**《语义网与知识图谱》实验手册**

**实验三 本体建模与本体语言OWL**

目的：

1. 熟悉本体建模工具Protégé,并掌握用Protégé建立本体模型；
2. 了解和熟悉OWL语言的基本语法，掌握采用OWL语言进行本体描述。

内容：

**【习题1】** 在eLearning系统或Protégé官网下载Protégé 5.5，并进行安装。在eLearning系统下载Protégé使用指南，参考使用指南熟悉Protégé的用法。

【**习题2**】 Use OWL DL to model the following sentences.

* The class **Vegetable** is a subclass of **PizzaTopping**.
* The class **PizzaTopping** does not share any elements with the class **Pizza**.
* The individual **aubergine** is an element of the class **Vegetable**.
* The abstract role **hasTopping** is only used for relationships between elements of the classes **Pizza** and **PizzaTopping.**
* The class **VegPizza** consists of those elements which are in the class **NoMeatPizza** and in the class **NoFishPizza**.
* The role **hasTopping** is a subrole of **hasIngredient**.

|  |
| --- |
| <owl:Class rdf:ID=”Vegetable”>  <rdfs:subClassOf rdf:resources=”PizzaTopping”/>  </owl:Class>  <owl:Class rdf.ID=”PizzaTopping”>  <rdfs:disjointedWith rdf:resource=”Pizza”/>  </owl:Class>  <owl:Class rdf:ID=”aubergine”>  <rdfs:subClassOf rdf:resources=”Vegetable”/>  </owl:Class>  <owl:ObjectProerty rdf:ID=”hasTopping”>  <rdfs:domain rdf:resources=”Pizza”/>  <rdfs:range rdf:resources=”PizzaTopping”/>  </owl:ObjectProerty >  <owl:Class rdf:ID=”VegPizza”>  <owl:intersectionOf rdf:parseType=”Collection”>  <owl:Class rdf:about=”NoMeatPizza /”>  <owl:Class rdf:about=”NoFishPizza /”>  </owl:intersectionOf>  </owl:Class>  <owl:ObjectProperty rdf:ID=hasIngredient”>  <rdfs:subPropertyOf rdf:resource=”hasTopping”/>  </owl:ObjectPropertyOf> |

【**习题3**】Use OWL DL to model the following sentences.

* Every **pizza** has at least **two** **toppings**.
* Every **pizza** has **tomato** as **topping**.
* Every **pizza** in the class **PizzaMargarita** has exactly **tomato** and **chesses** as toppings.

|  |
| --- |
| <owl:Class rdf:ID=”pizza”>  <rdfs:subClassOf>  <owl:Restriction>  <owl:onProperty rdf:resource=”hasTopping” />  <owl:minCardinality rdf:datatype=”&xsd;nonNegativeInteger”>2</owl:minCardinality>  </owl:Restriction>  </rdfs:subClassOf>  </owl:Class>  <owl:Class rdf:ID=”pizza”>  <rdfs:subClassOf>  <owl:Restriction>  <owl:onProperty rdf:resource=”hasTopping” />  ~~<owl:someValueFrom rdf:resource=”tomato” />~~  <owl:hasValue rdf:resource=”tomato” />  </owl:Restriction>  </rdfs:subClassOf>  </owl:Class>  <owl:Class rdf:ID=”PizzaMargarita”>  <rdfs:subClassOf>  <rdf:Class>  <owl:intersectionOf rdf:parseType=”Collection”>  <owl:Restriction>  <owl:onProperty rdf:resource=”hasTopping” />  <owl:hasValue rdf:resource=”tomatoTopping” />  </owl:Restriction>  <owl:Restriction>  <owl:onProperty rdf:resource=”hasTopping” />  <owl:hasValue rdf:resource=”cheeseTopping” />  </owl:Restriction>  </owl:intersectionOf>  </rdf:Class>  </rdf:subClassOf>  </rdf:Class> |

【**习题4**】Please model sentences in 3 with Protégé.

|  |
| --- |
|  |

**【习题5】** Write the following part of an RDF document in Turtle syntax, and display the part of an RDF document as a graph:

<owl:Class rdf:about="Parent">

<owl:equivalentClass>

<owl:Restriction>

<owl:onProperty rdf:resource="hasChild"/>

<owl:someValuesFrom rdf:resource="Person"/>

</owl:Restriction>

</owl:equivalentClass>

</owl:Class>

|  |
| --- |
| :Parent owl:equivalentClass [  rdf:type owl:Restriction ;  owl:onProperty :hasChild ;  owl:someValuesFrom :Person  ] .    Q: 何时应该在句尾加分号？ |

【**习题6**】 Write the following part of an RDF document in Turtle syntax, and display the part of an RDF document as a graph:

<rdf:Description rdf:about="Jack">

<rdf:type>

<owl:Class>

<owl:intersectionOf rdf:parseType="Collection">

<owl:Class rdf:about="Person"/>

<owl:Class>

<owl:complementOf rdf:resource="Parent"/>

</owl:Class>

</owl:intersectionOf>

</owl:Class>

</rdf:type>

</rdf:Description>

|  |
| --- |
| :Jack rdf:type [  Rdf:type owl:Class ;  Owl:intersectionOf (:Person [  Rdf:type owl:Class ;  Owl complementOf :Parent ] )  ] .    Q：交集等运算的图该怎么画？是否为如上这样？交集其中一个集合需要描述时，是否为上图这样？ |

【**习题7**】 Write the following Turtle triples in RDF/XML syntax: ([] can be used in Turtle

for a blank node.)

[ ] rdf:type owl:NegativePropertyAssertion ;

owl:sourceIndividual :Bill ;

owl:assertionProperty :hasDaughter ;

owl:targetIndividual :Susan .

|  |
| --- |
| <owl:Class>  <rdf:type rdf:resource = “&owl;NegativePropertyAssertion”>  <owl:sourceIndividual rdf:resource=”Bill” />  <owl:assertionProperty rdf:resource=”hasDaughter” />  <owl:targetIndividual rdf:resource=”Susan ”/ >  </rdf:type>  </rdf:Class> |