

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. add a new property `isWrittenBy` and state that it is the inverse of `isWriterOf`;
2. add a new class `WrittenByMultipleAuthors` and state that it corresponds to the class of movies written by at least two writers;
3. add the new class `allFemaleCast` and state that such a class corresponds to the class consisting of every movie whose writers, directors and actors are all women;
4. add a new class `LowBudgetMovie` and state that it corresponds to the class of movies played by at most 5 actors;
5. add the new class `EuropeanMovie` and state that such a class corresponds to the class consisting of every movie having at least a director who was born in Europe.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning (that is, the models) of the ontology? if so, identify and list such axioms.

① Declaration(`Class(my/ns:Person)`)Declaration(`Class(my/ns:Director)`)Declaration(`Class(my/ns:Writer)`)Declaration(`Class(my/ns:Actor)`)Declaration(`Class(my/ns:Country)`)Declaration(`Class(my/ns:Continent)`)Declaration(`Class(my/ns:Movie)`)Declaration(`Class(my/ns:Comedy)`)Declaration(`Class(my/ns:Drama)`)Declaration(`Class(my/ns:Man)`)Declaration(`Class(my/ns:Woman)`)② `subClassOf(my/ns:Man my/ns:Person)``subClassOf(my/ns:Woman my/ns:Person)`③ `subClassOf(my/ns:Comedy my/ns:Movie)``subClassOf(my/ns:Drama my/ns:Movie)`④ Declaration(`Object Property(my/ns:actsIn)`)Declaration(`Object Property(my/ns:bornIn)`)Declaration(`Object Property(my/ns:filmedInConty)`)Declaration(`Object Property(my/ns:isDirectedBy)`)Declaration(`Object Property(my/ns:isWrittenBy)`)⑤ `subClassOf(ObjectSomeValuesFrom(``ObjectInverseOf(my/ns:isDirectedBy)``owl:Thing)``my/ns:Director)`

subClassOf(ObjectSomeValuesFrom(

my/ns: IsDirectorOf
owl: Thing))

my/ns: Director)

6) subClassOf(owl: Thing

ObjectAllValuesFrom(my/ns: FilmedInYear

my/ns: Movie))

subClassOf(Date SomeValuesFrom(

my/ns: FilmedInYear
owl: Thing))

xsd: integer)

7) subClassOf(ObjectSomeValuesFrom(

ObjectInverseOf(my/ns: FilmedInCountry
owl: Thing))

my/ns: Movie)

subClassOf(ObjectSomeValuesFrom(

my/ns: FilmedInCountry
owl: Thing))

my/ns: Country)

8) subClassOf(ObjectSomeValuesFrom(

ObjectInverseOf(my/ns: bornIn)
owl: Thing))

my/ns: Person)

subClassOf(ObjectSomeValuesFrom(

my/ns: bornIn
owl: Thing))

my/ns: Country)

9) subclassOf(ObjectSomeValuesFrom(ObjectInverseOf(myns: actsIn)
owl: Thing))
myns: Actor)

subclassOf(ObjectSomeValuesFrom(myns: actsIn
owl: Thing))
myns: Movie)

10) subclassOf(ObjectSomeValuesFrom(ObjectInverseOf(myns: isInCountry
owl: Thing))
myns: Country)

subclassOf(ObjectSomeValuesFrom(myns: isInContinent
owl: Thing))
myns: Continent)

11) ObjectPropertyAssertion(myns: isDirectorOf
myns: ANN
myns: XYZ)

ObjectPropertyAssertion(myns: isWriterOf
myns: ANN
myns: XYZ)

12) ObjectPropertyAssertion(myns: actsIn
myns: Joe
myns: ABC)

ObjectPropertyAssertion(myns: actsIn
myns: Paul
myns: ABC)

13) Data Property Assertion (myns:filmedInYear
myns: ABC
'2015')

Object Property Assertion (myns:filmedInContinent
myns: ABC
myns: France)

14) Class Assertion (myns:Woman myns:Ann)

15) Object Property Assertion (myns:isInContinent
myns: France
myns: Europe)

Object Property Assertion (myns:isInContinent
myns: Italy
myns: Europe)

b)

1) Declaration (ObjectProperty (myns:isWrittenBy))

InverseObjectProperty(myns:isWrittenBy myns:isWriterOf)

2) Declaration (Class (WrittenByMultipleAuthors))

EquivalentClasses(myns:WrittenByMultipleAuthors
ObjectIntersectionOf(myns:Movie
ObjectMinCardinality(2 myns:isWrittenBy myns:Writer)
))

3) Declaration (Class (AllFemaleCast))

EquivalentClasses(myns:AllFemaleCast
ObjectIntersectionOf(myns:Movie
ObjectSomeValuesFrom(myns:isWrittenBy
myns:Woman))

ObjectSomeValuesFrom(ObjectInverseOf(myns:isDirectorOf)
myns:Woman)

ObjectSomeValuesFrom(ObjectInverseOf(myns:actsIn)
myns:Woman)

)

)

4) Declaration(*Class(LowBudgetMovie)*)

```
EquivalentClasses(myns:LowBudgetMovie  
    ObjectIntersectionOf(myns:Movie  
        ObjectMaxCardinality(5 ObjectInverseOf(myns:actsIn)  
            myns:Actor)  
        )  
)
```

5) Declaration(*Class(EuropeanMovie)*)

```
EquivalentClasses(myns:EuropeanMovie ObjectIntersectionOf(myns:Movie  
    ObjectSomeValuesFrom(ObjectInverseOf(myns:isDirectorOf)  
        ObjectHasValue(myns:bornIn myns:Europe)  
    )  
)
```

the declaration of the class "isWrittenBy" is redundant because we express the same thing using ObjectFroms(isWriterOf)