



# Knowledge Engineering with Semantic Web Technologies

## Lecture 4: OWL, Rules and Reasoning 4.4 OWL Complex Classes



Dr. Harald Sack  
Hasso Plattner Institute for IT Systems Engineering  
University of Potsdam  
Autumn 2015

# OWL Closed Classes - Nominals

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

NovelsInStore  $\sqsubseteq$  {AnimalFarm, NineteenEightyfour}

- There are only two novels available in the store.

# OWL Logical Class Constructors

- 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

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

BooksInStore  $\equiv$  ThingsInStore  $\sqcap$  Books

- The class “BooksInStore” results from the intersection of all individuals of the classes “ThingsInStore” and “Book”

# OWL Logical Class Constructors

```
: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

```
: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

- 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

```
: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 Restriction with Strict Binding

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

Poetry  $\sqsubseteq$   $\forall$ 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 Restriction with Loose Binding

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

Reader  $\sqsubseteq$   $\exists$  reads.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 Cardinality Restrictions

```
: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



## **05 OWL Property Restrictions**

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

**Lecture 4: OWL, Rules, and Reasoning**