



Knowledge Graphs

Lecture 4 – Ontologies as Key to Knowledge Representation

4.4 From simple to complex: Scaling up with OWL

Prof. Dr. Harald Sack

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

AIFB – Karlsruhe Institute of Technology

Autumn 2023



FIZ Karlsruhe

Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 4: Ontologies as Key to Knowledge Representation

4.1 From Aristotle to AI: Exploring Ontologies in Computer Science

4.2 The Crucial Role of Mathematical Logic

Excursion 5: Essential Logics in a Nutshell

Excursion 6: Description Logics

4.3 The Web Ontology Language OWL

4.4 From simple to complex: Scaling up with OWL

4.5 Unlocking the Potential of OWL

The Semantic Web Technology Stack (not a piece of cake...)

Most apps use only a subset of the stack

Querying allows fine-grained data access

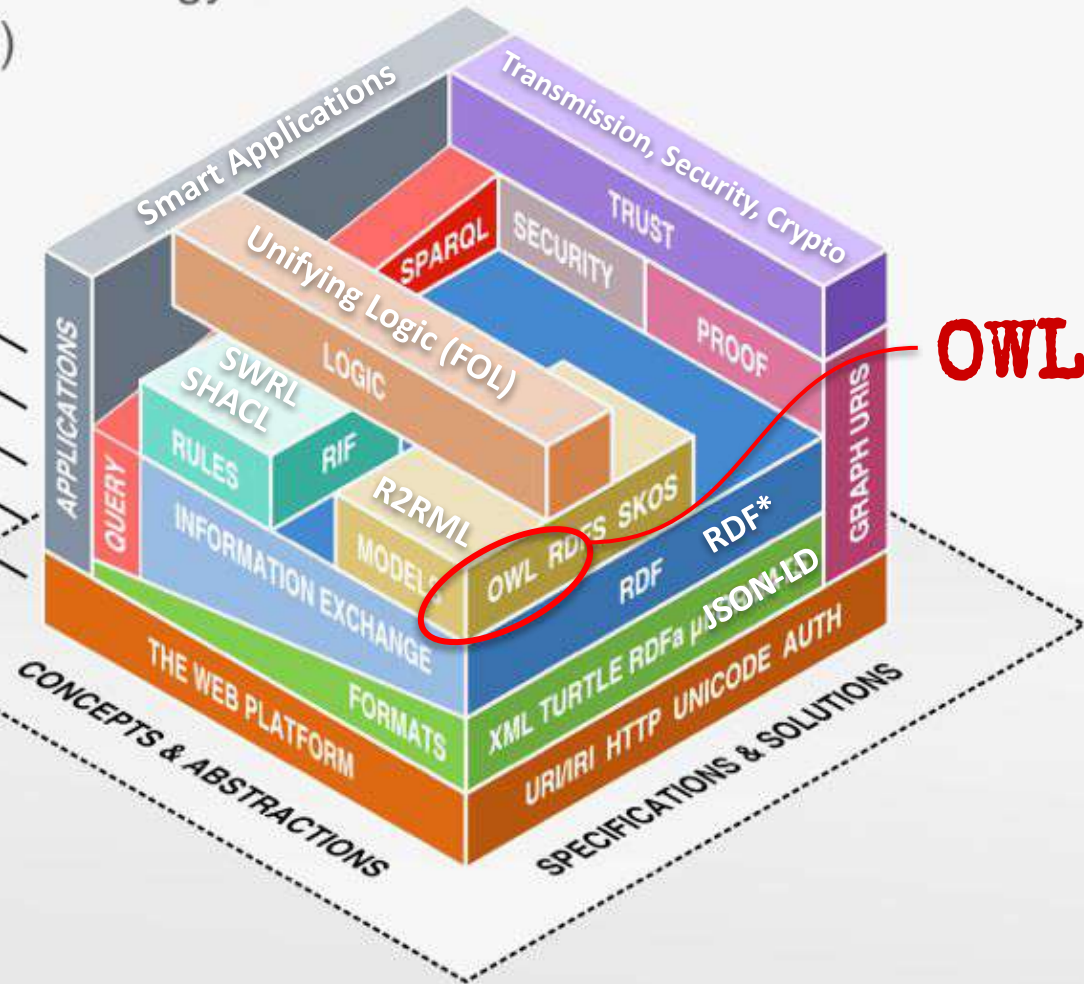
Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small
selection of technologies

LINKED DATA



OWL2 Complex Classes – Nominals

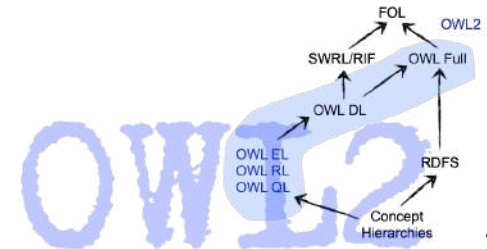
```
:Novel a owl:Class .  
:Foundation a :Novel .  
:FoundationAndEmpire a :Novel .  
:SecondFoundation a :Novel .  
  
:FoundationTrilogy a owl:Class ;  
  owl:oneOf  
    ( :Foundation  
      :FoundationAndEmpire  
      :SecondFoundation ) .
```



FoundationTrilogy \sqsubseteq { Foundation,
FoundationAndEmpire,
SecondFoundation }

[3]

The Foundation Trilogy consists of three novels.



4

OWL2 Logical Class Constructors

- logical AND (conjunction):
- logical OR (disjunction):
- logical negation:

`owl:intersectionOf`

□

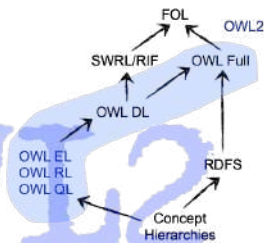
`owl:unionOf`

⊔

`owl:complementOf`

¬

used to create complex
classes from atomic classes

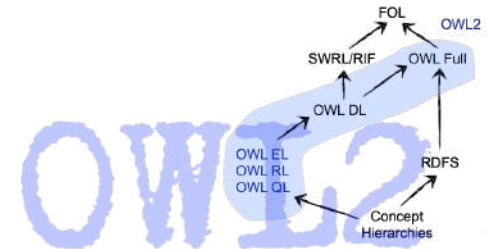


OWL2 Logical Class Constructors – Intersection

```
:Scientist a owl:Class .  
:Author a owl:Class .  
:ScientificAuthor a owl:Class ;  
    owl:intersectionOf (:Scientist :Author) .
```

ScientificAuthor \equiv Scientist \sqcap Author

"Scientific authors are authors, who are also scientists."

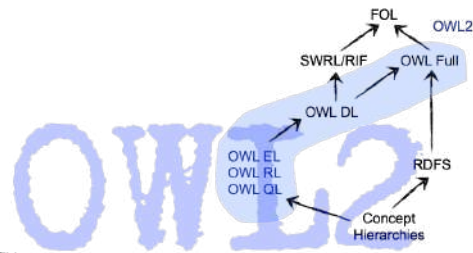


OWL2 Logical Class Constructors – Union

```
:Environmentalist a owl:Class ;  
  owl:equivalentClass [  
    owl:unionOf ( :ClimateActivist  
                  :AnimalRightsActivist  
                  :EnergySaver )  
  ] .
```

Environmentalist \equiv ClimateActivist \sqcup AnimalRightsActivist
 \sqcup EnergySaver

"Climate activists, animal rights activists, and energy savers
are all environmentalists."

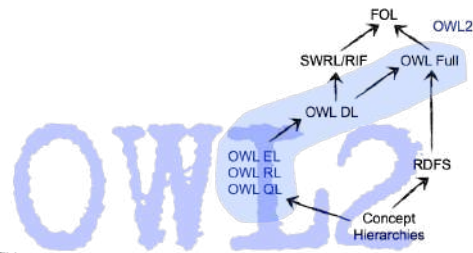


OWL2 Logical Class Constructors – Negation

```
:Dystopia a owl:Class .  
:Utopia a owl:Class ;  
    owl:complementOf (:Dystopia) .
```

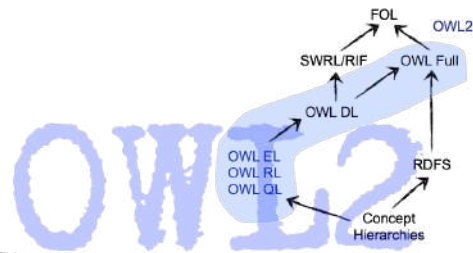
Utopia \equiv \neg Dystopia

"Utopia is the complement of Dystopia."



OWL2 Property Restrictions

- **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`



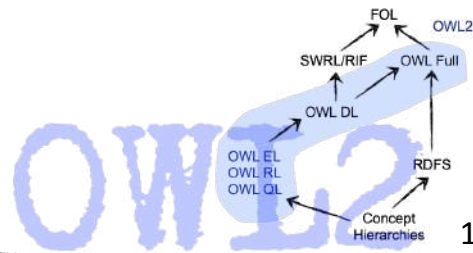
OWL2 Property Restrictions with Constants

```
:AsimovsWritings a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :author ;  
      owl:hasValue :IsaacAsimov ] .
```

AsimovsWritings \sqsubseteq \exists author.{IsaacAsimov}

The class `:AsimovsWritings` is described via fixed value assignment (=constant) of the individual `:IsaacAsimov` to the property `:author`.

"Asimov's writings are authored by Isaac Asimov"



OWL2 Property Restrictions with Strict Binding

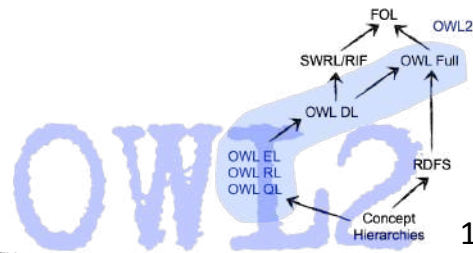
```
:VegetarianDish a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :ingredient ;  
      owl:allValuesFrom :VegetarianFood ] .
```

VegetarianDish \sqsubseteq
 $\forall \text{ingredient. VegetarianFood}$

"vegetarian dishes contain only vegetarian ingredients."

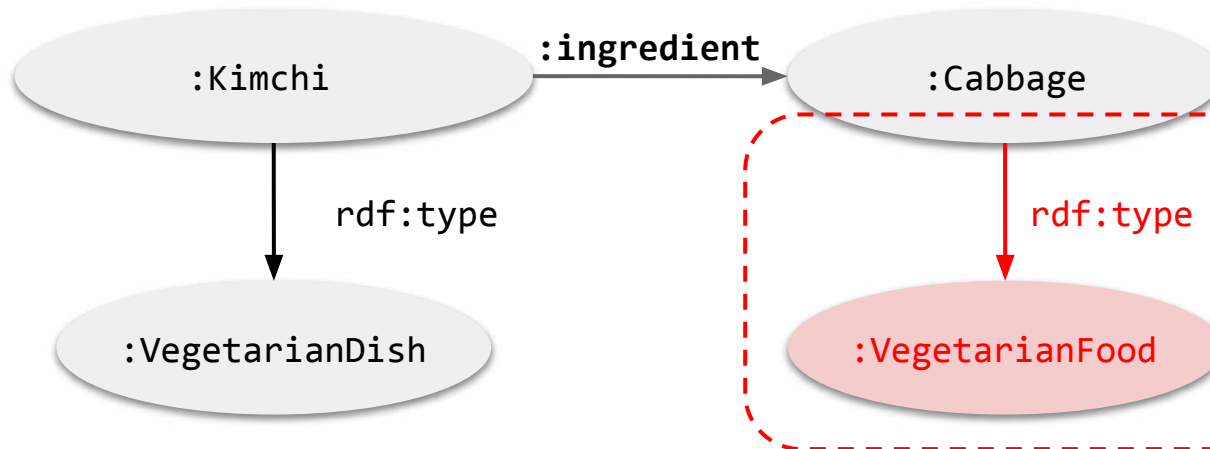
owl:allValuesFrom

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

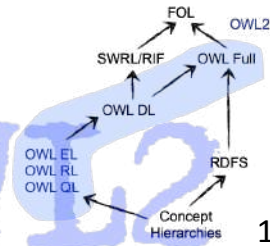


OWL2 Property Restrictions with Strict Binding

```
:VegetarianDish a owl:Class ;  
  rdfs:subClassOf  
    [ a owl:Restriction ;  
      owl:onProperty :ingredient ;  
      owl:allValuesFrom :VegetarianFood ] .
```



Logical inference



OWL2

OWL2 Property Restrictions with Loose Binding

```

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

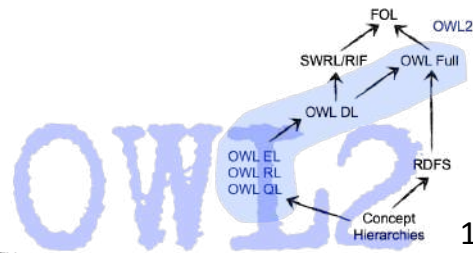
```

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

"A reader is somebody who reads (amongst other things) books."

owl:someValuesFrom

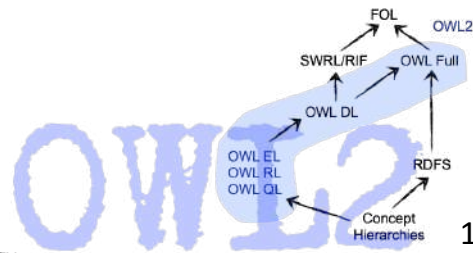
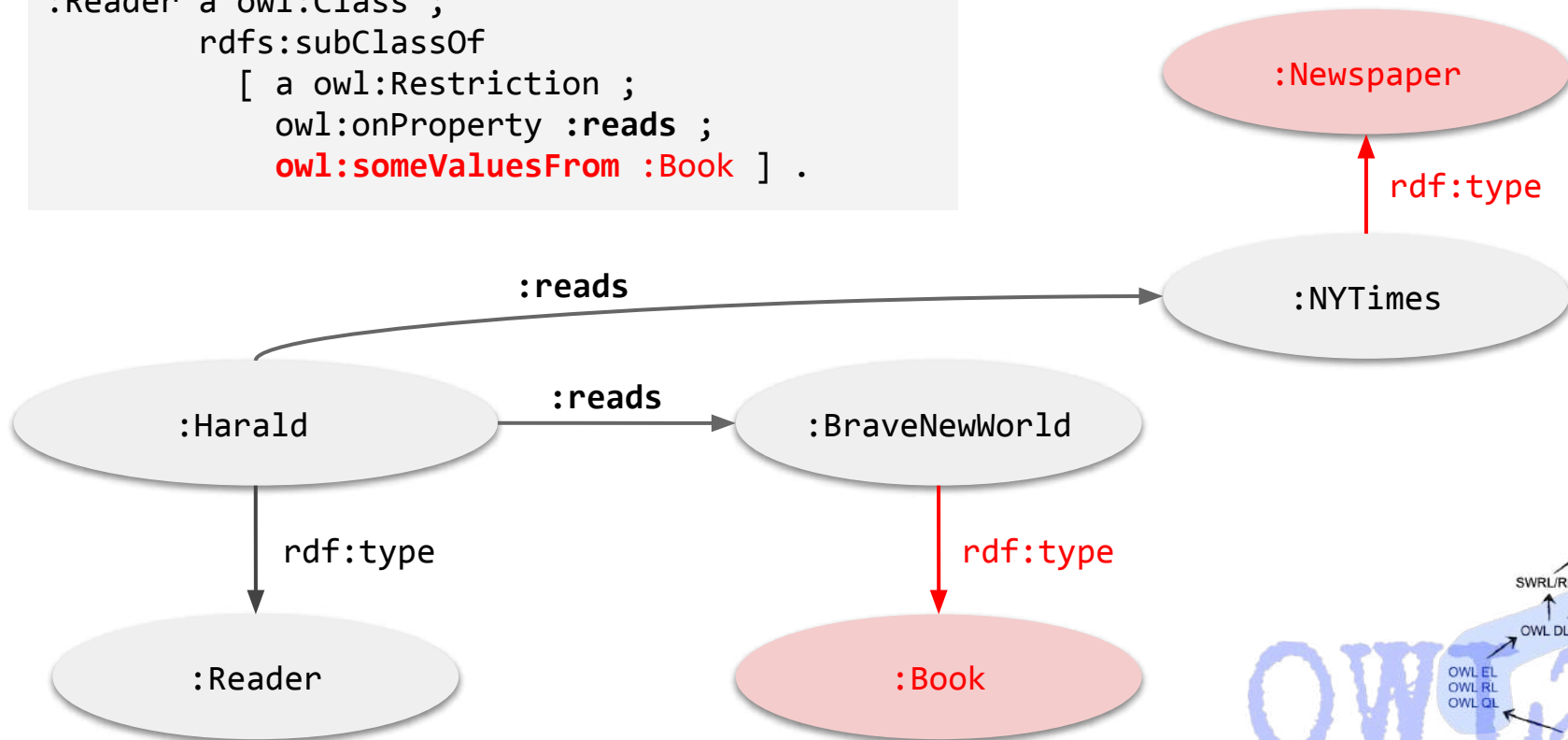
describes that there must exist an individual for p
and fixes its range to class C (loose binding) $\exists p.C$



OWL2

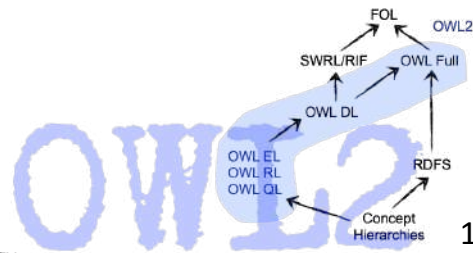
OWL2 Property Restrictions with Loose Binding

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



OWL2 Property Restrictions with Loose Binding

- 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`



OWL2 Property Restrictions with Cardinality

```

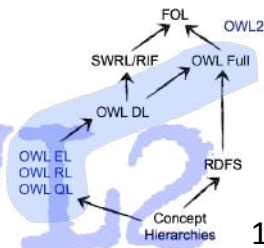
:Trilogy a owl:Class ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty :volume ;
      owl:cardinality "3"^^<http://www.w3.org/2001/XMLSchema#int> ] .

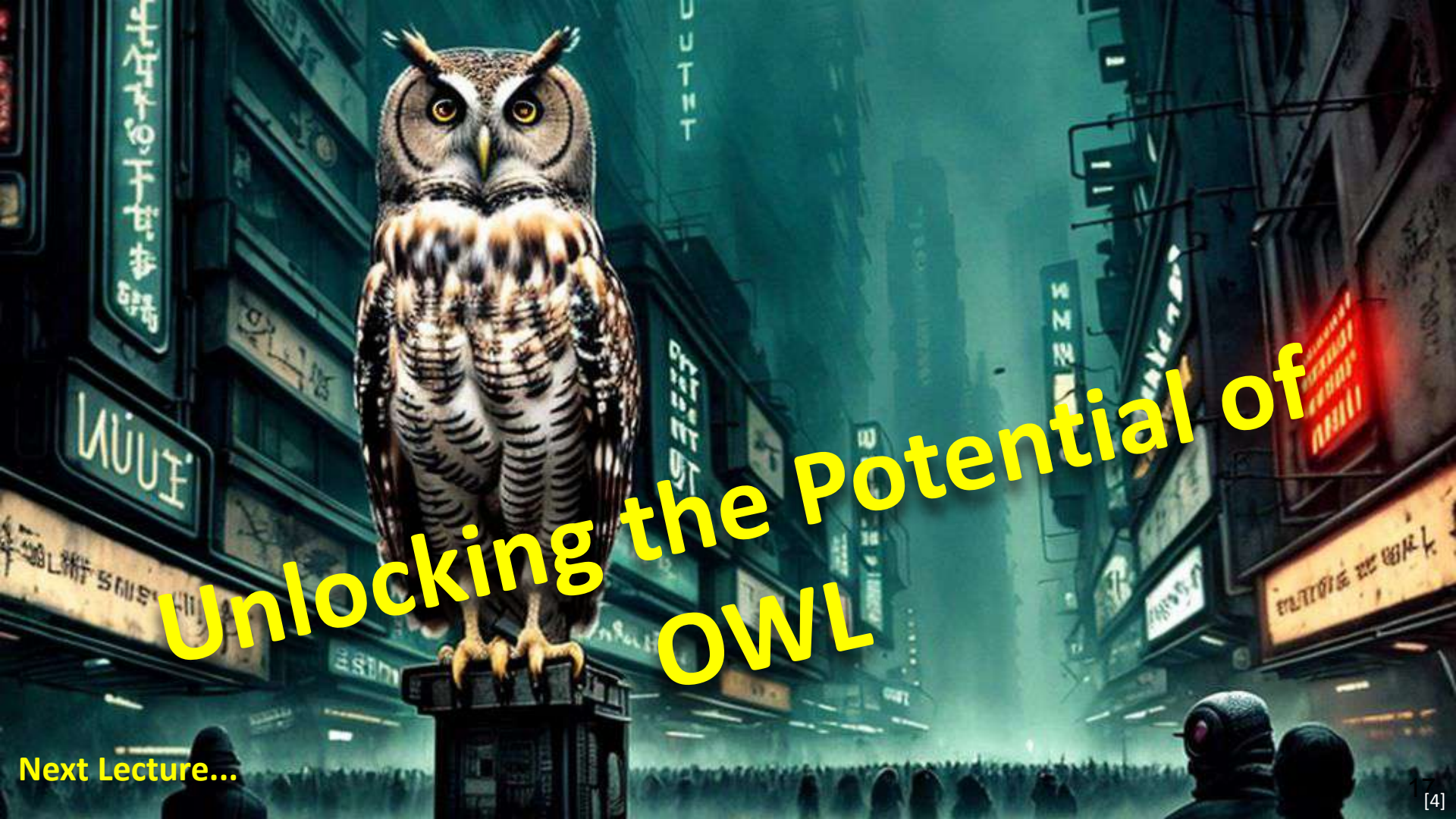
```

Trilogy \sqsubseteq =3.volume.⊤

"A trilogy always consists of 3 volumes."

- Class **:Trilogy** is restricted to exactly 3 volumes, i.e. any instance of **:Trilogy** must have exactly 3 values for the property **:volume**.
- For **owl:maxCardinality** and **owl:minCardinality** the restriction gives upper and lower bounds on property value cardinalities.





Unlocking the Potential of OWL

Next Lecture...

Bibliographic References:

- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph (eds., 2012), [*OWL 2 Web Ontology Language Primer \(Second Edition\)*](#), W3C Recommendation 11 December 2012.
- Aidan Hogan (2020), [*The Web of Data*](#), Springer.
 - Chap. 5.4.2 Properties, 200–215.
 - Chap. 5.4.3 Classes, 215–243.

Picture References:

- [1] “Several owls are walking on a crowded street in a Bladerunner like dystopian city environment.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>
- [2] Benjamin Nowack, *The Semantic Web - Not a Piece of cake ...*, at bnode.org, 2009-07-08 , [CC BY 3.0], <https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>
- [3] Dust-jacket by David Kyle of Foundation, by Isaac Asimov (1951), Dust-jacket by Edd Cartier of Foundation and Empire, by Isaac Asimov (1952), Dust-jacket by Ric Binkley of Second Foundation, by Isaac Asimov (1953), Gnome Press, [Fair Use] via WikiCommons: https://en.wikipedia.org/wiki/File:Foundation_gnome.jpg
- [4] “A large owl watching over a crowded street in a Bladerunner like dystopian city environment.”, created via ArtBot, Deliberate, 2023, [CC-BY-4.0], <https://tinybots.net/artbot>