

PD Dr. rer. nat. habil. Sven Groppe

Übungen zur Vorlesung

Semantic Web

WS 2011/2012

Übung 6 – OWL Hintergrundwissen

Lösung

Aufgabe 1:

Entwickle eine Ontologie über Verwandtschaftsbeziehungen:

- a) Modelliere die Beziehungen verheiratet, Kind von, Elternteil von, Vater, Mutter, Geschwister von, Schwester von, Bruder von, Großeltern von, Tante und Onkel.

```
:Woman rdfs:subClassOf :Person .
:Man rdfs:subClassOf :Person .
[] rdf:type owl:AllDisjointClasses ; owl:members ( :Woman :Man ) .

:married rdfs:domain :Person ;
 rdfs:range :Person .
:married rdf:type owl:SymmetricProperty . # :a :married :b => :b :married :a

:childOf rdfs:domain :Person ;
 rdfs:range :Person .
[] rdf:type owl:Restriction ;
 owl:cardinality "2"^^xsd:nonNegativeInteger ;
 owl:onProperty :childOf . # Jeder (mit Eltern) hat genau zwei Elternteile!

:parentOf owl:inverseOf :childOf . # Damit: :A :childOf :B. ⇔ :B :parentOf :A.

:motherOf rdfs:subPropertyOf :parentOf .
:motherOf rdfs:domain :Woman. # rdfs:range wird geerbt!

:fatherOf rdfs:subPropertyOf :parentOf .
:fatherOf rdfs:domain :Man. # rdfs:range wird geerbt!

:sibling owl:propertyChainAxiom ( :childOf :parentOf ) .
:sibling rdf:type owl:IrreflexiveProperty . # Niemand ist Geschwister von sich selbst!

:sisterOf rdfs:subPropertyOf :sibling .
:sisterOf rdfs:domain :Woman. # rdfs:range wird geerbt!

:brotherOf rdfs:subPropertyOf :sibling .
:brotherOf rdfs:domain :Man. # rdfs:range wird geerbt!

:grandparentOf owl:propertyChainAxiom ( :parentOf :parentOf ) .

:auntOf owl:propertyChainAxiom ( :sisterOf :parentOf ) .

:uncleOf owl:propertyChainAxiom ( :brotherOf :parentOf ) .

b) Modelliere die Beziehung verwandt.

:relativeOf rdf:type owl:TransitiveProperty .
:relativeOf rdf:type owl:ReflexiveProperty . # Self !
:relativeOf rdf:type owl:SymmetricProperty .
```

```
:bloodRelativeOf owl:propertyChainAxiom ( :childOf ).# zunächst alle Kinder und Eltern...  
:bloodRelativeOf rdfs:subClassOf :relativeOf.# damit trans./refl./sym. Eigenschaften geerbt!
```

zusätzlich zu den Blutsverwandten:

```
:relativeOf owl:propertyChainAxiom ( :relativeOf :married :relativeOf ).
```

c) Modelliere die Beziehung blutsverwandt.

siehe b)

d) Modelliere die Klasse eheliches Kind.

```
:siblingOfMarriedParent owl:propertyChainAxiom  
( :childOf :married :parentOf ).
```

```
:LegitimateChild owl:equivalentClass [  
  rdf:type owl:Restriction ;  
  owl:onProperty :siblingOfMarriedParent ;  
  owl:hasSelf "true"^^xsd:boolean . ] .
```

e) Modelliere die Klasse nicht-eheliches Kind.

```
:Child owl:equivalentClass [  
  rdf:type owl:Restriction ;  
  owl:minCardinality "1"^^xsd:nonNegativeInteger ;  
  owl:onProperty :childOf  
] .
```

```
:IllegitimateChild owl:equivalentClass [  
  rdf:type owl:Class ;  
  owl:intersectionOf (  
    :Child [ owl:complementOf :LegitimateChild ] )  
] .
```