

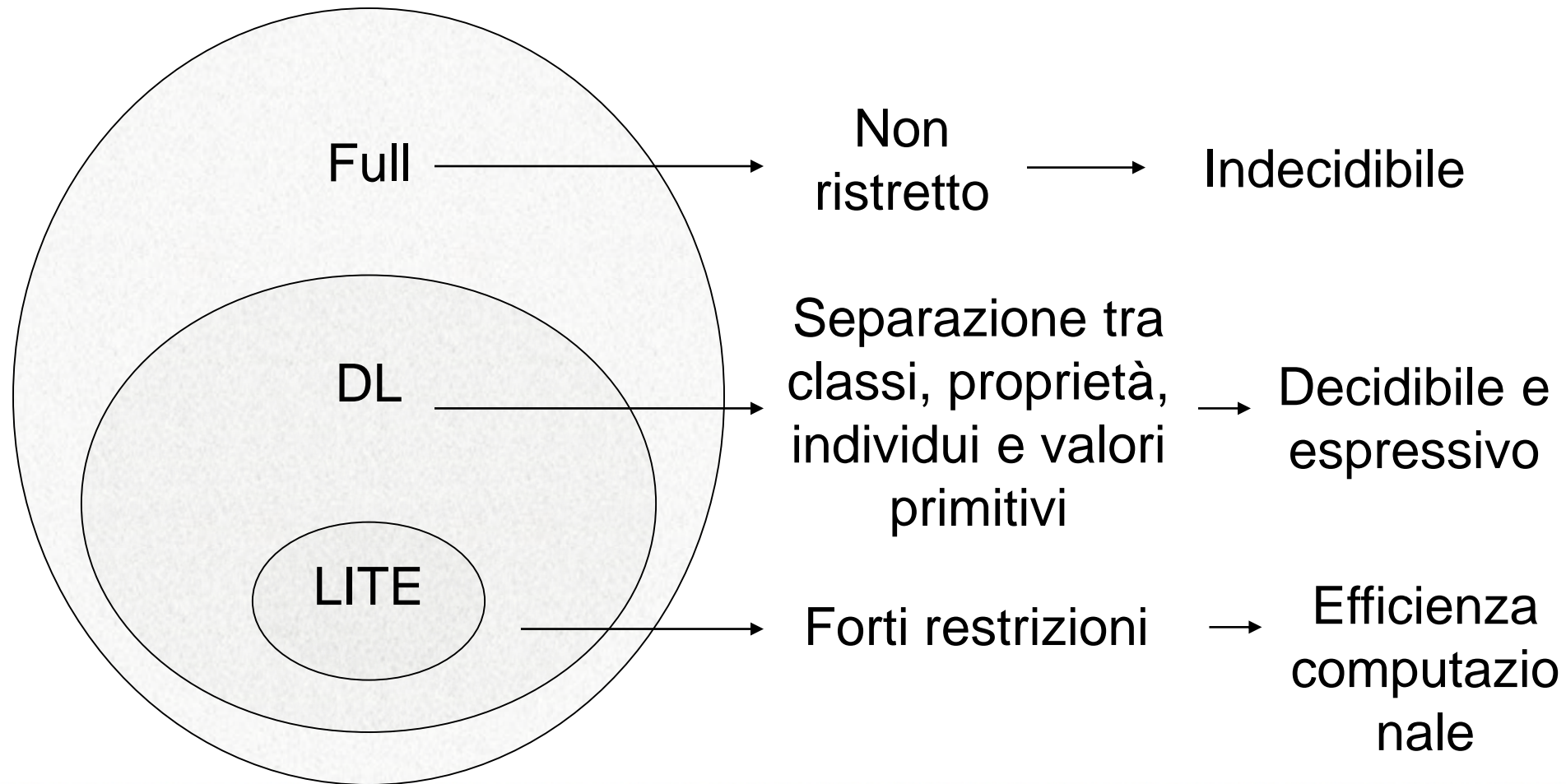
# Brevi appunti su OWL

Andrea Turbati

[turbati@info.uniroma2.it](mailto:turbati@info.uniroma2.it)

# Sottolinguaggi di OWL

Sottolinguaggi definiti attraverso restrizioni sull'uso dei costrutti OWL



Possono essere introdotte semplicemente dandole un nome

```
<owl:Class rdf:ID="Person" />
```

Da usare per descrivere le loro istanze

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

Oppure

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

```
    <rdf:type rdf:resource="#Person" />
```

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

- ***datatype property***, mettono in relazione individui con literal
- ***object property***, mettono in relazione individui
- ***annotation property***, fuori dalla semantica dell'ontologia; per commentare l'ontologia

# Object Property

```
<owl:ObjectProperty rdf:ID="ama">  
  <rdfs:domain rdf:resource="#Persona" />  
  <rdfs:range rdf:resource="#Persona" />  
  <rdfs:subPropertyOf rdf:resource="#conosce" />  
</owl:ObjectProperty>
```

# Datatype Property

```
<owl:DatatypeProperty rdf:ID="nome">  
  <rdfs:domain rdf:resource="#Persona" />  
  <rdfs:range rdf:resource="&xsd:string" />  
</owl:DatatypeProperty>
```

# Descrivere individui

```
<Persona rdf:ID="armando">  
  <conosce rdf:resource="#manuel" />  
  <nome rdf:datatype="&xsd:string">Armando</nome>  
</Persona>  
  
<owl:Thing rdf:ID="manuel">  
  <rdf:type rdf:resource="#Persona" />  
  <nome rdf:datatype="&xsd:string">Manuel</nome>  
</owl:Thing >
```

```
<Persona rdf:ID="armando">  
    <owl:differentFrom rdf:resource="#manuel" />  
    <owl:differentFrom rdf:resource="#andrea" />  
</Persona>  
  
<owl:Thing rdf:ID="manuel">  
    <owl:differentFrom rdf:resource="#andrea" />  
</owl:Thing>  
  
<owl:Thing rdf:ID="andrea" />
```



```
<owl:AllDifferent>
```

```
  <owl:distinctMembers rdf:parseType="Collection">
```

```
    <owl:Thing rdf:about="#armando" />
```

```
    <owl:Thing rdf:about="#manuel" />
```

```
    <owl:Thing rdf:about="#andrea" />
```

```
  </owl:distinctMembers>
```

```
</owl:AllDifferent>
```

Diversi tipi di *descrizione di una classe*

- Un *nome* di classe (URI)
- Una *enumerazione* esaustiva delle sue istanze
- Una *restrizione su una proprietà*
- *Intersezione* di due o più classi
- *Unione* di due o più classi
- Il *complemento* di una classe

---

```
<owl:Class rdf:ID="Human" />
```

In DL

Human

Definiamo una classe dando l'insieme degli individui che appartengono alla sua estensione

```
<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Europe"/>
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#Asia"/>
    <owl:Thing rdf:about="#America"/>
    <owl:Thing rdf:about="#Australia"/>
    <owl:Thing rdf:about="#Antarctica"/>
  </owl:oneOf>
</owl:Class>
```

In DL

{Europe, Africa, Asia, America, Australia, Antarctica}

Definiamo una classe come l'insieme di tutti gli individui che soddisfano certe restrizioni sull'uso di una proprietà.

- Vincolo sul valore
- Vincolo sulla cardinalità

# Restrizione sul valore `owl:allValuesFrom`

Definiamo la classe degli individui tali che tutti i valori di una certa proprietà (`hasParent`) appartengono ad una classe (se object property) o datarange (se datatype property)

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:allValuesFrom rdf:resource="#Human" />  
</owl:Restriction>
```

In DL

$\forall \text{ hasParent . Human}$

# Restrizione sul valore `owl:someValuesFrom`

Definiamo la classe degli individui tali che qualche valore di una certa proprietà (`hasParent`) appartiene ad una classe (se object property) o datarange (se datatype property)

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent" />
  <owl:someValuesFrom rdf:resource="#Physician" />
</owl:Restriction>
```

In DL

$\exists \text{ hasParent . Physician}$

# Restrizione sul valore `owl:hasValue`

Definiamo la classe degli individui tali che una certa proprietà (`hasParent`) ha almeno un valore semanticamente uguale a quello indicato (`clinton`)

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:hasValue rdf:resource="#clinton" />  
</owl:Restriction>
```

In DL

$\text{hasParent} \supseteq \text{clinton}$



# Restrizione sulla cardinalità

## owl:maxCardinality

La classe di tutti gli individui che hanno al massimo N (2) valori semanticamente diversi per una certa proprietà (hasParent)

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:maxCardinality  
rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxCardinality>  
</owl:Restriction>
```

In DL

$\leq 2$  hasParent

# Restrizione sulla cardinalità

## owl:minCardinality

La classe di tutti gli individui che hanno almeno N (2) valori semanticamente diversi per una certa proprietà (hasParent)

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:minCardinality  
rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>  
</owl:Restriction>
```

In DL

$\geq 2$  hasParent

# Restrizione sulla cardinalità

La classe di tutti gli individui che hanno esattamente N (2) valori semanticamente diversi per una certa proprietà (hasParent)

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent" />
  <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:cardinality>
</owl:Restriction>
```

In DL

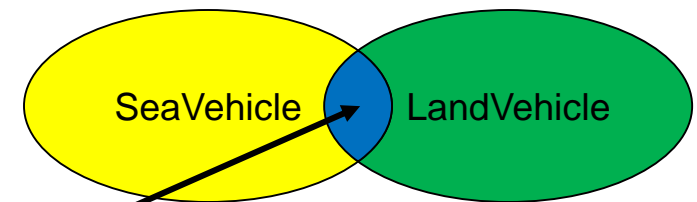
= 2 hasParent

La classe di tutti gli individui che appartengono a tutte le classi indicate (LandVehicle, SeaVehicle)

```
<owl:Class>  
  <owl:intersectionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#LandVehicle" />  
    <owl:Class rdf:about="#SeaVehicle" />  
  </owl:intersectionOf>  
</owl:Class>
```

In DL

$\text{LandVehicle} \sqcap \text{SeaVehicle}$

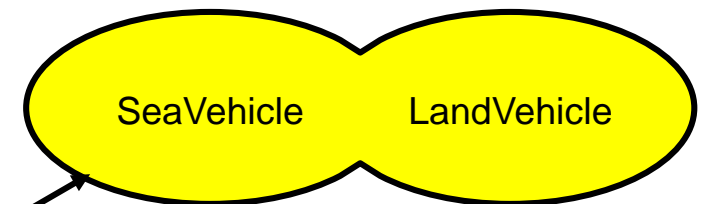


La classe di tutti gli individui che appartengono ad almeno una delle classi indicate (LandVehicle, SeaVehicle)

```
<owl:Class>  
  <owl:unionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#LandVehicle" />  
    <owl:Class rdf:about="#SeaVehicle" />  
  </owl:unionOf>  
</owl:Class>
```

In DL

LandVehicle  $\sqcup$  SeaVehicle

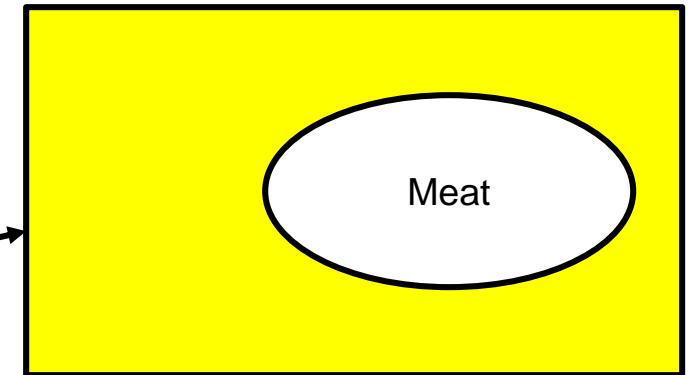


La classe di tutti gli individui che non appartengono ad una certa classe (Meat)

```
<owl:Class>  
  <owl:complementOf>  
    <owl:Class rdf:about="#Meat"/>  
  </owl:complementOf>  
</owl:Class>
```

In DL

$\neg$ Meat



OWL supporta i seguenti tipi di assiomi riguardanti le classi:

- `rdfs:subClassOf`
- `owl:equivalentClass`
- `owl:disjointWith`

# rdfs:subClassOf

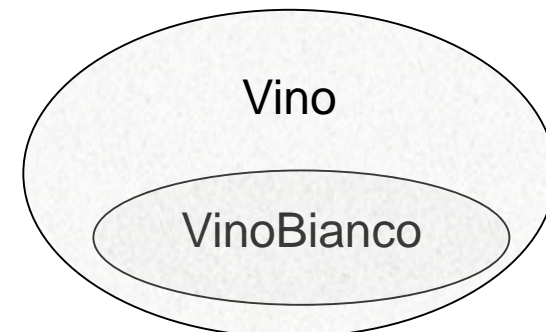
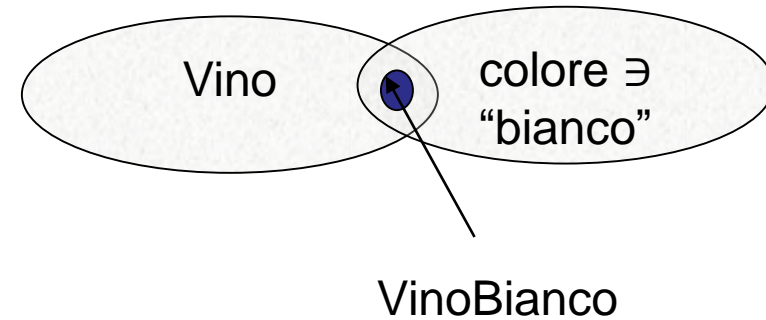
```
<owl:Class rdf:ID="aClass">
  <rdfs:subClassOf>
    class expression
  </rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="aClass">
  <rdfs:subClassOf
rdf:resource="class" />
</owl:Class>
```



```
<owl:Class rdf:ID="VinoBianco">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:intersectionOf parseType="Collection">
        <owl:Class rdf:about="#Vino" />
        <owl:Restriction>
          <owl:onProperty rdf:resource="colore" />
          <owl:hasValue rdf:datatype="&xsd:string">bianco</owl:hasValue>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:id="VinoBianco">
  <rdfs:subClassOf rdf:resource="Vino" />
</owl:Class>
```



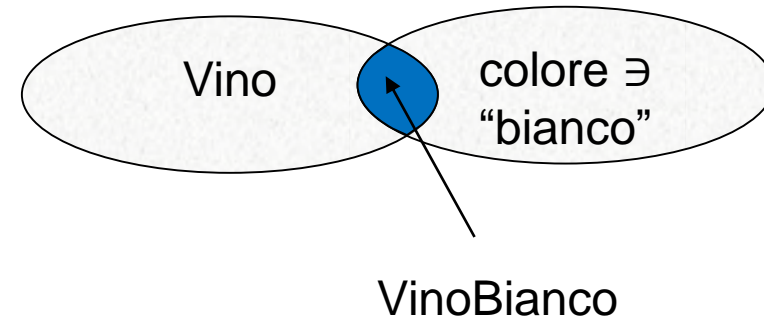
# owl:equivalentClass

```
<owl:Class rdf:ID="aClass">
  <owl:equivalentClass>
    class expression
  </owl:equivalentClass >
</owl:Class>
```

```
<owl:Class rdf:ID="aClass">
  <owl:equivalentClass
rdf:resource="class" />
</owl:Class>
```

# owl:equivalentClass

```
<owl:Class rdf:id="VinoBianco">  
  <owl:equivalentClass>  
    <owl:Class>  
      <owl:intersectionOf parseType="Collection">  
        <owl:Class rdf:about="#Vino" />  
        <owl:Restriction>  
          <owl:onProperty rdf:resource="colore" />  
          <owl:hasValue rdf:datatype="&xsd:string">bianco</owl:hasValue>  
        </owl:Restriction>  
      </owl:intersectionOf>  
    </owl:Class>  
  </owl:equivalentClass>  
</owl:Class>
```



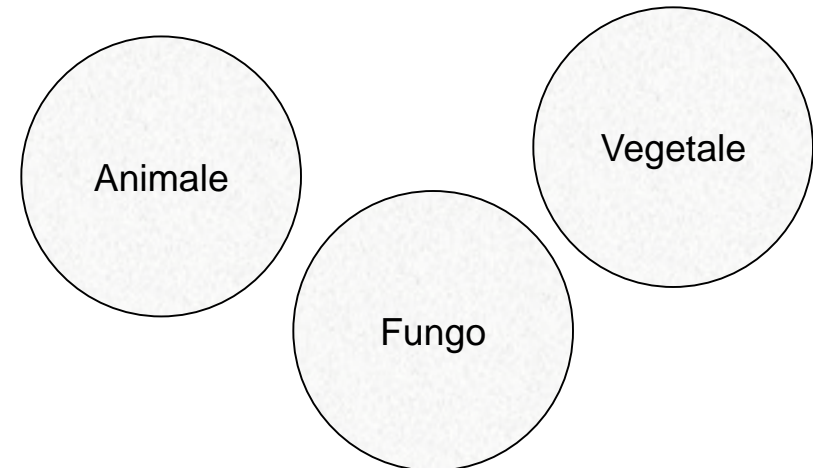
```
<owl:Class rdf:id="VinoBianco">  
  <rdfs:owl:equivalentClass rdf:resource="http://other.com/WhiteWine" />  
</owl:Class>
```

# owl:disjointWith

```
<owl:Class rdf:id="aClass">
  <owl:owlDisjointWith>
    class expression
  </owl:disjointWith>
</owl:Class>
```

```
<owl:Class rdf:id="aClass">
  <owl:disjointWith
rdf:resource="class" />
</owl:Class>
```

```
<owl:Class rdf:ID="Animale">
  <owl:disjointWith rdf:resource="#Vegetale"
/>
  <owl:disjointWith rdf:resource="#Fungo" />
</owl:Class>
<owl:Class rdf:ID="Vegetale">
  <owl:disjointWith rdf:resource="#Fungo" />
</owl:Class>
<owl:Class rdf:ID="Fungo" />
```



Le classi non possono avere  
istanze in comune

Con OWL è possibile esprimere varie caratteristiche delle proprietà:

- owl:TransitiveProperty
- owl:SymmetricProperty
- owl:FunctionalProperty
- owl:inverseOf
- owl:InverseFunctionalProperty

Se una proprietà,  $P$ , è di tipo *transitive* allora per ogni  $x, y, z$ :

$x P y$

$y P z$

\_\_\_\_\_

$x P z$

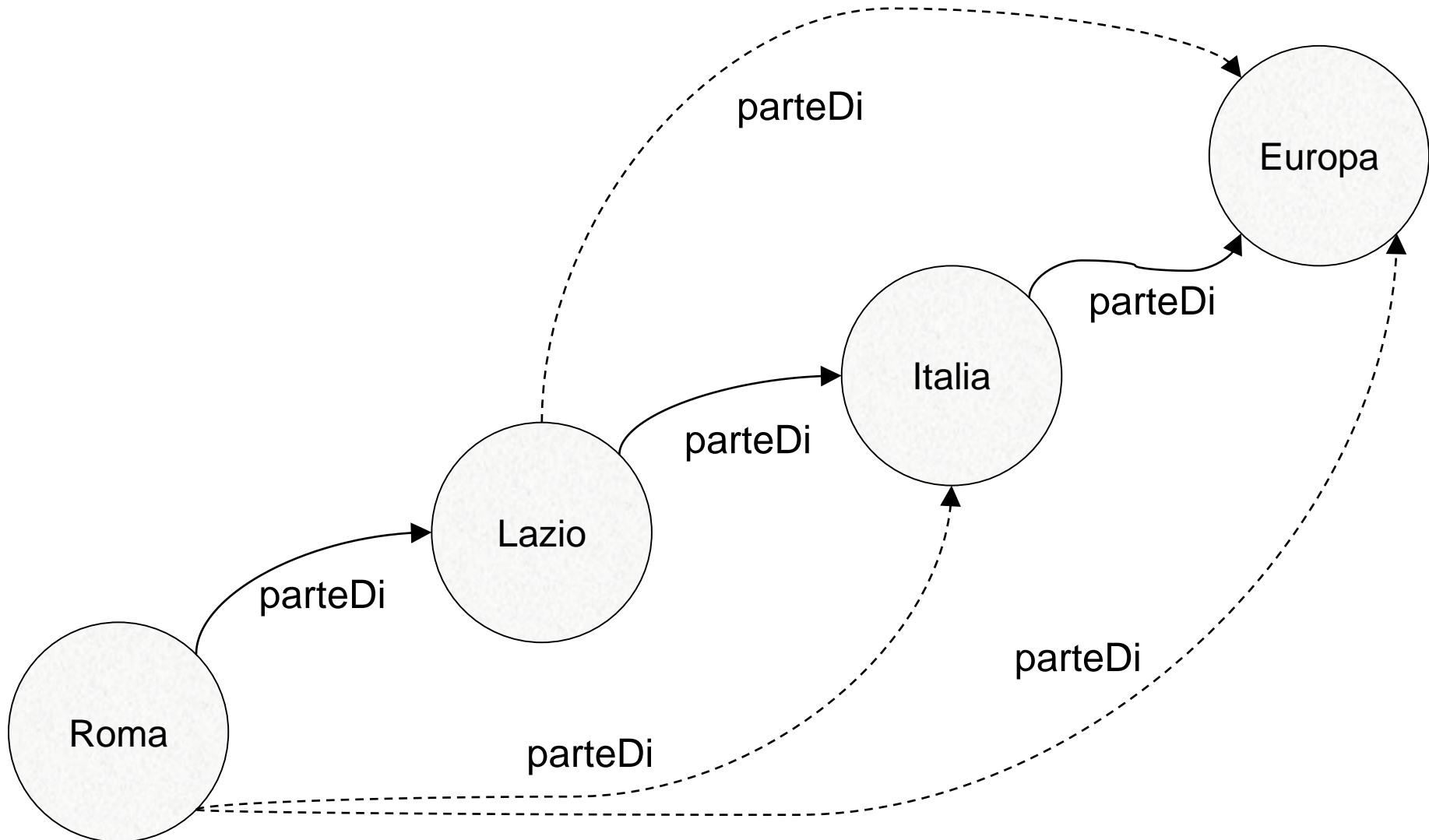
```
<owl:TransitiveProperty rdf:ID="parteDi" />
```

Oppure

```
<owl:ObjectProperty rdf:ID="parteDi" />  
  <rdf:type  
    rdf:resource="&owl;TransitiveProperty" />  
</owl:ObjectProperty>
```



# owl:TransitiveProperty



Se una proprietà,  $P$ , è di tipo *symmetric* allora per ogni  $x$  e  $y$ :

$x P y$

---

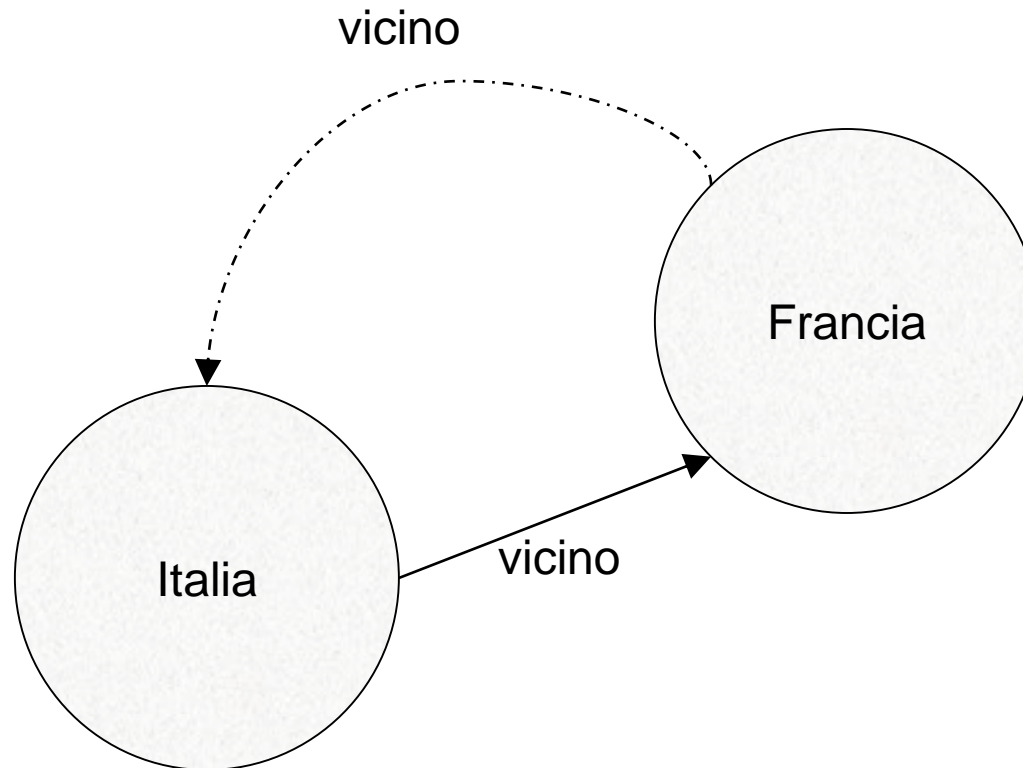
$y P x$

```
<owl:SymmetricProperty rdf:id="vicino" />
```

Oppure

```
<owl:ObjectProperty rdf:id="vicino" />  
  <rdf:type rdf:resource="&owl;  
SymmetricProperty" />  
</owl:ObjectProperty>
```

# owl:SymmetricProperty



Se una proprietà,  $P$ , è di tipo *functional* allora per ogni  $x, y, z$ :

$$x P y$$
$$x P z$$

---

$$y = z$$

```
<owl:FunctionalProperty rdf:ID="produttore" />
```

Oppure

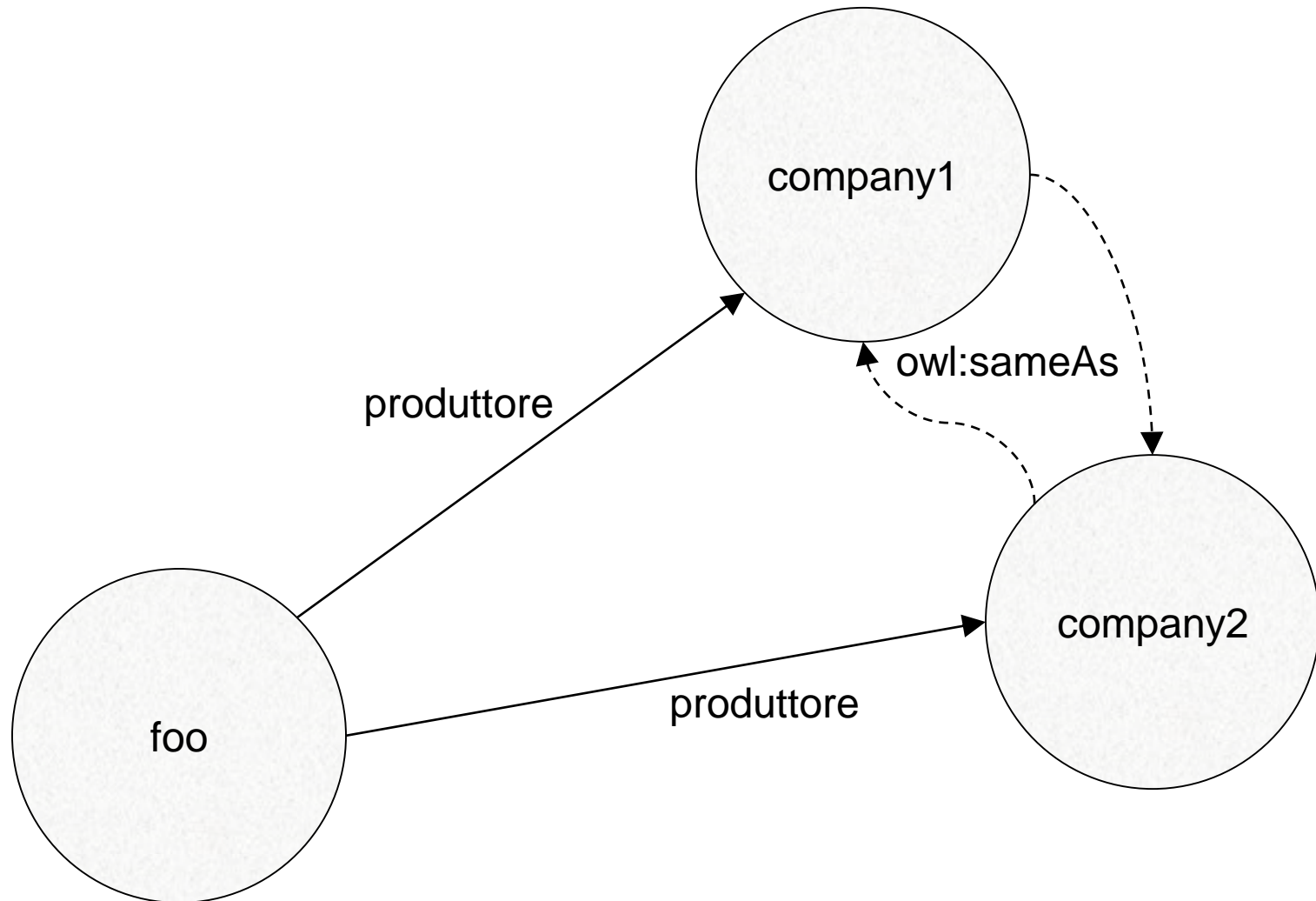
```
<owl:ObjectProperty rdf:ID="produttore"  
/>
```

```
    <rdf:type rdf:resource="&owl;  
FunctionalProperty" />
```

```
</owl:ObjectProperty>
```

Anche datatypeproperty possono essere functional

# owl:FunctionalProperty



X hasChild Y **se e solo se** Y hasParent X

```
<owl:ObjectProperty rdf:ID="hasChild">  
  <owl:inverseOf rdf:resource="#hasParent"/>  
</owl:ObjectProperty>
```



Se una proprietà,  $P$ , è di tipo *InverseFunctional*  
allora per ogni  $x, y, z$ :

$x P z$

$y P z$

\_\_\_\_\_

$x = y$

```
<owl:InverseFunctionalProperty rdf:ID="capitaleDi"  
>
```

Oppure

```
<owl:ObjectProperty rdf:ID="capitaleDi"  
>
```

```
    <rdf:type rdf:resource="&owl;  
InverseFunctionalProperty" />  
</owl:ObjectProperty>
```

In OWL DL: solo object property

# owl:InverseFunctionalProperty

