

11. Übung „Künstliche Intelligenz“

Wintersemester 2007/2008

RDF und OWL

1. Betrachten Sie die folgende RDF-Repräsentation:

```
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:NS0='http://iswww.edu/test1#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'>
  <rdf:Description rdf:about='http://iswww.edu/test1#roter_doppeldecker'>
    <rdf:type rdf:resource='http://iswww.edu/test1#flugzeug' />
  </rdf:Description>
  <rdf:Description rdf:about='http://iswww.edu/test1#fliegen'>
    <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
    <rdfs:domain rdf:resource='http://iswww.edu/test1#pilot' />
    <rdfs:range rdf:resource='http://iswww.edu/test1#flugzeug' />
  </rdf:Description>
  <rdf:Description rdf:about='http://iswww.edu/test1#flugzeug'>
    <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
    <rdfs:label xml:lang='en'>http://iswww.edu/test1#Flugzeug</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about='http://iswww.edu/test1#rote_baron'>
    <rdfs:label xml:lang='en'>http://iswww.edu/test1#Rote Baron</rdfs:label>
    <rdf:type rdf:resource='http://iswww.edu/test1#pilot' />
    <NS0:fliegen rdf:resource='http://iswww.edu/test1#roter_doppeldecker' />
  </rdf:Description>
  <rdf:Description rdf:about='http://iswww.edu/test1#pilot'>
    <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
    <rdfs:label xml:lang='en'>http://iswww.edu/test1#Pilot</rdfs:label>
  </rdf:Description>
</rdf:RDF>
```

- a) Beschreiben Sie in natürlicher Sprache, was durch diese Repräsentation ausgedrückt wird.

```

<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:NS0='http://iswww.edu/test1#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'>

<rdf:Description rdf:about='http://iswww.edu/test1#roter_doppeldecker'>
  <rdf:type rdf:resource='http://iswww.edu/test1#flugzeug' />
</rdf:Description>

```

Es gibt einen roten Doppeldecker, der ein Flugzeug ist.

```

<rdf:Description rdf:about='http://iswww.edu/test1#fliegen'>
  <rdf:type rdf:resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property' />
  <rdfs:domain rdf:resource='http://iswww.edu/test1#pilot' />
  <rdfs:range rdf:resource='http://iswww.edu/test1#flugzeug' />
</rdf:Description>

```

Nur Piloten fliegen und sie fliegen ausschließlich Flugzeuge.

```

<rdf:Description rdf:about='http://iswww.edu/test1#flugzeug'>
  <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
  <rdfs:label xml:lang='en'>http://iswww.edu/test1#Flugzeug</rdfs:label>
</rdf:Description>

```

„Flugzeug“ ist ein allgemeines Konzept.

```

<rdf:Description rdf:about='http://iswww.edu/test1#rote_baron'>
  <rdfs:label xml:lang='en'>http://iswww.edu/test1#Rote Baron</rdfs:label>
  <rdf:type rdf:resource='http://iswww.edu/test1#pilot' />
  <NS0:fliegen rdf:resource='http://iswww.edu/test1#roter_doppeldecker' />
</rdf:Description>

```

Der Rote Baron ist Pilot und fliegt den oben beschriebenen roten Doppeldecker.

```

<rdf:Description rdf:about='http://iswww.edu/test1#pilot'>
  <rdf:type rdf:resource='http://www.w3.org/2000/01/rdf-schema#Class' />
  <rdfs:label xml:lang='en'>http://iswww.edu/test1#Pilot</rdfs:label>
</rdf:Description>
</rdf:RDF>

```

„Pilot“ ist ein allgemeines Konzept.

- b) Erstellen Sie eine graphische Darstellung der obigen RDF-Repräsentation.

2. Wie würden Sie folgende Aussage reifizieren? Um diese Aufgabe zu lösen, können Sie die jeweiligen Entitäten mit IDs versehen.

Richard Stallman sagt: „GNU/Linux ist ein gutes Betriebssystem“. Bill Gates widerspricht seiner Aussage.

```
<rdf:Statement rdf:ID="RichardStallmansStatement">
  <rdf:subject rdf:resource="#GNUlinux"/>
  <rdf:predicate rdf:resource="#isGood"/>
  <rdf:object rdf:resource="#OperatingSystem"/>
</rdf:Statement>

<rdf:Description rdf:about="#RichardStallmansStatement">
  <mySchema:saidBy rdf:resource="#RichardStallman">
  <mySchema:contradictedBy rdf:resource="#BillGates">
</rdf:Description>
```

3. Übersetzen Sie die folgenden OWL-Statements in die Beschreibungslogik-Notation!

```
1 <owl:Class rdf:ID="Book">
2   <rdfs:subClassOf rdf:resource="#Publication"/>
3 </owl:Class>
4 <owl:Class rdf:ID="Professor"/>
5 <owl:Class rdf:ID="Author">
6   <rdfs:subClassOf rdf:resource="#Person"/>
7 </owl:Class>
8 <owl:Class rdf:ID="#Classbook">
9   <owl:equivalentClass>
10    <owl:intersectionOf rdf:parseType="Collection">
11      <owl:Class rdf:about="#Book"/>
12      <owl:Class rdf:about="#LectureMaterial"/>
13    </owl:intersectionOf>
14  </owl:equivalentClass>
15 </owl:Class>
16 <rdf:Description ID="hasAuthor">
17   <rdf:type resource="http://www.w3.org/...#Property"/>
18   <rdfs:domain rdf:resource="#Publication"/>
19   <rdfs:range rdf:resource="#Author"/>
20 </rdf:Description>
21 <ClassBook rdf:ID="AnalysisI">
22   <hasAuthor rdf:resource="#KurtEndl"/>
23   <hasAuthor rdf:resource="#WolfgangLuh"/>
24 </ClassBook>
```

```
25 <owl:Class rdf:about="#Classbook">
26   <rdfs:subClassOf>
27     <owl:Restriction>
28       <owl:onProperty rdf:resource="#hasAuthor"/>
29       <owl:allValuesFrom rdf:resource="#Professor"/>
30     </owl:Restriction>
31   </rdfs:subClassOf>
32 </owl:Class>
```

Hier die Beschreibungslogik-Notation und die dazugehörigen OWL Bausteine:

a) $Book \sqsubseteq Publication$

```
1 <owl:Class rdf:ID="Book">
2   <rdfs:subClassOf rdf:resource="#Publication"/>
3 </owl:Class>
```

b) Professor ist eine primitive Klasse

```
4 <owl:Class rdf:ID="Professor"/>
```

c) $Author \sqsubseteq Person$

```
5 <owl:Class rdf:ID="Author">
6   <rdfs:subClassOf rdf:resource="#Person"/>
7 </owl:Class>
```

d) $Classbook \equiv Book \sqcap LectureMaterial$

```
8 <owl:Class rdf:ID="#Classbook">
9   <owl:equivalentClass>
10     <owl:intersectionOf rdf:parseType="Collection">
11       <owl:Class rdf:about="#Book"/>
12       <owl:Class rdf:about="#LectureMaterial"/>
13     </owl:intersectionOf>
14   </owl:equivalentClass>
15 </owl:Class>
```

e) i. $\exists hasAuthor.T \sqsubseteq Publication$

ii. $T \sqsubseteq \forall hasAuthor.Author$

```
16 <rdf:Description ID="hasAuthor">
17   <rdf:type resource="http://www.w3.org/...#Property"/>
18   <rdfs:domain rdf:resource="#Publication"/>
19   <rdfs:range rdf:resource="#Author"/>
20 </rdf:Description>
```

f) i. $Classbook(AnalysisI)$

ii. $hasAuthor(AnalysisI, KurtEndl)$

iii. $hasAuthor(AnalysisI, WolfgangLuh)$

```
21 <ClassBook rdf:ID="AnalysisI">
22   <hasAuthor rdf:resource="#KurtEndl"/>
23   <hasAuthor rdf:resource="#WolfgangLuh"/>
24 </ClassBook>
```

g) $Classbook \sqsubseteq \forall hasAuthor.Professor$

```
25 <owl:Class rdf:about="#Classbook">
26   <rdfs:subClassOf>
27     <owl:Restriction>
28       <owl:onProperty rdf:resource="#hasAuthor"/>
29       <owl:allValuesFrom rdf:resource="#Professor"/>
30     </owl:Restriction>
31   </rdfs:subClassOf>
32 </owl:Class>
```

4. Welche logischen Schlüsse können Sie aus dem vorliegenden Wissen ziehen?

Es ergeben sich folgende Schlüsse:

- Aus 3(e)ii sowie 3(f)ii folgt *Author(KurtEndl)* und aus damit wegen 3c auch *Person(KurtEndl)*. Ebenso folgt aus 3(e)ii und 3(f)iii *Author(WolfgangLuh)* und damit wegen 3c auch *Person(WolfgangLuh)*.
- Aus 3a und 3d folgt *Classbook* \sqsubseteq *Publication*
- Aus 3(f)i, 3(f)ii und 3g folgt *Professor(KurtEndl)*. Ebenso folgt aus 3(f)i, 3(f)iii und 3g *Professor(WolfgangLuh)*.
- Aus 3d und 3(f)i folgt *Book(AnalysisI)* sowie *LectureMaterial(AnalysisI)*.
- Aus (3(e)i und 3(f)ii) oder (3(e)i und 3(f)iii) oder (3(f)i, 3d und 3a) folgt *Publication(AnalysisI)*.