific class C
as allowed range for a property p
(strict binding) $\forall p.C$



OWL Property Restrictions with STRICT BINDING

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Poetry a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :author ;  
      owl:allValuesFrom :Poet ] .
```



OWL Property Restrictions with LOOSE BINDING

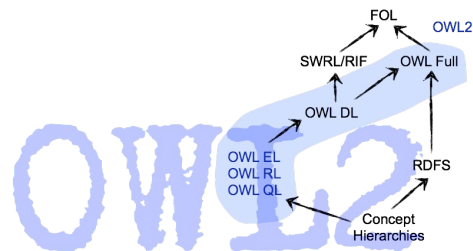
Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Reader a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :reads ;  
      owl:someValuesFrom :Book ] .
```

Reader $\sqsubseteq \exists \text{reads}.\text{Book}$

- **owl:someValuesFrom**

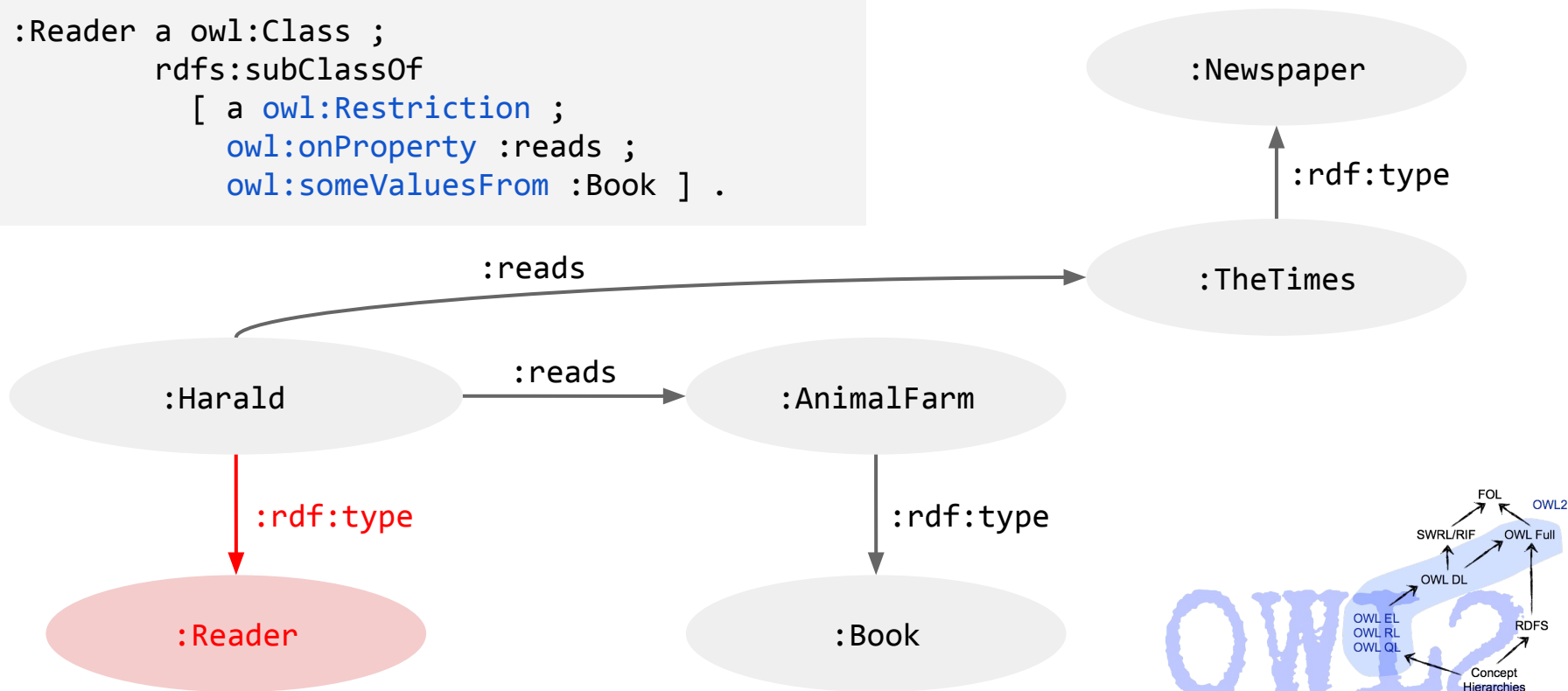
describes that there must exist an individual for p
and fixes its range to class C (loose binding) $\exists p.C$



OWL Property Restrictions with LOOSE BINDING

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Reader a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :reads ;  
      owl:someValuesFrom :Book ] .
```



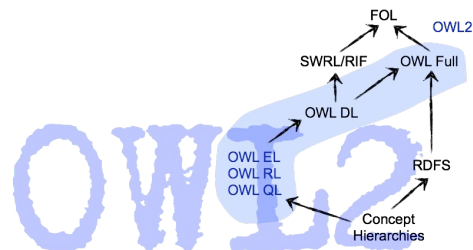
OWL Property Restrictions with CARDINALITIES

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

```
:Tetralogy a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :hasVolumes ;  
      owl:cardinality 4 ] .
```

Tetralogy $\sqsubseteq$ (=4)hasVolumes

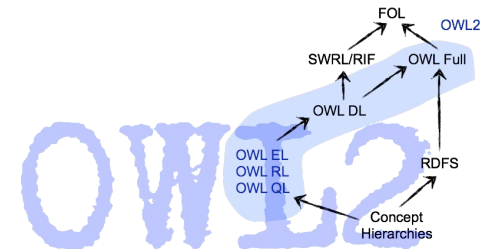
- **owl:cardinality** restricts to an exact number
- **owl:minCardinality** / **owl:maxCardinality** restricts to upper / lower bounds



Strict or Loose Binding?

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

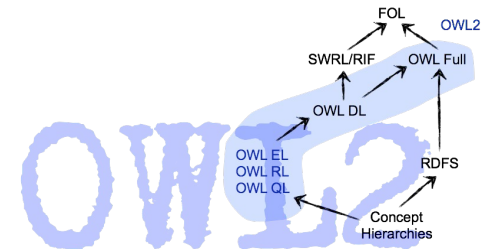
- **Vegetarian** a owl:Class ;
 rdfs:subClassOf [
 a owl:Restriction ;
 owl:onProperty :eats ;
 owl:allValuesFrom [
 owl:complementOf :Animal .
]
]



Strict or Loose Binding?

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.3 Web Ontology Language OWL II

- **Driver** a owl:Class ;
 rdfs:subClassOf [
 a owl:Restriction ;
 owl:onProperty :drives ;
 owl:someValuesFrom :Car .
]



OWL DL - Language Summary

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.2 Web Ontology Language OWL

Class Expressions

- Class names A, B
- Conjunction $C \sqcap D$
- Disjunction $C \sqcup D$
- Negation $\neg C$
- Exist. property restriction $\exists R.C$
- Univ. property restriction $\forall R.C$
- Self $\exists S.Self$
- Greater-than $\geq n \ S.C$
- Less-than $\leq n \ S.C$
- Enumerated classes $\{a\}$

Properties

- Property names R, S, T
- Simple properties S, T
- Inverse properties R^{-}
- Universal property U

Tbox (Class axioms)

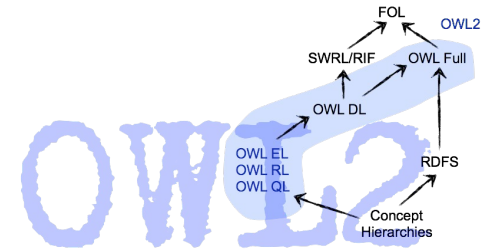
- Inclusion $C \sqsubseteq D$
- Equivalence $C \equiv D$

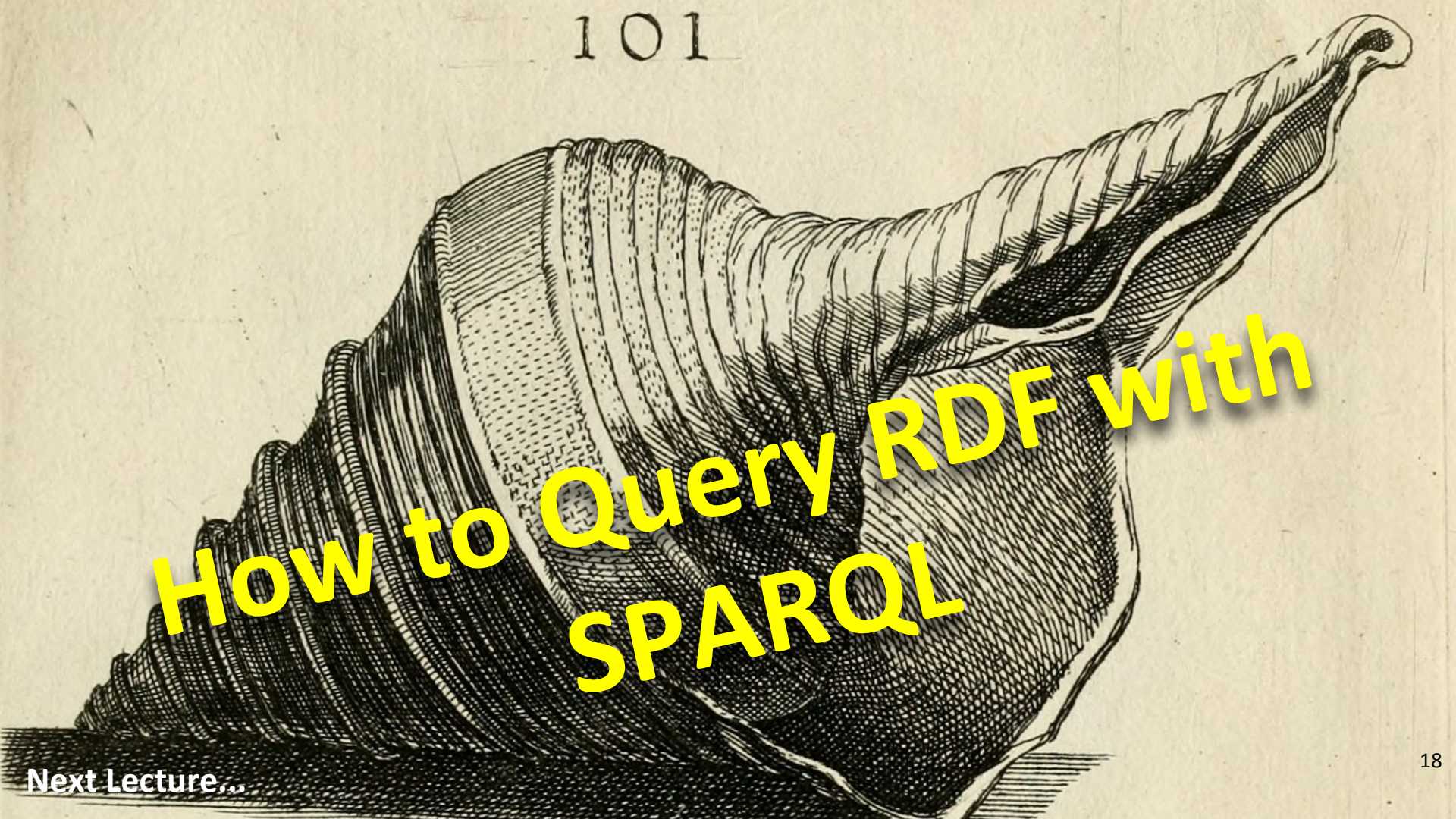
Rbox (Property Axioms)

- Inclusion $R_1 \sqsubseteq R_2$
- General Inclusion $R^{(-)}_1 \circ R^{(-)}_2 \circ \dots \circ R^{(-)}_n \sqsubseteq R$
- Transitivity
- Symmetry
- Reflexivity
- Irreflexivity
- Disjointiveness

Abox (Facts)

- Class membership $C(a)$
- Property relation $R(a, b)$
- Negated property relation $\neg S(a, b)$
- Equality $a=b$
- Inequality $a \neq b$





How to Query RDF with SPARQL

Next Lecture...

Picture References:

- P.17: Bonanni F. 1681. Ricreatione dell' occhio e dela mente nell oservation' delle Chiociolle, proposta a' curiosi delle opere della natura, c. 1681. Varese, Rome, xiv, 384 pp., 109 plates. figure 101
https://commons.wikimedia.org/wiki/File:Syrinx_aruanus_shell_4.jpg