

OWL

- RDF und RDFS sehr schwach
 - ♦ Keine Widersprüche möglich!
- Nicht möglich in RDF und RDFS:
 - ♦ „Jede Vorlesung hat mindestens 2 Hörer“
 - ♦ „Lehrveranstaltungen sind immer Vorlesungen, Übungen oder Seminare“
 - ♦ „Koblenzer Fachbereiche sind genau FB1, FB2, FB3, FB3“
 - ♦ „Der Professor betreut seine wissenschaftlichen Mitarbeiter und die von den Mitarbeitern betreuten Studenten“
- Beliebige Sprache zu ausdrucksstark
- Was ist also eine Sprache, die gerade so ausdrucksstark ist, dass man große Schemata/Ontologien beschreiben kann?
 - ♦ OWL – Web Ontology Language

OWL Versionen

Syntaktisch:

OWL Lite ½ OWL DL ½ OWL 1.1

OWL DL ½ OWL Full

Semantisch:

Falls KB in OWL Lite, dann KB auch in OWL DL

Falls KB in OWL DL, dann KB auch in OWL FULL

Falls F eine Schlußfolgerung aus KB in OWL Lite, dann ist F auch eine Schlußfolgerung aus KB in OWL DL.

Falls F eine Schlußfolgerung aus KB in OWL DL ist und KB in OWL Lite, dann ist F eine Schlußfolgerung aus KB in OWL lite.

Falls F eine Schlußfolgerung aus KB in OWL DL, dann ist F auch eine Schlußfolgerung aus KB in OWL FULL.

NICHT!: Falls F eine Schlußfolgerung aus KB in OWL FULL ist und KB in OWL DL, dann ist F eine Schlußfolgerung aus KB in OWL DL.

OWL Header

```
<owl:Ontology rdf:about=„>
  <rdfs:comment
    rdf:datatype=„http://www.w3.org/2001/XMLSchema#string“
    >SWRC Ontologie in der Version vom Dez
    05</rdfs:comment>
  <owl:versionInfo>v0.5</owl:versionInfo>
  <owl:imports rdf:resource=„http://www.example.org/foo“/>
  <owl:priorVersion
    rdf:resource=„http://ontoware.org/projects/swrc“/>
</owl:Ontology>
```

Weitere Primitive: owl:backwardCompatibleWith
owl:incompatibleWith owl:DeprecatedClass
owl:DeprecatedProperty

Klassen

<isweb>

```
<rdf:Description rdf:about=„Professor“>
  <rdf:type rdf:resource=„&owl:Class“/>
</rdf:description>

Oder

<owl:Class rdf:about=„Professor“/>

Und

<owl:Class rdf:about=„Professor“>
  <rdfs:subClassOf
    rdf:resource=„Dozent“/>
</owl:Class>
```

Mit

```
<owl:Class rdf:about=„Dozent“>
  <rdfs:subClassOf
    rdf:resource=„Person“/>
</owl:Class>
```

Folgt, dass Professor eine Person ist.

OWL Properties

<isweb>

```
<owl:ObjectProperty rdf:about=„liest“/>

<owl:DatatypeProperty rdf:about=„hatPersonalnummer“/>

<owl:ObjectProperty rdf:about=„liest“>
  <rdfs:domain rdf:resource=„Professor“/>
  <rdfs:range rdf:resource=„Vorlesung“/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about=„hatPersonalnummer“>
  <rdfs:domain rdf:resource=„Professor“/>
  <rdfs:range rdf:resource=„&xsd:positiveInteger“/>
</owl:DatatypeProperty>
```

XML Datentypen für OWL

<isweb>

```
Xsd:string xsd:float xsd:time xsd:gYear xsd:gMonth
xsd:anyURI xsd:language xsd:NCName xsd:long xsd:short
xsd:unsignedLong xsd:unsignedByte xsd:boolean
xsd:double xsd:date xsd:gMonthDay xsd:hexBinary
xsd:token xsd:NMTOKEN xsd:Name xsd:int xsd:byte
xsd:unsignedInt xsd:integer xsd:decimal xsd:dateTime
xsd:gYearMonth xsd:gDay xsd:base64Binary
xsd:normalizedString xsd:positiveInteger
xsd:nonPositiveInteger xsd:negativeInteger
xsd:nonNegativeInteger xsd:unsignedShort
```

Instanzen

<isweb>

```
<Professor rdf:about=„Sokrates“/>

Und

<Professor rdf:about=„Sokrates“>
  <liest rdf:resource=„Logik“/>
  <hatPersonalnummer rdf:datatype=„&xsd:positiveInteger“
    >2125</hatPersonalnummer>
</Professor>

Deswegen ist Sokrates auch eine Person
```

```
<rdf:Description rdf:about = „Sokrates“>
  <owl:sameAs rdf:resource=„Socrates“/>
</rdf:Description>
```

Das heisst: der Bezeichner für ein Individuum in RDF oder in OWL muss nicht der einzige Bezeichner für dieses Individuum sein.

```
<owl:Class rdf:about=„Professor“>
  <rdfs:subClassOf
    rdf:resource=„Dozent“/>
</owl:Class>
```

```
<owl:Class rdf:about=„Vorlesung“>
  <rdfs:subClassOf
    rdf:resource=„Veranstaltung“/>
</owl:Class>
```

```
<owl:Class rdf:about=„Dozent“>
  <rdfs:subClassOf
    rdf:resource=„Person“/>
</owl:Class>
```

```
<owl:Class rdf:about=„Person“>
  <owl:disjointWith
    rdf:resource=„Veranstaltung“/>
</owl:Class>
```

Auch Professor und Vorlesung
sind disjunkt

„Sokrates“ und „Grundzüge“
bezeichnen verschiedene Dinge

```
<owl:AllDifferent>
  <owl:distinctProfessors rdf:ParseType=„Collection“>
    <Person rdf:about=„Sokrates“/>
    <Person rdf:about=„Russel“/>
    <Person rdf:about=„Kopernikus“/>
    <Person rdf:about=„Popper“/>
    <Person rdf:about=„Augustinus“/>
    <Person rdf:about=„Curie“/>
    <Person rdf:about=„Kant“/>
  </owl:distinctMembers>
</owl:AllDifferent>
```

Was ist der Unterschied von „owl:AllDifferent“ und
„owl:disjointWith“?

```
<owl:Class rdf:about=„KoblenzerFachbereich“>
  <owl:oneOf rdf:parseType=„Collection“>
    <Fachbereich rdf:about=„FB1“/>
    <Fachbereich rdf:about=„FB2“/>
    <Fachbereich rdf:about=„FB3“/>
    <Fachbereich rdf:about=„FB4“/>
  </owl:oneOf>
</owl:AllDifferent>
```

Fügt man hinzu

```
<KoblenzerFachbereich rdf:about=„FB Informatik“>
```

dann folgt, dass einer der FB1..FB4 mit „FB Informatik“ identisch sein muss

```
<owl:Class rdf:about=„KoblenzerFachbereiche“>
  <owl:oneOf rdf:parseType=„Collection“>
    <Fachbereich rdf:about=„FB1“/>
    <Fachbereich rdf:about=„FB2“/>
    <Fachbereich rdf:about=„FB3“/>
    <Fachbereich rdf:about=„FB4“/>
  </owl:oneOf>
</owl:AllDifferent>
```

Fügt man (mittels owl:AllDifferent) hinzu, dass „Institut fuer Informatik“ disjunkt ist von FB1..FB4, dann folgt, dass das „Institut fuer Informatik“ kein Fachbereich ist

- UND – owl:intersectionOf
- ODER – owl:unionOf
- NICHT – owl:complementOf

```
<owl:Class rdf:about=„WeiblicheProfessor“>
  <owl:intersectionOf rdf:parseType=„Collection“>
    <owl:Class rdf:about=„Frau“/>
    <owl:Class rdf:about=„Professor“/>
  </owl:intersectionOf>
</owl:Class>
```

```
<owl:Class rdf:about=„WeiblicherProfessorIn“>
  <owl:intersectionOf rdf:parseType=„Collection“>
    <owl:Class rdf:about=„Frau“/>
    <owl:Class rdf:about=„Professor“/>
  </owl:intersectionOf>
</owl:Class>
```

```
<owl:Class rdf:about=„WeiblicherProfessorIn“>
  <owl:intersectionOf rdf:parseType=„Collection“>
    <owl:Class rdf:about=„Frau“/>
    <owl:Class rdf:about=„Professor“/>
  </owl:intersectionOf>
</owl:Class>
```

Ist Abkürzung von

```
<owl:Class rdf:about=„WeiblicherProfessorIn“>
  <owl:equivalentClass>
    <owl:intersectionOf rdf:parseType=„Collection“>
      <owl:Class rdf:about=„Frau“/>
      <owl:Class rdf:about=„Professor“/>
    </owl:intersectionOf>
  </owl:equivalentClass>
</owl:Class>
```

```
<owl:Class rdf:about=„Professor“>
  <rdfs:subClassOf>
    <owl:unionOf rdf:parseType=„Collection“>
      <owl:Class rdf:about=„aktivLehrend“/>
      <owl:Class rdf:about=„imRuhestand“/>
    </owl:unionOf>
  </rdfs:subClassOf>
</owl:Class>
```

D.h. jeder Professor ist entweder in der Klasse „aktivLehrend“ oder in der Klasse „imRuhestand“

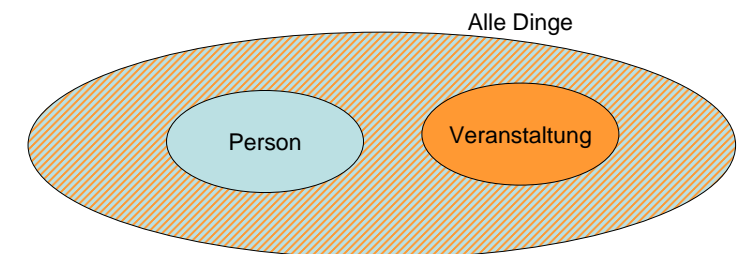
```
<owl:Class rdf:about=„Professor“>
  <rdfs:subClassOf>
    <owl:unionOf rdf:parseType=„Collection“>
      <owl:Class rdf:about=„aktivLehrend“/>
      <owl:Class rdf:about=„imRuhestand“/>
    </owl:unionOf>
  </rdfs:subClassOf>
</owl:Class>
```

Aber

```
<aktivLehrend rdf:about=„KurtLautenbach“>
<imRuhestand rdf:about=„KurtLautenbach“>
```

Ist erlaubt und möglich!

```
<owl:Class rdf:about=„Person“>
  <rdfs:subClassOf>
    <owl:complementOf rdf:resource=„Veranstaltung“/>
  </rdfs:subClassOf>
</owl:Class>
```



Schachtelungen beliebiger Tiefe

<is web>

```
<owl:Class rdf:about=„Professor“>
  <rdfs:subClassOf>
    <owl:unionOf rdf:parseType=„Collection“>
      <owl:intersectionOf rdf:parseType=„Collection“>
        <owl:Class rdf:about=„Person“/>
        <owl:Class
          rdf:about=„Universitaetsangehoeriger“/>
      </owl:intersectionOf>
    <owl:intersectionOf rdf:parseType=„Collection“>
      <owl:Class rdf:about=„Person“/>
      <owl:complementOf rdf:resource=„Doktorand“>
    </owl:intersectionOf>
  </owl:unionOf>
</rdfs:subClassOf>
</owl:Class>
```

Rolleneinschränkungen: allValuesFrom

<is web>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hatPruefer“/>
      <owl:allValuesFrom rdf:resource=„Professor“/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Jede Prüfung hat nur Prüfer, die Professoren sind.

Rolleneinschränkungen: someValuesFrom

<is web>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hatPruefer“/>
      <owl:someValuesFrom
        rdf:resource=„Professor“/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Jede Prüfung hat mindestens einen Prüfer, der Professor ist.

Rolleneinschränkungen: someValuesFrom

<is web>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hatPruefer“/>
      <owl:someValuesFrom
        rdf:resource=„Professor“/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Jede Prüfung hat mindestens einen Prüfer, der Professor ist.

Rolleneinschränkungen: someValuesFrom <isweb>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource=„hatPruefer“/>
      <owl:allValuesFrom
        rdf:resource=„Professor“/
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource=„hatPruefer“/>
      <owl:someValuesFrom
        rdf:resource=„Professor“/
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Jede Prüfung hat nur Professoren als Prüfer, aber davon mindestens einen!

Rolleneinschränkungen: Kardinalitäten <isweb>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource=„hatPruefer“/>
      <owl:maxCardinality
        rdf:datatype=
          „&xsd;nonNegativeInteger“
        4
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource=„hatPruefer“/>
      <owl:minCardinality
        rdf:datatype=
          „&xsd;nonNegativeInteger“
        2
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Maximal 4 Prüfer!

Mindestens 2 Prüfer!

Kardinalitäten <isweb>

```
<owl:Class rdf:about=„Pruefung“>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hatPruefer“/>
      <owl:cardinality
        rdf:datatype=„&xsd;nonNegativeInteger“
        3
    </owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Genau 3 Prüfer!

Werte als Rollenrestriktion <isweb>

```
<owl:Class rdf:about=„VorlesungBeiKant“>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„gelesenVon“/>
      <owl:hasValue rdf:resource=„Kant“/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

```
<owl:Class rdf:about=„VorlesungBeiKant“>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„gelesenVon“/>
      <owl:someValuesFrom>
        <owl:oneOf rdf:parseType=„Collection“>
          <owl:Thing rdf:about=„Kant“/>
        </owl:oneOf>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Gleichbedeutend mit der vorangegangenen Folie

```
<Vorlesung rdf:about=„Grundzüge“>
  <gelesenVon rdf:resource=„Kant“/>
</Vorlesung>
<Vorlesung rdf:about=„Die 3 Kritiken“/>
<Vorlesung rdf:about=„Ethik“/>
```

Open World Semantik!

Aus diesem Abschnitt und der vorhergehenden Folie folgt:
 <VorlesungBeiKant rdf:about=„Grundzüge“>

Es wird nicht gefolgert:

- dass die 3 Kritiken nicht von Kant gelesen werden und damit nicht zu VorlesungBeiKant gehört
- dass die Ethik nicht von Kant gelesen wird und damit nicht zu VorlesungBeiKant gehört
- dass die „Grundzüge“ nur von Kant gelesen werden

```
<owl:Class rdf:about=„Hacker“>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hoert“/>
      <owl:allValuesFrom rdf:resource=„InformatikVorlesung“/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Open World Semantik!

```
<Person rdf:about=„Schopenhauer“>
  <hoert><InformatikVorlesung rdf:resource=„Logik“/></hoert>
</Person>
<Person rdf:about=„Carnap“>
  <hoert><PhilosophieVorlesung rdf:resource=„Ethik“/></hoert>
</Person>
```

Schopenhauer ist deswegen noch kein Hacker
 Carnap kann kein Hacker sein

```
<owl:Class rdf:about=„FleissigerStudent“>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hoert“/>
      <owl:minCardinality rdf:datatype=„xsd:nonNegativeInteger“>3</owl:minCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

```
<Person rdf:about=„Schopenhauer“>
  <hoert><InformatikVorlesung rdf:resource=„Logik“/></hoert>
</Person>
<Person rdf:about=„Carnap“>
  <hoert><Vorlesung rdf:resource=„Grundzüge“/></hoert>
  <hoert><InformatikVorlesung rdf:resource=„Logik“/></hoert>
  <hoert><PhilosophieVorlesung rdf:resource=„Ethik“/></hoert>
</Person>
```

Schopenhauer ist deswegen noch kein unfleissiger Student
 Carnap ist jetzt fleissiger Student


```

<owl:Class rdf:about=„FaulerStudentOderDiplomarbeiter“>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=„hoert“/>
      <owl:maxCardinality
        rdf:datatype=„&xsd:nonNegativeInteger“>
        0
      </owl:maxCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>

<Person rdf:about=„Schopenhauer“>
  <hoert><InformatikVorlesung rdf:resource=„Logik“/></hoert>
</Person>
<Person rdf:about=„Aristoxenos“/>
</Person>
  
```

} ?
 } ?

```

<owl:ObjectProperty rdf:about=„hatPruefer“>
  <rdfs:subPropertyOf rdf:resource=„haAnwesenden“/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about=„hatPruefer“>
  <owl:inverseOf rdf:resource=„prueferVon“>
</owl:ObjectProperty>

<Pruefung rdf:about=„EthikPruefung“>
  <hatPruefer rdf:resource=„Sokrates“/>
</Pruefung>
  
```

<ul style="list-style-type: none"> ▪ Transitivität <pre> <owl:ObjectProperty rdf:about=„hatKollegen“> <rdf:type rdf:resource=„&owl;TransitiveProperty“/> </owl:ObjectProperty> </pre>	<pre> <Person rdf:about=„Kant“> <hatKollegen rdf:resource=„Sokrates“/> <hatKollegen rdf:resource=„Russel“/> <hatKollegen rdf:resource=„Popper“/> <liest rdf:resource=„Grundzüge“/> </Person> </pre>
<ul style="list-style-type: none"> ▪ Symmetrie <pre> <owl:ObjectProperty rdf:about=„hatKollegen“> <rdf:type rdf:resource=„&owl;SymmetricProperty“/> </owl:ObjectProperty> </pre>	<pre> <Vorlesung rdf:about=„Ethik“> <gelesenVon rdf:resource=„Sokrates“/> <gelesenVon rdf:resource=„Socrates“/> </Vorlesung> </pre>
<ul style="list-style-type: none"> ▪ Funktionalität <pre> <owl:ObjectProperty rdf:about=„gelesenVon“> <rdf:type rdf:resource=„&owl;FunctionalProperty“/> </owl:ObjectProperty> </pre>	<pre> <owl:ObjectProperty rdf:about=„liest“> <rdf:type rdf:resource=„&owl;InverseFunctionalProperty“/> </owl:ObjectProperty> </pre>