INF3580/INF4580 – MANDATORY EXERCISE 6

Published date: 03.04.2018.

First due date: 25.04.2018 23:59

Final due date: 16.05.2018 23:59

Delivery files: 3 files: 2 OWL files, 1 pdf file containing answers to questions.

Delivery attempts: 2.

1 Exercise: OWL modelling

In this exercise you shall create an OWL ontology. We suggest you use Protégé to do this. (Tip: we had the best reasoning response using Pellet or FaCT++, and not HermiT.)

Classes	**Object properties**	**Datatype properties**
`Animal`	`hasCompeted`	`name`
`Horse`	`hasWon`	`age`
`Male`	`hasTrainer`	
`Female`	`trains`	
`WarmbloodedHorse`	`hadInterestIn`	
`ColdbloodedHorse`	`sire`	
`Filly`	`dam`	
`Stallion`	`ancestor`	
`RaceHorse`		
`MonteHorse`		
`SuperHorse`		
`HorseRace`		
`MonteRace`		
`Person`		

Use resources from the table above to model the statements below. Create a suitable namespace for your ontology, and put all statements in the same OWL file.

Statements:

1. The range of `name` is `xsd:string`, and the range of `age` is `xsd:int`.
2. `name` and `age` are functional.
3. Every horse is an animal.
4. Every animal is a male or a female.
5. Nothing is both a male and a female.
6. Every warmblooded horse is a horse.
7. Every coldblooded horse is a horse.
8. Nothing is both a warmblooded horse and a coldblooded horse.
9. A filly is by definition a female horse younger than 4.
10. A stallion is by definition a male horse of age 4 or older.

11. A race horse is by definition a horse which has competed in a horse race.
12. Every monté race is a horse race.
13. Every monté horse is a race horse which has only competed in monté races.
14. A superhorse is by definition a race horse which has won more then 100 horse races.
15. You must compete in a race to win a race.
16. Every warmblooded horse which has competed in a monté race must be between 4–12 years old. (Tip: you might need to use the "general axioms" part of the "Active Ontology" pane in Protégé to express this statement.)
17. Every race horse has a name.
18. `name` is a key for race horse.
19. Every race horse has at least one trainer which is a person.
20. Every person has a name.
21. If a horse has a trainer, then this trainer trains the horse.
22. Every trainer had interest in a race if a horse he/she trains competed in this race. (Tip: use property chains.)
23. No person is a horse, no person is a horse race, and no horse is a horse race.
24. Every horse has exactly one sire (father), which must be a male horse, and one dam (mother), which must be a female horse.
25. `sire` and `dam` are properties which relate horses.
26. `sire` and `dam` are subproperties of `ancestor`. `ancestor` is a transitive property.
27. Set natural domain and range restrictions to the properties `sire`, `dam`, `hasTrainer`

Add some data about individuals (create suitable identifiers):

1. The horse called "Rex Rodney" is a superhorse. Its trainer is "Kjell Håkonsen", and it won the race Elitloppet (in 1986, use the identifier `:Elitloppet1986`). Its sire has the name "Doctor Rodney".
2. `:Elitloppet1986` is not a monté race.
3. The horse called "Steady, Ready, Go" is a filly and monté horse.

1.1 DL Axioms

Express the statements 3, 4, 5, 13, 14, 15, 19, 21; and the three statements about individuals as DL Axioms, e.g., no. 6: "Every warmblooded horse is a horse" can be written as $WarmbloodedHorse \sqsubseteq Horse$.

1.2 Questions

Answer the following questions with "Yes", "No" or "Unknown", and a sentence or two to back up the claim, referring to the axioms you have created in your ontology as arguments.

1. Is Rex Rodney a male or a female animal?
2. Does Kjell Håkonsen have interest in Elitloppet 1986?
3. Is Rex Rodney a warmblooded horse?
4. Is Steady, Ready, Go a warmblooded horse?
5. Is Steady, Ready, Go a coldblooded horse?
6. Did Steady, Ready, Go compete in Elitloppet 1986?

2 Exercise: Mapping dbpedia to your ontology

The task of this exercise is to create a mapping of parts of the dbpedia vocabulary about horses and race horses to your ontology from the exercise above.

Below is an excerpt from the result of running a DESCRIBE query[1] on the dbpedia.org SPARQL endpoint: `http://dbpedia.org/sparql`. The excerpt contains data about two race horses, one of them is listed below, a race horse called "In the Wings". The exercise is to create a mapping ontology which maps all properties mentioned in the excerpt to the ontology you created in the exercise above.

The mapping shall be created by giving a series of axioms (i.e., sub-class/-property and equivalence axioms) which relate the dbpedia properties to your classes and properties, e.g,

```
dbpprop:dam  owl:equivalentProperty :dam
```

creates a mapping between the `dbpprop:dam` and "your" `:dam` property. Note that the lefthand- and righthand-side of the axioms can be complex classes or properties, e.g., `dbpprop:horsename` should be mapped to some combination of `:name` and `:Horse` (not only to `:name`) (Tip: you need to use the general axioms feature in Protégé to express this.)

The excerpt:

```
<http://dbpedia.org/resource/In_the_Wings_%28horse%29>
    dbpprop:f <http://dbpedia.org/resource/Sadler%27s_Wells_%28horse%29> ;
    dbpprop:ff [ dbpprop:name "Northern Dancer" ] ;
    dbpprop:fm [ dbpprop:name "Fairy Bridge" ] ;
    dbpprop:horsename "In the Wings" ;
    dbpprop:m [ dbpprop:name "High Hawk" ];
    dbpprop:mf dbpedia:Shirley_Heights ;
    dbpprop:mm [ dbpprop:name "Sunbittern" ] ;
    dbpprop:name "In the Wings" ;
    dbpprop:race <http://dbpedia.org/resource/Breeders%27_Cup_Turf>,
                 dbpedia:Coronation_Cup,
                 dbpedia:Grand_Prix_de_Saint-Cloud,
                 dbpedia:Prix_Foy,
                 <http://dbpedia.org/resource/Prix_du_Prince_d%27Orange> ;
    dbpprop:sex dbpedia:Stallion ;
    dbpprop:sire <http://dbpedia.org/resource/Sadler%27s_Wells_%28horse%29> ;
    dbpprop:trainer <http://dbpedia.org/resource/Andr%C3%A9_Fabre> ;
    ### ...
    .
```

[1] FYI: The query is put in the "Appendix".

This excerpt, including all the properties you need to map, is found in the ontology at the following address:

http://sws.ifi.uio.no/inf3580/v14/oblig/6/racehorse

Your mapping ontology should be a separate file and it should import the ontology at the above address from this address and additionally the ontology you have made in the first exercise.

Notes:

- In the properties `dbpprop:m`, `dbpprop:fm`, `dbpprop:mm`, and so on, `m` means "mother" and `f` "father", so two letters means a grand-parent, e.g., `dbpprop:fm` is the mother to the father (norwegian: farmor) to the horse.
- In the excerpt `dbpedia:Stallion` is an individual (and not a class).
- If the intended meaning of the properties or other resources in the excerpt is not clear, use a reasonable interpretation you find fit.
- Since you will be working with resources from different ontologies, keeping them apart is smart. To display names in Protégé using qualified names enable the setting found in Protégé under the File -> Preferences -> Renderer -> "Render by qualified name".

2.1 Questions

Apply reasoning to your ontology and answer the following questions:

1. List all stallions.
2. List all male horses.
3. Since "Northern Dancer" is an ancestor of "In the Wings", the ancestors of "Northern Dancer" should also be the ancestors of "In the Wings". Why is this not the case?

3 INF4580

3.1 Punning

In oblig 5, you learnt that the official RDF semantics gives a meaning also to triples where the subjects and objects can be classes or properties. This is not supported by the "DL" semantics however, and indeed, non of the reasoners connected to Protégé are able to deal with ontologies where individuals, classes, and properties are not clearly separated.

Still, it is possible to load such ontologies into Protégé, based on a mechanism called "punning," see http://www.w3.org/TR/owl2-new-features/#F12:_Punning.

Let Γ_3 be the ontology listed below, using Manchester syntax for brevity. A turtle variant is listed in the appendix.

```
ObjectProperty: member
    InverseOf: rdf:type

Class: OrderOfAnimals
```

```
Class: OrderOfMammals
    EquivalentTo: OrderOfAnimals and (member only Mammal)

Class: Mammal
    Types: OrderOfMammals

Class: Elephant
    SubClassOf: Mammal

Individual: Bobo
    Types: Elephant
```

An OrderOfMammals is an OrderOfAnimals whose members must be Mammal-s. Since Elephant is a subclass of Mammal, which is a (, i.e., has type) OrderOfMammals, one could expect that a DL reasoner would infer that Elephant is an OrderOfMammals. However, this is not the case. Explain why.

3.2 Annotation properties

In OWL, rdfs:label is a predefined annotation property, see http://www.w3.org/TR/owl-ref/#Annotations. Explain what an annotation property is and why they are useful.

3.3 OWL profiles

For each of the axioms in 1.1 decide if they are "in isolation" a

1. valid OWL EL axiom, and
2. valid OWL QL axiom.

Here, "in isolation" means that you shall consider each axiom as though it is the only axiom in a given ontology, i.e., you don't have to consider other axioms that the resources may appear in.

4 Ending notes

Mandatory exercises are to be handed in using Devilry[2] . Make sure that you are registered in the system by logging on and finding that an oblig6 is available as an assignment in INF3580 or INF4580. *Check this before you start solving the exercises!* If you are not registered in the system, give notice to leifhka@ifi.uio.no.

Good Luck!

5 Appendix

5.1 SPARQL mapping query

This is the SPARQL query used to create the excerpt.

[2] https://devilry.ifi.uio.no/

```
DESCRIBE ?horse
WHERE
{
  ?horse dbpprop:horsename [] ;
  dbpprop:race [] .
  { ?horse dcterms:subject
      <http://dbpedia.org/resource/Category:Racehorses>  }
  UNION
  { ?horse dcterms:subject
      [ skos:broader
       <http://dbpedia.org/resource/Category:Racehorses> ] }
}
```

The result of this query (8MB) is found at

http://sws.ifi.uio.no/inf3580/v14/oblig/6/dbpedia-dump

5.2 Punning Turtle file

```
@prefix : <http://example.org#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:member rdf:type owl:ObjectProperty ;
        owl:inverseOf rdf:type .

:OrderOfAnimals rdf:type owl:Class .

:OrderOfMammals rdf:type owl:Class ;
    owl:equivalentClass  [ rdf:type owl:Class ;
                           owl:intersectionOf
                              ( :OrderOfAnimals
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty :member ;
                                  owl:allValuesFrom :Mammal
                                ]
                              )
                         ] .

:Mammal  rdf:type  :OrderOfMammals .

:Elephant  rdfs:subClassOf :Mammal .

:Bobo    rdf:type  :Elephant .
```