

# Esercizi Ontologie

## Esercizio 1

### Progettazione di una ontologia.

Progettare ed esprimere una semplice ontologia, utilizzando il formalismo delle Description Logics o una formulazione (astratta o XML) della sintassi OWL, che descriva il dominio universitario così come richiesto nelle specifiche.

Al fine di semplificare la scrittura, è possibile rappresentare la gerarchia esplicita delle classi attraverso un formalismo grafico (un albero di classi).

### Specifiche:

Una università è costituita da elementi: personale amministrativo, accademico e da studenti.

Ogni elemento universitario è caratterizzato da dati personali (nome, cognome, indirizzo, eventuali recapiti telefonici) e-mail di riferimento.

Gli studenti devono avere un solo numero di matricola.

Ogni membro del personale amministrativo e accademico ha uno studio/ufficio assegnato, con un numero di stanza ed un numero telefonico associato.

In ogni stanza possono lavorare più persone, ma un accademico ed un amministrativo non condivideranno mai la stessa stanza.

Un docente avrà dei corsi assegnati, ai quali possono iscriversi degli studenti.

\*\*\*\*\*

### Problema da risolvere

Descrivere la classe degli studenti del professor Rossi, alla luce di quanto rappresentato nell'ontologia.

## Soluzione:

### tassonomia delle IS-A esplicite



### Elenco delle proprietà e delle relative caratteristiche (facets).

- docente ↔ insegna
- insegna ↔ docente
- iscritti ↔ frequenta
- frequenta ↔ iscritti
- ha\_stanza ↔ assegnata\_a
- assegnata\_a ↔ ha\_stanza
- TelefonoStanza
- NumeroDiStanza
- NumeroDiMatricola
- e-mail
- RecapitiTelefonici
- Indirizzo
- Cognome
- Nome

### Dettaglio in sintassi astratta di OWL

```
DatatypeProperty(Nome Functional
  domain(MembroUniversità)
  range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(Cognome Functional
  domain(MembroUniversità)
  range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(Indirizzo Functional
  domain(MembroUniversità)
  range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(RecapitiTelefonici
  domain(MembroUniversità)
  range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(e-mail functional
  domain(MembroUniversità)
  range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(NumeroDiMatricola
```

```
Functional
InverseFunctional
domain(Studente)
range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
DatatypeProperty(NumeroDiStanza
Functional
InverseFunctional
domain(Stanza)
range(<http://www.w3.org/2001/XMLSchema#integer>))
```

```
DatatypeProperty(TelefonoStanza Functional
InverseFunctional
domain(Stanza)
range(<http://www.w3.org/2001/XMLSchema#string>))
```

```
ObjectProperty(assegnata_a inverseOf(ha_stanza)
InverseFunctional
domain(Stanza)
range(unionOf(Amministrativo
Accademico)))
```

```
ObjectProperty(ha_stanza inverseOf(assegnata_a)
domain(unionOf(Amministrativo
Accademico))
range(Stanza))
```

```
ObjectProperty(frequenta inverseOf(iscritti)
domain(Studente)
range(Corso))
```

```
ObjectProperty(iscritti inverseOf(frequenta)
domain(Corso)
range(Studente))
```

```
ObjectProperty(insegna inverseOf(docente)
domain(Accademico)
range(Corso))
```

```
ObjectProperty(docente inverseOf(insegna)
domain(Corso)
range(Accademico))
```

```
Class(Studio
equivalentClass(restrictionOnProperty(assegnata_a
someValuesFrom(Accademico)))
DisjointWith(Ufficio))
```

```
Class(Ufficio
equivalentClass(restrictionOnProperty(assegnata_a
someValuesFrom(Amministrativo)))
DisjointWith(Studio))
```

La classe degli studenti del professor Rossi è fornita dalla seguente espressione DL:

$\exists$  frequenta.(docente  $\exists$  Rossi)

## Esercizio 2

L'esempio seguente presenta qualche inconsistenza? Se sì, su quale oggetto e perchè?

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Document">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        1
      </owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="author"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#author">
  <rdfs:range rdf:resource="#Person"/>
</owl:ObjectProperty>
<Document rdf:ID="myDoc"/>
```

### Soluzione:

Nessuna; si potrebbe pensare ad una incompletezza di myDoc. Ma l'assunzione di Open World ammette la possibilità che l'autore del libro myDoc (la cui esistenza è forzata dal vincolo di mincard=1 sulla proprietà author quando questa è applicata ad un libro) sia definito "da qualche parte".

## Esercizio 3

L'esempio seguente presenta qualche inconsistenza? Se sì, su quale oggetto e perchè?

```
<owl:Class rdf:ID="Institute"/>
<owl:Class rdf:ID="Document">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        1
      </owl:maxCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="copyrightHolder"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="author">
  <rdfs:range rdf:resource="#Institute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#copyrightHolder">
  <rdfs:range rdf:resource="#Institute"/>
</owl:ObjectProperty>
<Institute rdf:ID="Institute1"/>
<Institute rdf:ID="Institute2"/>
<Document rdf:ID="myDoc">
  <copyrightHolder rdf:resource="#Institute1"/>
</Document>
<owl:Thing rdf:about="#myDoc">
  <copyrightHolder rdf:resource="#Institute2"/>
</owl:Thing>
```

### Soluzione:

Nessuna; si potrebbe pensare che il vincolo di maxcard=1 sulla proprietà copyrightHolder quando applicata a un Document sia violato dai due istituti Institute1 e Insitute2 legati a myDoc, ma l'assunzione di "No Unique Name" garantisce la consistenza delle ontologie inferendo l'identità dei due istituti.

## Esercizio 4

L'esempio seguente presenta qualche inconsistenza? Se sì, su quale oggetto e perchè?

```
<owl:Class rdf:ID="Institute"/>
<owl:Class rdf:ID="Document">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        1
      </owl:maxCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="copyrightHolder"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="author">
  <rdfs:range rdf:resource="#Institute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#copyrightHolder">
  <rdfs:range rdf:resource="#Institute"/>
</owl:ObjectProperty>
<Institute rdf:ID="Institute1">
  <owl:differentFrom rdf:resource="#Institute2"/>
</Institute>
<Institute rdf:ID="Institute2"/>
<Document rdf:ID="myDoc">
  <copyrightHolder rdf:resource="#Institute1"/>
</Document>
<owl:Thing rdf:about="#myDoc">
  <copyrightHolder rdf:resource="#Institute2"/>
</owl:Thing>
```

### Soluzione:

In questo caso il tag differentFrom applicato alle due risorse Institute1 e Institute2 ci previene dall'unificare i due nomi. Per tale motivo l'ontologia presenta una inconsistenza.

## Esercizio 5

Riclassificare le istanze rispetto alle classi presenti nella ontologia.  
Suggerimento: porre attenzione a Duffy e a MyDoc2.

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Duck"/>
<owl:Class rdf:ID="Document">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#author"/>
      <owl:allValuesFrom rdf:resource="#Person"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<owl:ObjectProperty rdf:ID="author"/>
<Person rdf:ID="Dave"/>
<Document rdf:ID="myDoc">
  <author rdf:resource="#Duffy"/>
</Document>
<owl:Thing rdf:ID="myDoc2">
  <author rdf:resource="#Dave"/>
</owl:Thing>
<Duck rdf:ID="Duffy"/>
```

### Soluzione:

Duffy è (contrariamente a quanto il buon senso suggerirebbe!) una Person. È stato possibile determinare tale relazione osservando la dichiarazione di equivalenza tra la classe degli oggetti con restrizione sulla proprietà author alle sole Person e la classe Document.

Non possiamo invece dire nulla riguardo myDoc2. Se è vero che l'unico suo autore dichiarato è una persona, non possiamo comunque sapere la natura di altre eventuali autori di tale libro. Per tale motivo myDoc2 non può essere ritenuto con certezza un documento.

## Esercizio 6

Dato il seguente frammento di ontologia OWL:

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Animal"/>
<owl:ObjectProperty rdf:ID="is_pet_of">
  <rdfs:range rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_pet">
  <rdfs:range rdf:resource="#Animal"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:Thing rdf:ID="Pete"/>
<owl:Thing rdf:ID="Spike">
  <is_pet_of rdf:resource="#Pete"/>
</owl:Thing>
```

Determinare:

- (se esistono) eventuali inconsistenze
- (se possibile) le classi di appartenenza di Pete e Spike:
- Motivate il vostro ragionamento
- nel caso le classi per uno o entrambi gli individui non siano deducibili, completare l'ontologia di modo da poter assegnare, per deduzione, una classe ad entrambi gli individui, secondo una vostra interpretazione del dominio.

**Soluzione:**

Non esistono inconsistenze nell'ontologia.

*Pete* e *Spike* sono dichiarati come **Thing**, che è la classe generale di owl per non assegnare nessun tipo specifico ad un individuo.

La proprietà *is\_pet\_of* ha tuttavia range **Person**, quindi possiamo inferire che *Pete* è sicuramente una **Person**.

Nulla possiamo dire a proposito di *Spike*, visto che il suo nome è utilizzato solo relativamente alla proprietà *is\_pet\_of* che non ci informa sulla natura del suo dominio.

Possiamo, secondo buon senso, creare una nostra interpretazione del dominio per cui *Spike* è un animale. In questo caso possiamo aggiungere l'informazione:

```
<owl:ObjectProperty rdf:ID="has_pet">
  <owl:inverseOf rdf:resource="#is_pet_of"/>
  <rdfs:range rdf:resource="#Animal"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
```

sotto alla proprietà *has\_pet*. Di conseguenza il range di *has\_pet* (**Animal**) diviene il dominio di *is\_pet\_of*. In questo modo sono state anche relazionate le due proprietà presenti nell'ontologia (che, basandosi su una interpretazione del loro nome, sembrano individuare in effetti la stessa relazione sul dominio).

Una soluzione alternativa è di riportare esplicitamente il dominio di *is\_pet\_of* come **Animal**.

## Esercizio 7

Dato il seguente frammento di ontologia OWL:

```
<owl:Class rdf:ID="Obiettivo">
  <owl:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int">34</owl:hasValue>
      <owl:onProperty rdf:resource="#eta"/>
    </owl:Restriction>
  </owl:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Uomo"/>
<owl:Class rdf:ID="Donna"/>
<owl:FunctionalProperty rdf:ID="eta">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Uomo"/>
</owl:FunctionalProperty>
<owl:ObjectProperty rdf:ID="ama">
  <rdfs:domain rdf:resource="#Uomo"/>
  <rdfs:range rdf:resource="#Donna"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
</owl:ObjectProperty>
<Uomo rdf:ID="Pino">
  <owl:differentFrom rdf:resource="#Mario"/>
  <ama rdf:resource="#Caterina"/>
</Uomo>
<Donna rdf:ID="Caterina"/>
<Donna rdf:ID="Elvira">
  <owl:sameAs rdf:resource="#Caterina"/>
</Donna>
<Uomo rdf:ID="Mario">
  <eta rdf:datatype="http://www.w3.org/2001/XMLSchema#int">34</eta>
  <ama rdf:resource="#Elvira"/>
</Uomo>
```

Determinare:

- (se esistono) eventuali inconsistenze
- le istanze (esplicite o inferibili) della classe **Obiettivo**.
- Motivate il vostro ragionamento

**Soluzione:**

Vi è una inconsistenza: dalle dichiarazioni apprendiamo che *Caterina* ed *Elvira* sono lo stesso individuo (*sameAs* su *Elvira*).

Sappiamo poi che *Pino ama Caterina* e *Mario ama Elvira*; dall'affermazione precedente, sappiamo che quindi amano la stessa persona.

La proprietà *ama* è però *InversaFunzionale*, implica quindi che *Mario* e *Pino* sono la stessa individuo perchè amano lo stesso individuo (*Caterina-Elvira*). Ciò è contraddetto dall'affermazione su *Pino*, dove si riporta un *differentFrom* con *Mario*.

La classe **Obiettivo** è *sottoclasse* della classe rappresentata da tutti gli individui di 34 anni. *Eta* corrispondente a 34 è quindi una condizione necessaria ma non sufficiente per la classe **Obiettivo**. Non possiamo quindi affermare nulla circa eventuali appartenenti a tale classe perché non abbiamo alcuna istanza esplicita di tale classe né condizioni sufficienti di appartenenza.