

Semantic Web Grundlagen

Lösung zur Übung 3: OWL

Birte Glimm

WS 2011/2012

Lösung (3.1).

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
]>
<rdf:RDF
  xmlns="http://example.org/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <owl:Class rdf:about="Gemüse">
    <rdfs:subClassOf rdf:resource="PizzaBelag"/>
  </owl:Class>

  <owl:Class rdf:about="Pizza">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="hatBelag"/>
        <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
          2
        </owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="PizzaBelag"/>
  </owl:Class>

  <owl:NamedIndividual rdf:about="Aubergine">
    <rdf:type rdf:resource="Gemüse"/>
  </owl:NamedIndividual>

```

oder

```

<Gemüse rdf:about="Aubergine"/>

<owl:ObjectProperty rdf:about="hatBelag">
  <rdfs:domain rdf:resource="Pizza"/>
  <rdfs:range rdf:resource="PizzaBelag"/>
</owl:ObjectProperty>

<owl:Class rdf:about = "PizzaMargaritta">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hatBelag"/>
      <owl:someValuesFrom rdf:resource="Tomate"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="hatBelag"/>
          <owl:someValuesFrom rdf:resource="Fleisch"/>
        </owl:Restriction>
      </owl:complementOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="VegetarischePizza">
  <rdfs:subClassOf rdf:resource="PizzaOhneFisch"/>
  <rdfs:subClassOf rdf:resource="PizzaOhneFleisch"/>
</owl:Class>
oder
<owl:Class rdf:about="VegetarischePizza">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:resource="PizzaOhneFisch"/>
    <owl:Class rdf:resource="PizzaOhneFleisch"/>
  </owl:intersectionOf>
</owl:Class>

</rdf:RDF>

```

In Beschreibungslogik Syntax:

$$\begin{aligned} \text{Gemüse} &\sqsubseteq \text{PizzaBelag} \\ \text{Pizza} &\sqsubseteq \geq 2 \text{ hatBelag. } \top \\ \text{Pizza} \sqcap \text{PizzaBelag} &\sqsubseteq \perp \end{aligned}$$

weitere Variante

$$\begin{aligned} \text{Pizza} &\sqsubseteq \neg \text{PizzaBelag} \\ &\quad \text{Gemüse(Aubergine)} \\ &\quad \top \sqsubseteq \forall \text{hatBelag. PizzaBelag} \\ \exists \text{hatBelag. } \top &\sqsubseteq \text{Pizza} \\ \text{PizzaMargaritta} &\sqsubseteq \exists \text{hatBelag. Tomate} \end{aligned}$$

oder falls Tomate als Individuum modelliert wird

$$\begin{aligned} \text{PizzaMargaritta} &\sqsubseteq \exists \text{hatBelag. } \{ \text{Tomate} \} \\ \text{PizzaMargaritta} &\sqsubseteq \neg \exists \text{hatBelag. Fleisch} \\ \text{VegetarischePizza} &\sqsubseteq \text{PizzaOhneFisch} \\ \text{VegetarischePizza} &\sqsubseteq \text{PizzaOhneFleisch} \end{aligned}$$

Lösung (3.2).

- Die Rolle hatZutat ist transitiv. ✓
- Die Rolle hatBelag ist funktional. ✗
- Die Rolle hatBelag ist invers funktional. ✓
- Die Rolle hatZutat ist asymmetrisch. ✓

Lösung (3.3).

(a) Jeder, der ehrlich ist und ein Verbrechen verübt hat, zeigt sich selbst an.

```
<owl:Class>
  <owl:intersectionOf rdf:parseType = "Collection">
    <owl:Class rdf:resource = "Ehrlich" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="verübt" />
```

```

        <owl:someValuesFrom rdf:resource="Verbrechen"/>
    </owl:Restriction>
</owl:intersectionOf>
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="anzeigen"/>
        <owl:hasSelf rdf:datatype="&xsd:boolean">true</owl:hasSelf>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

In BL Syntax: Ehrlich \sqcap \exists verübt.Verbrechen \sqsubseteq \exists anzeigen.Self

(b) Wer klug und ehrlich ist, verübt kein Verbrechen.

```

<owl:Class>
    <owl:intersectionOf rdf:parseType = "Collection">
        <owl:Class rdf:resource = "Ehrlich" />
        <owl:Class rdf:resource = "Klug" />
    </owl:intersectionOf>
<rdfs:subClassOf>
    <owl:complementOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="verübt"/>
            <owl:someValuesFrom rdf:resource="Verbrechen"/>
        </owl:Restriction>
    </owl:complementOf>
</rdfs:subClassOf>
</owl:Class>

```

In BL Syntax: Ehrlich \sqcap Klug \sqsubseteq $\neg\exists$ verübt.Verbrechen

In NNF: Ehrlich \sqcap Klug \sqsubseteq \forall verübt.(\neg Verbrechen)

(c) Bonnie zeigt Clyde nicht an.

```

<owl:Class>
    <owl:oneOf rdf:parseType="Collection">
        <rdf:Description rdf:about="Bonnie"/>
    </owl:oneOf>
<rdfs:subClassOf>
    <owl:Class>
        <owl:complementOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="anzeigen"/>
                <owl:someValuesFrom>
                    <owl:Class>
                        <owl:oneOf rdf:parseType="Collection">
                            <rdf:Description rdf:about="Clyde"/>
                        </owl:oneOf>
                    </owl:Class>
                </owl:someValuesFrom>
            </owl:Restriction>
        </owl:complementOf>
    </owl:Class>

```

```

        </owl:Class>
        </owl:someValuesFrom>
    </owl:Restriction>
    </owl:complementOf>
</owl:Class>
</rdfs:subClassOf>
</owl:Class>

```

In BL Syntax: $\{ \text{Bonnie} \} \sqsubseteq \neg \exists \text{anzeigen} . \{ \text{Clyde} \}$

In OWL 2:

```

<owl:NegativePropertyAssertion>
  <owl:sourceIndividual rdf:about="Bonnie"/>
  <owl:assertionProperty rdf:about="anzeigen"/>
  <owl:targetIndividual rdf:about="Clyde"/>
</owl:NegativePropertyAssertion>

```

In BL Syntax: $\neg \text{anzeigen}(\text{Bonnie}, \text{Clyde})$

- (d) Niemand zeigt einen Menschen an, mit dem gemeinsam er ein Verbrechen verübt hat.

Nicht möglich in OWL.

- (e) Clyde hat mindestens 10 Verbrechen verübt.

```

<rdf:Description rdf:about="Clyde">
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty rdf:resource="verübt"/>
      <owl:onClass rdf:resource="Verbrechen"/>
      <owl:maxQualifiedCardinality
        rdf:datatype="&xsd;nonNegativeInteger">10
      </owl:maxQualifiedCardinality>
    </owl:Restriction>
  </rdf:type>
</rdf:Description>

```

In BL Syntax: $(\geq 10 \text{ verübt. Verbrechen})(\text{Clyde})$

- (f) Bonnie und Clyde haben mindestens ein Verbrechen gemeinsam verübt.

```

<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <rdf:Description rdf:about="Bonnie"/>
  </owl:oneOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="verübt"/>
    <owl:someValuesFrom>
      <owl:Class>

```

```

<owl:intersectionOf rdf:parseType="Collection">
  <rdf:Description rdf:about="Verbrechen"/>
  <owl:Restriction>
    <owl:onProperty>
      <rdf:Description>
        <owl:inverseOf rdf:resource="verübt"/>
      </rdf:Description>
    </owl:onProperty>
    <owl:someValuesFrom>
      <owl:Class>
        <owl:oneOf rdf:parseType="Collection">
          <rdf:Description rdf:about="Clyde"/>
        </owl:oneOf>
      </owl:Class>
    </owl:someValuesFrom>
  </owl:Restriction>
</owl:intersectionOf>
</owl:Class>
</owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

In BL Syntax: $\{Bonnie\} \sqsubseteq \exists \text{verübt} . (\text{Verbrechen} \sqcap \exists \text{verübt}^{-} . \{Clyde\})$

Was bedeutet das? Wir sagen, dass Bonnie eine Person ist, die ein Verbrechen begangen hat, welches auch von Clyde verübt wurde (Nutzung der inversen Rolle von verübt).

- (g) Wer gemeinsam mit seinem Ehepartner ein Verbrechen verübt hat, der ist nicht ehrlich.
Nicht möglich in OWL.
- (h) Jeder, der einen Verdächtigen kennt, ist selbst verdächtig.

```

<owl:ObjectProperty rdf:about="verdächtig">
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <owl:ObjectProperty rdf:resource="verdächtig"/>
    <owl:ObjectProperty>
      <owl:inverseOf rdf:resource="kennen"/>
    </owl:ObjectProperty>
  </owl:propertyChainAxiom>
</owl:ObjectProperty>

```

In BL Syntax: $\text{verdächtig} \circ \text{kennen}^{-} \sqsubseteq \text{verdächtig}$

Lösung (3.4).

Eine vegetarische Pizza ist eine Pizza, die ...

- (a) "... keine Zutat hat, die gleichzeitig Fleisch und Fisch ist."
- (b) "... nur Beläge hat, die entweder nicht Fleisch oder nicht Fisch sind."
- (c) "... keinen Belag hat, der Fleisch ist, und keinen Belag hat, der Fisch ist."
- (d) "... einen Belag hat, der kein Fleisch ist, und einen Belag hat, der nicht Fisch ist."
- (e) "... nur Zutaten hat, die weder Fleisch noch Fisch sind."

Demnach sind nur die Definitionen von (c) und (e) sinnvoll. (c) ist genauer, weil hier auch fleischliche Zutaten im Teig explizit ausgeschlossen werden.

Lösung (3.5).

Wir verwenden hier als "korrekte" Definition aus Aufgabe 3.4 Axiom (c). Für Axiom (e) müssten wir die Axiome noch weiter korrigieren, um nicht nur einen vegetarischen Belag sicherzustellen sondern allgemein vegetarische Zutaten.

Lösung zu (a) und (b):

- $\text{KäsePizza} \equiv \text{Pizza} \sqcap \exists \text{hatBelag.Käse}$
Keine Unterklasse von VegetarischePizza . Korrektur:
 $\text{KäsePizza} \equiv \text{Pizza} \sqcap \exists \text{hatBelag.Käse} \sqcap \forall \text{hatBelag.Käse}$
- $\text{PizzaSpinat} \equiv \exists \text{hatBelag.Spinat} \sqcap \exists \text{hatBelag.Käse} \sqcap \forall \text{hatBelag.}(\text{Spinat} \sqcup \text{Käse})$
Keine Unterklasse von VegetarischePizza . Korrektur: Erweiterung der Ontologie durch
 $\text{Spinat} \sqsubseteq \text{Gemüse}$
- $\text{PizzaCarnivorus} \equiv \text{Pizza} \sqcap \forall \text{hatBelag.}(\text{Fleisch} \sqcap \text{Fisch})$
Unterklasse von VegetarischePizza . Korrektur:
 $\text{PizzaCarnivorus} \equiv \text{Pizza} \sqcap \forall \text{hatBelag.}(\text{Fleisch} \sqcup \text{Fisch})$
- $\text{LeerePizza} \equiv \text{Pizza} \sqcap \neg \exists \text{hatBelag.T}$
Unterklasse von VegetarischePizza (keine Korrektur nötig).

(c) Verändertes Axiom:

$$\text{VegetarischePizza} \sqsubseteq \text{Pizza} \sqcap \forall \text{hatZutat.}(\neg \text{Fleisch} \sqcap \neg \text{Fisch})$$

Es würde also keine Klasse mehr als Unterklasse von VegetarischePizza klassifiziert werden, z.B., gilt weiterhin (mit obigen Korrekturen):

$$\text{KäsePizza} \sqsubseteq \text{Pizza} \sqcap \forall \text{hatZutat.}(\neg \text{Fleisch} \sqcap \neg \text{Fisch})$$

aber daraus folgt nicht

$$\text{KäsePizza} \sqsubseteq \text{VegetarischePizza}$$