Narrative Ontology

Semantic Web

Elia Piccoli
621332

Q8. Identify two different assertions that would make the ontology inconsistent.

In order to create an inconsistency in our ontology, we declare an individual that has type *NotHuman* and *Object* at the same time. This will lead to inconsistency since in the class taxonomy *NotHuman* is *subClassOf Character* and the latter is defined to be disjoint with *Object*.

```
###  http://www.semanticweb.org/exam/narrative#inconsistentIndividual1
:inconsistentIndividual1 rdf:type owl:NamedIndividual ,
                                  :NotHuman ,
                                  :Object .
```

Protégé Explanation:

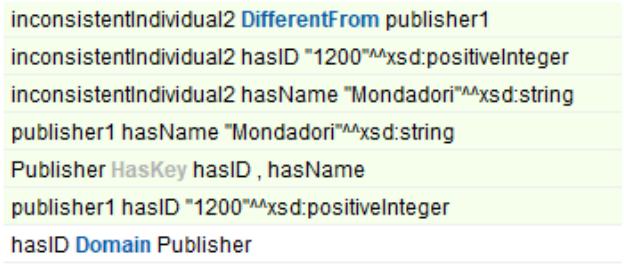
Another possibility to create an inconsistency in our ontology, is to violate the *HasKey* property. In the class taxonomy we defined that *Publisher* is identified by *hasID, hasName*. If we declare an individual that has same value of the DataProperty of one of the publishers but is defined to be a different individual with respect to it, then we will have an inconsistency.

```
###  http://www.semanticweb.org/exam/narrative#inconsistentIndividual2
:inconsistentIndividual2 rdf:type owl:NamedIndividual ,
                                  :Publisher ;
            :hasID "1200"^^xsd:positiveInteger ;
            :hasName "Mondadori"^^xsd:string .

###  http://www.semanticweb.org/exam/narrative#publisher1
:publisher1 rdf:type owl:NamedIndividual ,
                     :Publisher ;
            :offersContractTo :narrator1 ;
            :sellsBookIn :bookshop1 ,
                         :bookshop2 ;
            :hasID "1200"^^xsd:positiveInteger ;
            :hasName "Mondadori"^^xsd:string .

[ rdf:type owl:AllDifferent ;
  owl:distinctMembers ( :inconsistentIndividual2
                        :publisher1
                      )
] .
```

Protégé Explanation:



There are other possible ways to create an inconsistency for instance forcing the cardinality constraint of the *ObjectProperty* (i.e. Book hasPublisher exactly 1 Publisher)

Q9. Define the complex role inclusion axiom capturing the fact that if a narrator creates a narrative that is reported in a book that is published by a publisher, then the narrator has a contract with that publisher.

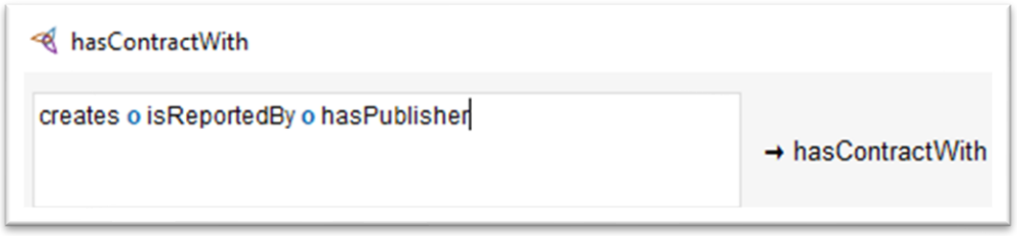
In order to define the complex role inclusion axiom, we define the sets of *Classes* and *ObjectProperty* involved.

V_c = {:Book, :Narrator, :Narrative, :Publisher}
V_{op} = {:creates, :hasPublisher, :hasContractWith, :isReportedBy}

Now we can define the complex role inclusion axiom.

```
SubObjectPropertyOf(
    ObjectPropertyChain(
        :creates
        :isReportedBy
        :hasPublisher
    )
    :hasContractWith
)
```

The same chain can be represented in Protégé with the following formula:



Q10. Verify if the created ontology (including the complex role inclusion axiom defined in Q9) satisfies the global restrictions on the axioms of an OWL 2 DL ontology.

Global Restrictions on OWL 2 DL ontology axioms must be satisfied in order to retain decidability of the ontology.
There are 5 different constraints which need to be met:

Restriction on owl:topDataProperty

This restriction is satisfied because *owl:topDataProperty* appears only on the second argument of *SubDataPropertyOf* axioms, in fact in the ontology it does not exist a super-property of *owl:topDataProperty*.

Restrictions on Datatypes

This restriction is satisfied because in the ontology there are not definition of new *Datatypes*. In fact, all of them come from *OWL 2 datatype map* for which all the conditions are satisfied.

Restriction on Simple Roles

This restriction is satisfied because in the ontology there are two composite object properties – *hasContractWith*, *writtenBy* – and both are not used in an axiom of the critical ones. Further developing the previous statement, the focus should shift to the ontology axioms, where it is easy to see that *writtenBy* has no *PropertyRestrictions* nor *PropertyCharacteristics*. On the other hand, considering *hasContractWith*, a *PropertyRestriction* must be defined because of axiom 9 ("The narrator has at least one contract with a publisher"). In the ontology this restriction has been represented using the *ObjectPropertyRestriction ObjectSomeValuesFrom* which is not part of the forbidden ones so does not imply inconsistency.

An interesting observation that can be developed over this restriction is that it could also be represented using the *ObjectCardinalityPropertyRestriction ObjectMinCardinality* (In Protégé: Narrator hasContractWith min 1 Publisher). At first glance, knowing that *hasContractWith* is composite *ObjectProperty*, it would suggest that this formalization does not satisfy the *GlobalRestriction*, even if semantically this has the same meaning of the definition used in the ontology. The interesting part of this observation is related to the behavior of the reasoners. As a matter of fact, reasoning with *HermiT* does not raise any exception when using *minCardinality 1* since it may consider this definition equivalent to *some*. Instead other reasoners will complain stating that the ontology does not respect the *GlobalRestrictions*.

Restriction on the Property Hierarchy

This restriction is satisfied since in the ontology there are only two property chain *hasContractWith, writtenBy* and for both we can define a partial ordering such that it meets the condition and assures the absence of cycles.

Restrictions on Anonymous Individuals

This restriction is satisfied since in the ontology there are no *Anonymous Individuals*.

Q11. Write the following queries in SPARQL:

Q11.1. Find how many events occurred in real locations, grouped by location.

```
prefix : <http://www.semanticweb.org/exam/narrative#>

SELECT ?location (COUNT(?event) AS ?numEvent)
WHERE {
    ?location a :RealLocation .
    ?event a :Event ;
        :occursIn ?location .
}
GROUP BY ?location
```

Q11.2. Find all the books with the ID of the publisher lower than 5000.

```
prefix : <http://www.semanticweb.org/exam/narrative#>

SELECT ?book
WHERE {
    ?book a :Book .
    ?publisher a :Publisher ;
        :publishes ?book ;
        :hasID ?publisherID .
    FILTER ( ?publisherID < 5000 )
}
```

Q11.3. Find all the events that do not have any human participants.

```
prefix : <http://www.semanticweb.org/exam/narrative#>

SELECT ?event
WHERE {
    ?event a :Event .
    FILTER NOT EXISTS {
        ?partecipant a :Human ;
            :isCharacterOf ?event .
    }
}
```

Q11.4. Find the number of the narratives that are published in a book, along with the title of the book, the ISBN code of the book and the publisher of the book.

```
prefix : <http://www.semanticweb.org/exam/narrative#>

SELECT ?book ?bookISBN ?bookTitle ?publisher (COUNT(?narrative) AS ?numNarrativ
es)
WHERE {
    ?book a :Book ;
        :hasISBN ?bookISBN ;
        :hasTitle ?bookTitle ;
        :reports ?narrative ;
        :hasPublisher ?publisher .
    ?narrative a :Narrative .
    ?publisher a :Publisher .
    }
GROUP BY ?book ?bookISBN ?bookTitle ?publisher
```

Q11.5. Find all the distinct events that have a human participant or occur in a real location.

```
prefix : <http://www.semanticweb.org/exam/narrative#>

SELECT DISTINCT ?event
WHERE {
    ?event a :Event .
    {
        ?human a :Human ;
            :isCharacterOf ?event .
    }
    UNION
    {
        ?location a :RealLocation ;
            :isRelatedTo ?event .
    }
}
```