



TECHNISCHE
UNIVERSITÄT
DRESDEN

FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

OWL – Syntax & Intuition

Sebastian Rudolph

Dresden, 26 April 2013

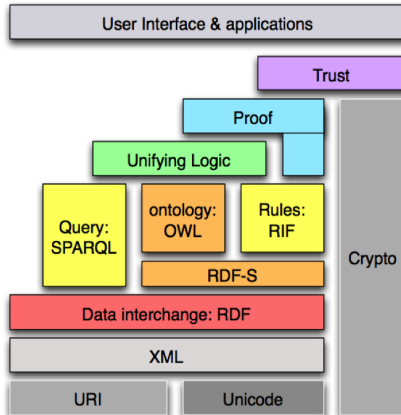


DRESDEN
SCHOOL
OF INFORMATICS
SEBASTIAN RUDOLPH

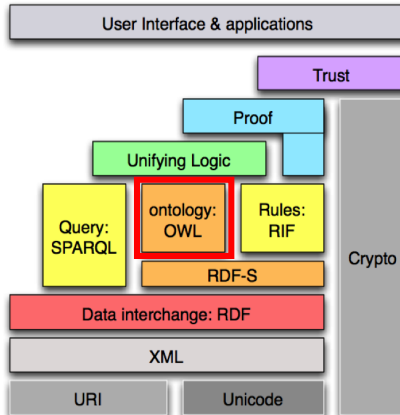
Content

Overview & XML	9 APR DS2	Hypertableau II	7 JUN DS5
Introduction into RDF	9 APR DS3	Tutorial 5	11 JUN DS2
RDFS – Syntax & Intuition	12 APR DS5	SPARQL Syntax & Intuition	11 JUN DS3
RDFS – Semantics	16 APR DS2	SPARQL – Semantics	14 JUN DS2
RDFS Rule-based Reasoning	16 APR DS3	SPARQL 1.1	14 JUN DS3
Tutorial 1	19 APR DS5	SPARQL Entailment	14 JUN DS5
OWL – Syntax & Intuition	26 APR DS5	Tutorial 6	18 JUN DS2
Tutorial 2	3 MAY DS5	SPARQL Implementation	18 JUN DS3
OWL & Description Logics	7 MAY DS2	Ontology Editing	2 JUL DS2
OWL 2	7 MAY DS3	Ontology Engineering	2 JUL DS3
Tutorial 3	10 MAY DS5	Tutorial 7	9 JUL DS2
Tableau I	14 MAY DS2	Linked Data	9 JUL DS3
Tableau II	14 MAY DS3	Applications	12 JUL DS5
Tutorial 4	17 MAY DS5	Test Exam	16 JUL DS2
Tableau – Optimizations	7 JUN DS2	Test Exam Evaluation	16 JUL DS3
Hypertableau I	7 JUN DS3	Q&A Session	19 JUL DS5

OWL



OWL



Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Ontology in Philosophy

- notion exists only in singular (no “ontologies”)
- denotes the “study of being”
- can be found in philosophical writings of Aristotle (Socrates), Thomas Aquinas, Descartes, Kant, Hegel, Wittgenstein, Heidegger, Quine, ...
- term first mentioned in 17th century

Ontology in Computer Science

Gruber (1993):

“An Ontology is a

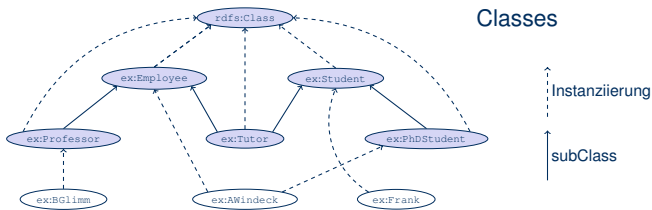
- | | |
|--------------------------|------------------------------|
| formal specification | ⇒ interpretable by machines |
| of a shared | ⇒ based on consensus |
| conceptualization | ⇒ describes relevant notions |
| of a domain of interest” | ⇒ referring to a “topic” |

Ontologies in Practice

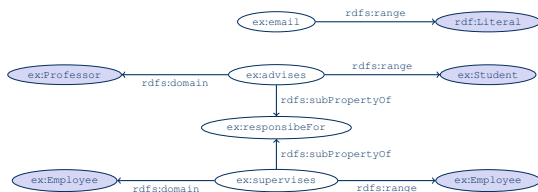
Some Requirements

- instantiation of classes by individuals
- conceptual hierarchies (taxonomies, “inheritance”):
classes, concepts
- binary relations between individuals: properties, roles
- characteristics of relations (z.B. range, transitive)
- datatypes (e.g. numbers): concrete domains
- logical operators
- clear semantics

RDFS – Simple Ontologies



Relations



RDF Schema as Ontology Language?

- appropriate for simple ontologies
- advantage: automated inferencing relatively efficient
- but: not appropriate for more complex modeling
- resort to more expressive languages, like
 - OWL
 - RIF ...

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

OWL – General Remarks

- W3C Recommendation since 2004
- semantic fragment of FOL
- three variants:
 - OWL Lite
 - OWL DL
 - OWL Full
- no reification in OWL DL
- ↔ RDFS is fragment of OWL Full
- OWL DL is decidable
corresponds to description logic SHOIN(D)
- W3C documents contain details that cannot all be covered here

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools
- OWL DL
 - contains OWL Lite and is sublanguage of OWL Full
 - widely supported by tools
 - complexity NExpTime (worst-case)

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools
- OWL DL
 - contains OWL Lite and is sublanguage of OWL Full
 - widely supported by tools
 - complexity NExpTime (worst-case)
- OWL Lite
 - sublanguage of OWL DL and OWL Full
 - low expressivity
 - complexity ExpTime (worst-case)

OWL Documents

- are RDF documents
(at least in the standard syntax; there are others)
- consist of
 - head with general information
 - rest with actual ontology

Head of an OWL Document

definition of name spaces in the root

```
<rdf:RDF
  xmlns="http://example.org/exampleontology#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  ...
</rdf:RDF>
```

Head of an OWL Document

general information

```
<owl:Ontology rdf:about="">
  <rdfs:comment
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    SWRC ontology, version of June 2007
  </rdfs:comment>
  <owl:versionInfo>v0.7.1</owl:versionInfo>
  <owl:imports rdf:resource="http://www.example.org/foo" />
  <owl:priorVersion
    rdf:resource="http://ontoware.org/projects/swrc" />
</owl:Ontology>
```

Head of an OWL Document

taken from RDFS

`rdfs:comment`
`rdfs:label`
`rdfs:seeAlso`
`rdfs:isDefinedBy`

in addition

`owl:imports`

for versioning

`owl:versionInfo`
`owl:priorVersion`
`owl:backwardCompatibleWith`
`owl:incompatibleWith`
`owl:DeprecatedClass`
`owl:DeprecatedProperty`

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Classes, Roles and Individuals

three building blocks of ontology axioms

- classes
 - comparable with classes in RDFS
- individuals
 - comparable with “proper” instances in RDFS
- roles
 - comparable with properties in RDFS

classes

definition

- `<owl:Class rdf:about = "Professor" />`
- equivalent to

```
<rdf:Description rdf:about="#Professor">  
  <rdf:type  
    rdf:resource="http://www.w3.org/2002/07/owl#Class"/>  
</rdf:Description>
```

pre-defined

- `owl:Thing`
- `owl:Nothing`

Individuals

definition via class membership

```
<rdf:Description rdf:about="rudiStuder">  
  <rdf:type rdf:resource="#Professor"/>  
</rdf:Description>
```

equivalent:

```
<Professor rdf:about="rudiStuder"/>
```


Abstrakte Rollen

abstract roles are defined in a way similar to classes

```
<owl:ObjectProperty rdf:about="hasAffiliation" />
```

domain and range of abstract roles

```
<owl:ObjectProperty rdf:about="hasAffiliation">  
  <rdfs:domain rdf:resource="Person" />  
  <rdfs:range rdf:resource="Organization" />  
</owl:ObjectProperty>
```

Concrete Roles

concrete roles have datatypes as range

```
<owl:DatatypeProperty rdf:about="firstName" />
```

domain and range of concrete roles

```
<owl:DatatypeProperty rdf:about="firstName">  
  <rdfs:domain rdf:resource="Person" />  
  <rdfs:range rdf:resource="&xsd:string" />  
</owl:DatatypeProperty>
```

many XML datatypes can be used

Individuals and Roles

```
<Person rdf:about="rudiStuder">
  <hasAffiliation rdf:resource="aifb" />
  <hasAffiliation rdf:resource="fzi" />
  <firstName rdf:datatype="&xsd:string">
    Rudi
  </firstName>
</Person>
```

in general roles are not functional

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- **Class Relationships**
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Simple Class Relationships

```
<owl:Class rdf:about="Professor">  
  <rdfs:subClassOf rdf:resource="FacultyMember" />  
</owl:Class>  
<owl:Class rdf:about="FacultyMember">  
  <rdfs:subClassOf rdf:resource="Person" />  
</owl:Class>
```

it follows by inference that `Professor` is a subclass of `Person`

Simple Class Relationships

```
<owl:Class rdf:about="Professor">
  <rdfs:subClassOf rdf:resource="FacultyMember" />
</owl:Class>
<owl:Class rdf:about="Book">
  <rdfs:subClassOf rdf:resource="Publication" />
</owl:Class>
<owl:Class rdf:about="FacultyMember">
  <owl:disjointWith rdf:resource="Publication" />
</owl:Class>
```

it follows by inference that `Professor` and `Buch` are also disjoint classes

Simple Class Relationships

```
<owl:Class rdf:about="Man">  
  <rdfs:subClassOf rdf:resource="Person" />  
</owl:Class>  
<owl:Class rdf:about="Person">  
  <owl:equivalentClass rdf:resource="Human" />  
</owl:Class>
```

it follows by inference that `Man` is a subclass of `Human`

Individuals and Class Relationships

```
<Book rdf:about="http://semantic-web-book.org/uri">  
<author rdf:resource="pascalHitzler" />  
<author rdf:resource="markusKroetzsch" />  
<author rdf:resource="sebastianRudolph" />  
</Book>  
<owl:Class rdf:about="Book">  
<rdfs:subClassOf rdf:resource="Publication" />  
</owl:Class>
```

it follows by inference that Foundations of Semantic Web Technologies **is a** Publication.

Relationships between Individuals

```
<Professor rdf:about="rudiStuder" />  
  <rdf:Description rdf:about="rudiStuder">  
    <owl:sameAs rdf:resource="professorStuder" />  
  </rdf:Description>
```

it follows by inference that `rudiStuder` is a `Professor`
distinctness of individuals expressed via `owl:differentFrom`.

Relationships between Individuals

```
<owl:AllDifferent>  
<owl:distinctMembers rdf:parseType="Collection">  
<Person rdf:about="rudiStuder" />  
<Person rdf:about="dennyVrandecic" />  
<Person rdf:about="peterHaase" />  
</owl:distinctMembers>  
</owl:AllDifferent>
```

abbreviated notation instead of using several `owl:differentFrom`

usage of `owl:AllDifferent` and `owl:distinctMembers` exclusively for this purpose

Closed Classes

```
<owl:Class rdf:about="SecretariesOfStuder">
  <owl:oneOf rdf:parseType="Collection">
    <Person rdf:about="giselaSchillinger" />
    <Person rdf:about="anneEberhardt" />
  </owl:oneOf>
</owl:Class>
```

tells that there are only exactly these two `SecretariesOfStuder`

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Logical Class Constructors

- **logical and (conjunction):**
`owl:intersectionOf`
- **logical or (disjunction):**
`owl:unionOf`
- **logical not (negation):**
`owl:complementOf`
- **used to construct complex classes from simple classes**

Conjunktion

```
<owl:Class rdf:about="SecretariesOfStuder">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="Secretaries" />
    <owl:Class rdf:about="MembersOfStudersGroup" />
  </owl:intersectionOf>
</owl:Class>
```

it follows by inference that all `SecretariesOfStuder` **are also** `Secretaries`

Disjunction

```
<owl:Class rdf:about="Professor">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="ActivelyTeaching" />
        <owl:Class rdf:about="Retired" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

Negation

```
<owl:Class rdf:about="FacultyMember">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf rdf:resource="Publication" />
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

semantically equivalent:

```
<owl:Class rdf:about="FacultyMember">
  <owl:disjointWith rdf:resource="Publication" />
</owl:Class>
```


Role Restrictions (allValuesFrom)

used to define complex classes via roles

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:allValuesFrom rdf:resource="Professor" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

i.e., all examiners of an exam have to be professors

Role Restrictions (someValuesFrom)

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:someValuesFrom rdf:resource="Person" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

i.e., every exam must have at least one examiner

Role Restrictions (Cardinalities)

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer"/>
      <owl:maxCardinality
        rdf:datatype="&xsd;nonNegativeInteger">
        2
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

an exam may have at most two examiners

Role Restrictions (Cardinalities)

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasTopic"/>
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">3
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

an exam must cover at least three topics

Role Restrictions (Cardinalities)

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasTopic"/>
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">3
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

an exam must cover exactly three topics

Role Restrictions (hasValue)

```
<owl:Class rdf:about="ExamStuder">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:hasValue rdf:resource="rudiStuder" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

`owl:hasValue` **always** refers to one singular individual
the above is equivalent to the example on the next slide

Role Restrictions (hasValue)

```
<owl:Class rdf:about="ExamStuder">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:someValuesFrom>
        <owl:oneOf rdf:parseType="Collection">
          <owl:Thing rdf:about="rudiStuder" />
        </owl:oneOf>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Role Relationships

```
<owl:ObjectProperty rdf:about="hasExaminer">  
  <rdfs:subPropertyOf rdf:resource="hasParticipant" />  
</owl:ObjectProperty>
```

likewise: owl:equivalentProperty
roles can be inverses of each other:

```
<owl:ObjectProperty rdf:about="hasExaminer">  
  <owl:inverseOf rdf:resource="examinerOf"/>  
</owl:ObjectProperty>
```

Role Characteristics

- domain
- range
- transitivity, i.e.
 $r(a, b)$ and $r(b, c)$ imply $r(a, c)$
- symmetry, i.e.
 $r(a, b)$ implies $r(b, a)$
- functionality
 $r(a, b)$ and $r(a, c)$ imply $b = c$
- inverse functionality
 $r(a, b)$ and $r(c, b)$ imply $a = c$

Domain and Range

```
<owl:ObjectProperty rdf:about="isMemberOf">  
  <rdfs:range rdf:resource="Organization" />  
</owl:ObjectProperty>
```

equivalent to:

```
<owl:Class rdf:about="&owl;Thing">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="isMemberOf" />  
      <owl:allValuesFrom rdf:resource="Organization" />  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

Domain and Range: Caution!

```
<owl:ObjectProperty rdf:about="isMemberOf">
  <rdfs:range rdf:resource="Organization" />
</owl:ObjectProperty>
<number rdf:about="five">
  <isMemberOf rdf:resource="PrimeNumbers" />
</number>
```

it follows that `PrimeNumbers` are an `Organization`!

Role Characteristics

```
<owl:ObjectProperty rdf:about="hasColleague">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="hasProjectLeader">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="isProjectLeaderFor">
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty" />
</owl:ObjectProperty>
<Person rdf:about="peterHaase">
  <hasColleague rdf:resource="philippCimiano" />
  <hasColleague rdf:resource="steffenLamparter" />
  <isProjectLeaderFor rdf:resource="neOn" />
</Person>
<Project rdf:about="x-Media">
  <hasProjectLeader rdf:resource="philippCimiano" />
  <hasProjectLeader rdf:resource="cimianoPhilipp" />
</Project>
```

Consequences from the Example

- `steffenLamparter hasColleague peterHaase`
- `steffenLamparter hasColleague philippCimiano`
- `philippCimiano owl:sameAs cimianoPhilipp`

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- [OWL Variants](#)
- OWL Ontologies: Reasoning Tasks

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools
- OWL DL
 - contains OWL Lite and is sublanguage of OWL Full
 - widely supported by tools
 - complexity NExpTime (worst-case)

OWL 1 Variants

- OWL Full
 - contains OWL DL and OWL Lite
 - contains all of RDFS (as the only OWL variant)
 - semantics contains some aspects that are problematic from a logical perspective
 - undecidable
 - limited support by tools
- OWL DL
 - contains OWL Lite and is sublanguage of OWL Full
 - widely supported by tools
 - complexity NExpTime (worst-case)
- OWL Lite
 - sublanguage of OWL DL and OWL Full
 - low expressivity
 - complexity ExpTime (worst-case)

OWL Full

- unrestricted use of all OWL and RDFS language elements (has to be valid RDFS)
- difficult e.g.: non-existent type separation (classes, roles, individuals), thus:
 - `owl:Thing` becomes the same as `rdfs:resource`
 - `owl:Class` becomes the same as `rdfs:Class`
 - `owl:DatatypeProperty` becomes a subclass of `owl:ObjectProperty`
 - `owl:ObjectProperty` becomes the same as `rdf:Property`

Example for Confusion of Types in OWL Full

```
<owl:Class rdf:about="Book">  
  <germanName rdf:datatype="&xsd:string">Buch</germanName>  
  <frenchName rdf:datatype="&xsd:string">livre</frenchName>  
</owl:Class>
```

inferences about such constructs are rarely needed in practice

OWL DL

- only usage of RDFS language elements that are explicitly allowed (like those in our examples)
not allowed: `rdfs:Class`, `rdfs:Property`
- type separation: classes and roles have to be explicitly declared
- concrete roles must not be specified as transitive, symmetric, inverse or inverse functional
- number restrictions must not be used with transitive roles, their subroles, or inverses thereof

OWL Lite

- all restrictions of OWL DL
- moreover:
 - **not allowed:** `oneOf`, `unionOf`, `complementOf`, `hasValue`, `disjointWith`
 - number restrictions only allowed with 0 and 1
 - some constraints referring to anonymous (complex) classes, e.g., only in the subject of `rdfs:subClassOf`

Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks

Terminological Queries to OWL Ontologies

- class equivalence
- subclass relationships
- disjointness of classes
- global consistency (aka satisfiability)
- class consistency: a class is inconsistent if it is equivalent to `owl:Nothing` – this hints to a modeling error:

```
<owl:Class rdf:about="Book">  
  <owl:subClassOf rdf:resource="Publication"/>  
  <owl:disjointWith rdf:resource="Publication"/>  
</owl:Class>
```


Assertionale Queries to OWL Ontologies

- instance checking: does a given individual belong to a given class?
- search for all individuals that are members of a given class
- are two given individuals linked by a role?
- search for all individual pairs that are linked by a certain role
- ... caution: only “provable” answers will be given!

OWL 1 Language Elements

head

- `rdfs:comment`
- `rdfs:label`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`
- `owl:versionInfo`
- `owl:priorVersion`
- `owl:backwardCompatibleWith`
- `owl:incompatibleWith`
- `owl:DeprecatedClass`
- `owl:DeprecatedProperty`
- `owl:imports`

relationships between individuals

- `owl:sameAs`
- `owl:differentFrom`
- `owl:AllDifferent`
- `owl:distinctMembers`

pre-defined datatypes (OWL 1)

- `xsd:string`
- `xsd:integer`

OWL Language Elements

class constructors and -relationships

- owl:Class
- owl:Thing
- owl:Nothing
- rdfs:subClassOf
- owl:disjointWith
- owl:equivalentClass
- owl:intersectionOf
- owl:unionOf
- owl:complementOf

role restrictions

- owl:allValuesFrom
- owl:someValuesFrom
- owl:hasValue
- owl:cardinality
- owl:minCardinality
- owl:maxCardinality
- owl:oneOf

OWL Language Elements

role constructors, relationships and characteristics

- `owl:ObjectProperty`
- `owl:DatatypeProperty`
- `rdfs:subPropertyOf`
- `owl:equivalentProperty`
- `owl:inverseOf`
- `rdfs:domain`
- `rdfs:range`
- `owl:TransitiveProperty`
- `owl:SymmetricProperty`
- `owl:FunctionalProperty`
- `owl:InverseFunctionalProperty`

Further Literature

- <http://www.w3.org/2004/OWL/>
central W3C web page for OWL
- <http://www.w3.org/TR/owl-features/>
overview over OWL
- <http://www.w3.org/TR/owl-ref/>
comprehensive description of the OWL language components
- <http://www.w3.org/TR/owl-guide/>
introduction into OWL knowledge modeling
- <http://www.w3.org/TR/owl-semantics/>
describes the semantics of OWL and the abstract syntax for OWL DL (→ later lecture)