

Esempio 1

```
<owl:Class rdf:ID="Document"> —————> Document  $\equiv$   $\forall$ author.Person
  <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" />

<Document rdf:ID="doc">
  <author rdf:resource="#pippo" />
</Document>
<owl:Thing rdf:ID="pippo" />
```

Posso inferire:

pippo a Person . (perché pippo è author di un Document, ed un Document ha solo author di tipo Person)

Esempio 2

```
<owl:Class rdf:ID="Document2"> —————> Document2  $\equiv$   $\exists$ author.Person
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#author" />
      <owl:someValuesFrom rdf:resource="#Person" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<owl:ObjectProperty rdf:ID="author" />

<Document2 rdf:ID="doc">
  <author rdf:resource="#pippo" />
</Document2>
<owl:Thing rdf:ID="pippo" />
```

So che doc2 ha un author di tipo Person, ma non so se esso coincide con pippo: per l'ipotesi di mondo aperto doc2 potrebbe avere altri author.

Non si può, quindi, inferire che pippo è di tipo Person.

Esempio 3

`<owl:Class rdf:ID="Document">` \longrightarrow **Document $\equiv \forall \text{author. Person}$**

```
<owl:equivalentClass>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#author" />
    <owl:allValuesFrom rdf:resource="#Person" />
  </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

`<owl:Class rdf:ID="Document2">` \longrightarrow **Document2 $\equiv \exists \text{author. Person}$**

```
<owl:equivalentClass>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#author" />
    <owl:someValuesFrom rdf:resource="#Person" />
  </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

`<owl:Class rdf:ID="Document3">` \longrightarrow **Document3 $\equiv \text{author } \exists \text{ pippo}$**

```
<owl:equivalentClass>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#author" />
    <owl:hasValue rdf:resource="#pippo" />
  </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="author" />
```

```
<Document rdf:ID="doc">
  <author rdf:resource="#pippo" />
</Document>
```

```
<owl:Thing rdf:ID="pippo" />
```

Esempio 3 (continuazione)

```
Document ≡ ∀author.Person  
Document2 ≡ ∃author.Person  
Document3 ≡ author ∃ pippo  
author a owl:ObjectProperty .
```

```
doc a Document .  
doc author pippo .  
pippo a owl:Thing .
```

Posso inferire:

```
pippo a Person . (gli author di un Document sono Person)  
doc a Document3 . (perché pippo è un author di doc)  
Document3 rdfs:subClassOf Document2 .  
(perché chi ha pippo come author ha almeno un author di tipo Person)  
doc a Document2 . (segue dai due risultati precedenti)
```

Esempio 4

```
<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:Class rdf:ID="Document2">
```

```
  <owl:equivalentClass>
```

```
    <owl:Restriction>
```

```
      <owl:onProperty rdf:resource="#author" />
```

```
      <owl:someValuesFrom rdf:resource="#Person" />
```

```
    </owl:Restriction>
```

```
  </owl:equivalentClass>
```

```
</owl:Class>
```

```
<owl:Class rdf:ID="Document3">
```

```
  <owl:equivalentClass>
```

```
    <owl:Restriction>
```

```
      <owl:onProperty rdf:resource="#author" />
```

```
      <owl:hasValue rdf:resource="#pippo" />
```

```
    </owl:Restriction>
```

```
  </owl:equivalentClass>
```

```
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="author" />
```

```
<owl:Thing rdf:ID="doc">
```

```
  <author rdf:resource="#pippo" />
```

```
</owl:Thing >
```

```
<Person rdf:ID="pippo" />
```

Document $\equiv \forall \text{author. Person}$

Document2 $\equiv \exists \text{author. Person}$

Document3 $\equiv \text{author} \ni \text{pippo}$

Esempio 4 (continuazione)

```
Document ≡ ∀author.Person  
Document2 ≡ ∃author.Person  
Document3 ≡ author ∃ pippo  
author a owl:ObjectProperty .
```

```
doc a owl:Thing .  
doc author pippo .  
pippo a Person .
```

Posso inferire:

```
doc a Document2 . (perché ha un author di tipo Person)  
doc a Document3 . (perché ha pippo come author)  
Document3 rdfs:subClassOf Document2  
(perché chi ha pippo come author ha almeno un author che è di tipo Person)
```

Esempio 5

```
<owl:Class rdf:ID="Document2">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#author" />
      <owl:someValuesFrom rdf:resource="#Person" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="author" />

<owl:Thing rdf:ID="doc">
  <author rdf:resource="#pippo" />
</owl:Thing>
<Person rdf:ID="pippo" />
```

non posso inferire che doc è un Document2,
perché avere un author di tipo Person è
condizione necessaria ma non sufficiente di
appartenenza alla classe Document2.

\exists author.Person
Document2

Esempio 6

`<owl:Class rdf:ID="Document">` \longrightarrow **Document \equiv (=2 author)**

```
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource"#author" />
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">2</ow
        l:cardinality>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

```
<owl:Thing rdf:ID="doc">
  <author rdf:resource="#a1" />
  <author rdf:resource="#a2" />
</owl:Thing>
```

Non posso inferire che doc è di tipo Document, perché nulla esclude un terzo author.

Esempio 7

```
<owl:Class rdf:ID="Document">  —————>Document ≡ (≥2 author)
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource"#author" />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">2</owl:
        minCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

```
<owl:Thing rdf:ID="doc">
  <author rdf:resource="#a1" />
  <author rdf:resource="#a2" />
</owl:Thing>
```

Non posso inferire che doc è di tipo Document, perché a1 e a2 potrebbero coincidere (per la *no unique name assumption*).

Esempio 8

```
<owl:Class rdf:ID="Document">           —————>Document ≡ (≥2 author)
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource"#author" />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">2</owl:min
        Cardinality>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
<owl:Thing rdf:ID="doc">
  <author rdf:resource="#a1" />
  <author rdf:resource="#a2" />
</owl:Thing >
<owl:Thing rdf:ID="a1">
  <owl:differentFrom rdf:resource="#a2" />
</owl:Thing>
```

Posso inferire che

doc a Document .