

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 `WrittenByManyAuthors` and state that it corresponds to the class of movies written by at least three 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 `ItalianMovie` and state that such a class corresponds to the class consisting of every movie that was filmed in Italy;
6. add the new class `Blockbuster` and state that such a class corresponds to the class consisting of every movie whose box office gross is \$100,000,000.

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:Person`))Declaration (class (`my:Director`))Declaration (class (`my:Writer`))Declaration (class (`my:Actor`))Declaration (class (`my:Country`))Declaration (class (`my:Continent`))Declaration (class (`my:Movie`))Declaration (class (`my:Comedy`))Declaration (class (`my:Drama`))Declaration (class (`my:Rom`))Declaration (class (`my:War`))② subClassOf (`my:Person` `my:Person`)subClassOf (`my:War` `my:Person`)③ subClassOf (`my:Comedy` `my:Movie`)subClassOf (`my:Drama` `my:Movie`)④ Declaration (ObjectProperty (`my:actsIn`))Declaration (ObjectProperty (`my:bornIn`))Declaration (ObjectProperty (`my:filmedIn`))Declaration (ObjectProperty (`my:isDirectedBy`))Declaration (ObjectProperty (`my:isWrittenBy`))⑤ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:isDirectedBy`) `owl:Thing`) `my:Director`)subClassOf (ObjectSomeValuesFrom (`my:isDirectedBy` `owl:Thing`) `my:Director`)⑥ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:filmedIn`) `owl:Thing`) `xsd:integer`)subClassOf (ObjectSomeValuesFrom (`my:filmedIn` `owl:Thing`) `xsd:integer`)⑦ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:filmedIn`) `owl:Thing`) `my:Movie`)subClassOf (ObjectSomeValuesFrom (`my:filmedIn` `owl:Thing`) `my:Country`)⑧ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:bornIn`) `owl:Thing`) `my:Person`)subClassOf (ObjectSomeValuesFrom (`my:bornIn` `owl:Thing`) `my:Country`)⑨ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:actsIn`) `owl:Thing`) `my:Actor`)subClassOf (ObjectSomeValuesFrom (`my:actsIn` `owl:Thing`) `my:Movie`)⑩ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:isIn`) `owl:Thing`) `my:Country`)subClassOf (ObjectSomeValuesFrom (`my:isIn` `owl:Thing`) `my:Continent`)⑪ subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (`my:boxOfficeGross`) `owl:Thing`) `my:Movie`)

- subClassOf (ObjectSomeValuesFrom (myns:isWrittenBy owl:Thing , myns:Movie)
 11) subClassOf (ObjectSomeValuesFrom (ObjectInverseOf (myns:hasBoxOfficeGross) owl:Thing) myns:Movie)
 subClassOf (ObjectSomeValuesFrom (myns:hasBoxOfficeGross owl:Thing) xsd:integer)
- 12) ObjectPropertyAssertion (myns:isWrittenOf myns:Ann myns:XYZ)
 ObjectPropertyAssertion (myns:isWrittenOf myns:Ann myns:XYZ)
- 13) ObjectPropertyAssertion (myns:actsIn myns:Tom myns:ABC)
 ObjectPropertyAssertion (myns:actsIn myns:Paul myns:ABC)
- 14) DataPropertyAssertion (myns:filmedInYear myns:ABC "2015" \wedge xsd:integer)
- 15) ClassAssertions (myns:Woman myns:Ann)
- 16) ObjectPropertyAssertion (myns:isInContinent myns:Italy myns:Europe)
 ObjectPropertyAssertion (myns:isInContinent myns:France myns:Europe)

b)

- 1) Declaration(ObjectProperty(myns:isWrittenBy))
 EquivalentObjectProperty(myns:isWrittenBy ObjectInverseOf(myns:isWriterOf))
- 2) Declaration(Class(myns:WrittenByMultipleAuthors))
 EquivalentClasses(myns:WrittenByMultipleAuthors ObjectIntersectionOf(myns: Movie ObjectMinCardinality(2 myns:isWrittenBy myns:Writers))
- 3) Declaration(Class(myns:AllFemaleCast))
 EquivalentClasses(myns:AllFemaleCast ObjectIntersectionOf(myns:Movie ObjectAllValuesFrom(myns:isWrittenBy myns:Woman)
 ObjectAllValuesFrom(ObjectInverseOf(myns:isDirectorOf) myns:Woman) ObjectAllValuesFrom(ObjectInverseOf(myns:actsIn) myns:Woman)))
- 4) Declaration(Class(myns:LowBudgetMovie))
 EquivalentClasses(myns:LowBudgetMovie ObjectIntersectionOf(myns:Movie ObjectMaxCardinality(5 ObjectInverseOf(myns:actsIn) myns:Actor)))
- 5) Declaration(Class(myns:ItalianMovie))
 EquivalentClasses(myns:ItalianMovie ObjectIntersectionOf(myns:Movie ObjectHasValue(myns:filmedIn myns:Italy)))
- 6) Declaration(Class(myns:Blockbuster))
 EquivalentClasses(myns:Blockbuster ObjectIntersectionOf(myns:Movies DataHasValue(myns:hasBoxOfficeGross "100,000,000")))