

ESERCIZIO 1

```
<owl:Class rdf:ID="Classe1"/>
<owl:Class rdf:ID="Classe2"/>
<owl:Class rdf:ID="Obiettivo1">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#prop1"/>
      <owl:someValuesFrom rdf:resource="#Classe2"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Obiettivo2">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="#prop1"/>
          <owl:someValuesFrom rdf:resource="#Classe2"/>
        </owl:Restriction>
        <owl:Class rdf:about="#Classe1"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Obiettivo3">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="#prop1"/>
          <owl:hasValue rdf:resource="#Inst1"/>
        </owl:Restriction>
        <owl:Class rdf:about="#Classe1"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
```

</owl:Class>

<owl:ObjectProperty rdf:ID="prop1">

<rdfs:range rdf:resource="#Classe2"/>

</owl:ObjectProperty>

<Class2 rdf:ID="Inst1 "/>

determinare:

- se esistono eventuali inconsistenze
- ristrutturare l'albero delle classi (a seguito di relazioni rdfs:subClassOf inferite)

Suggerimento: rappresentare le classi come insiemi (non dimenticate le restrizioni).

SOLUZIONE:

Albero delle classi:

Classe1

Obiettivo2

Classe1

Obiettivo3

Obiettivo1

Obiettivo3

Spiegazione:

- Classe1 è sottoclasse di Obiettivo2 perchè Obiettivo2 è **equivalente** all'**unione** di Classe1 con qualcosa, quindi sicuramente se un'istanza appartiene a Classe1 allora apparterrà anche a Obiettivo2, ma non è detto il contrario
- Obiettivo3 è sottoclasse di Classe1 perchè Obiettivo3 è **equivalente** all'**intersezione** di Classe1 con qualcosa, quindi se qualcosa appartiene a Obiettivo3 appartiene sicuramente a Classe1, ma non è detto il contrario
- Obiettivo1 è sottoclasse di Obiettivo2 perchè Obiettivo2 è **equivalente** all'**unione** di una restrizione , che è equivalente a Obiettivo1, con qualcosa,
- Obiettivo3 è sottoclasse di Obiettivo1 perchè Obiettivo3 è **equivalente** all'**intersezione** di una restrizione che risulta essere sottoclasse della restrizione che è equivalente ad Obiettivo1. Infatti la restrizione di Obiettivo1 dice che basta avere come valore della proprietà prop1 una istanza della classe Classe2, mentre la restrizione usata in Obiettivo3 dice che bisogna avere una precisa istanza come valore di prop1 e tale istanza appartiene a Class2
- Se viene fatta una rappresentazione ad insiemi delle classi, tutto il ragionamento risulta più chiaro

NOTA BENE: Obiettivo3 risulta essere sottoclasse sia di Classe1 che di Obiettivo1, senza che tra questa due classi ci sia nessun legame. Questo NON è una inconsistenza, è perfettamente possibile

Esercizio2

```
<owl:Class rdf:ID="Classe1"/>
<owl:Class rdf:ID="Obiettivo1">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#prop1"/>
      <hasValue rdf:resource="#B"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Obiettivo2">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#prop2"/>
      <owl:hasValue rdf:resource="#B"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>

<owl:ObjectProperty rdf:ID="prop1">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
  <rdfs:domain rdf:resource="#Class1"/>
  <rdfs:range rdf:resource="#Classe1"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="prop2">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
  <rdfs:subPropertyOf rdf:resource="#prop1"/>
</owl:ObjectProperty>

<owl:Thing rdf:ID="A">
  <prop2 rdf:resource="#B"/>
</owl:Thing>
<owl:Thing rdf:ID="B">
  <prop2 rdf:resource="#C"/>
</owl:Thing>
<owl:Thing rdf:ID="C">
  <prop2 rdf:resource="#D"/>
```

```

</owl:Thing>
<owl:Thing rdf:ID="E">
    <prop1 rdf:resource="#C"/>
</owl:Thing>

```

Suggerimento: rappresentare il grafo delle connessioni tra le istanze (esplicite ed inferite).

determinare:

- se esistono eventuali inconsistenze
- ristrutturare l'albero delle classi (a seguito di relazioni `rdfs:subClassOf` inferite)
- riclassificare le istanze ove possibile

SOLUZIONE:

Istanze

Classe1: A, B, C, D, E perché tutte queste classi hanno prop1 o prop2 in ingresso o in uscita e prop1 e prop2 hanno come dominio Class1 (prop2, essendo sottoproprietà di prop1, ne eredita il dominio ed in range e NON EREDITA ALTRO)

Obiettivo1 (istanze connesse tramite prop1 a B):

- A: perché direttamente connessa a B con prop2 e prop2 è sottoproprietà di prop1, quindi ogni volta che ho prop2 posso inferire che prop2 connette le stesse istanze
- B: perché connessa con prop2 a C, ma prop2 è simmetrica e quindi posso inferire che "C prop2 B" e poi, visto che prop2 è sottoproprietà di prop1 ho che "B prop1 C" e "C prop1 B" e prop1 è transitiva, quindi ho "B prop1 B"
- C: per quanto detto sopra
- D: ho "C prop2 D", quindi "D prop2 C" (simmetrica), "D prop1 C" (sottoproprietà), e poi, usando le informazioni dette prima con prop1 transitiva, ho "D prop1 B"
- E: usando "B prop2 C", e "E prop1 C", sfrutto la sottoproprietà, la simmetria e la transitività per avere "E prop1 B"

Obiettivo2 (le istanze connesse a B tramite prop2 a B)

- A: ho direttamente che "A prop2 B"
- C: ho che "B prop2 C" e, visto che prop2 è simmetrica, ho che "C prop2 B"

Albero delle classi:

Classe1

Obiettivo1

Obiettivo2

Abbiamo che Obiettivo2 è sottoclasse di Obiettivo1, perché per appartenere a Obiettivo2 devo essere connesso con prop2 a B, ma visto che prop2 è sottoproprietà di prop1, allora ogni volta che ho "X prop2 B", allora avrò sempre che "X prop1 B", che è proprio la condizione per appartenere a Obiettivo1

NOTA BENE: attenzione che tutti questi ragionamenti sono stati possibili visto che sia Obiettivo1 che Obiettivo2 sono EQUIVALENTI a delle restrizioni, se invece erano sottoclassi delle restrizioni, non potevo inferire l'appartenenza dell'istanza, ad esempio se Obiettivo1 era sottoclasse della stessa restrizione espressa nel testo dell'esercizio, anche se avevo ancora che "A prop2 B" e quindi "A prop1 B", avevo che Obiettivo1 era sottoclasse della restrizione e quindi NON potevo più inferire che A appartiene a Obiettivo1. Quindi quando si ha l'equivalenza tra classi, posso fare le inferenze in entrambi i versi (posso passare da una classe all'altra), ma quando ho la sottoclasse, posso passare solo dalla sottoclasse alla superclasse (cioè se mi viene detto che ho una istanza di Obiettivo1 ALLORA posso inferire che appartiene alla restrizione, ma non posso dire il contrario)

Esercizio 3

Svolgere l'esercizio2, ma questa volta le caratteristiche delle proprietà sono invertire, cioè abbiamo:

```
<owl:ObjectProperty rdf:ID="prop1">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
  <rdfs:domain rdf:resource="#Class1"/>
  <rdfs:range rdf:resource="#Classe1"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="prop2">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
  <rdfs:subPropertyOf rdf:resource="#prop1"/>
</owl:ObjectProperty>
```

Esercizio 4

Data la seguente ontologia:

```
<owl:Class rdf:ID="Persona"/>
<owl:Class rdf:ID="Uomo">
  <rdfs:subClassOf rdf:resource="#Persona"/>
</owl:Class>
<owl:Class rdf:ID="Donna">
  <rdfs:subClassOf rdf:resource="#Persona"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="amico">
  <rdfs:domain rdf:resource="#Persona"/>
  <rdfs:range rdf:resource="#Persona"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="genitore">
  <rdfs:domain rdf:resource="#Persona"/>
  <rdfs:comment xml:lang="it"> va dal figlio alla padre/madre </rdfs:comment>
  <rdfs:range rdf:resource="#Persona"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="età">
  <rdfs:domain rdf:resource="#Persona"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="peso">
  <rdfs:domain rdf:resource="#Persona"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
```

D1:

Definire una query SPARQL che restituisca la lista, senza ripetizioni, di tutte le donne che hanno almeno 3 amici, di cui almeno uno con almeno 30 anni

D2:

Definire una query SPARQL che restituisca la lista, senza ripetizioni, di tutte le donne che hanno un peso compreso tra i 50 e i 90 kg

D3:

Definire una query SPARQL che restituisca la lista di tutti le donne che hanno una sorella

SOLUZIONI:

R1:

```
SELECT DISTINCT ?donna
```

```
WHERE{
```

```
    ?donna      a      :Donna .
```

```
    ?donna      :amico  ?amico1 .
```

```
    ?donna      :amico  ?amico2 .
```

```
    ?donna      :amico  ?amico3 .
```

```
    FILTER( ?amico1 != ?amico2 && ?amico1 != ?amico3 && ?amico2 != ?amico3)
```

```
    ?amico2      :età      ?età2 .
```

```
    FILTER( ?età1 >= 30)
```

```
}
```

R2:

```
SELECT DISTINCT ?donna
```

```
WHERE{
```

```
    ?donna      a      :Donna .
```

```
    ?donna      :peso      ?peso.
```

```
    FILTER( ?peso >= 50 && ?peso <= 90)
```

```
}
```

R3:

```
SELECT ?donna
```

```
WHERE{
```

```
    ?donna      a      :Donna.
```

```
    ?donna      :genitore  ?genitore .
```

```
    ?sorella     :genitore  ?genitore .
```

```
    ?sorella     a      :Donna.
```

```
    FILTER( ?donna != ?sorella)
```

```
}
```