COMP318

Ontologies and Semantic Web

OWL - Part 3

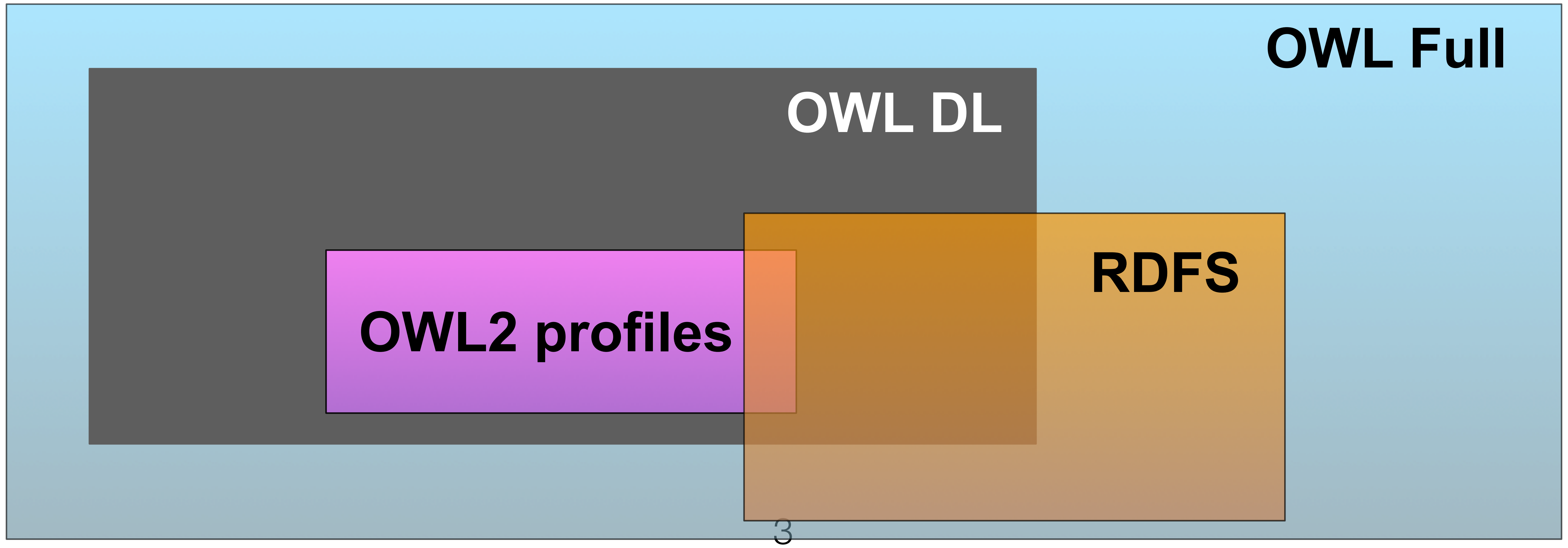
Dr Valentina Tamma
V.Tamma@liverpool.ac.uk

Where were we

- OWL, a KR language for the web
    - OWL extends RDFS

Compatibility between OWL and RDF(S)

- OWL uses to a great extent RDF(S)
  - One of the possible syntax formats for OWL is in RDF/XML
  - instances are declared in RDF
    - using `rdf:Description` and `rdf:type`
  - The OWL constructs `owl:Class`, `owl:DatatypeProperty` and `owl:ObjectProperty` are specialisations of the corresponding RDFS constructs



OWL and DL

- Description logics (DLs) are a family of knowledge representation languages
  - with expressivity typically between propositional logic an first order logic
  - the core reasoning problems for most DLs are usually decidable and the decision procedures implementing them are typically efficient and decidable
- DLs provides a logical formalism for ontologies and the Semantic Web
  - the Web Ontology Language (OWL) and its profiles are based on DLs.
- The most popular application of OWL is in biomedical informatics where DLs are used to encode biomedical knowledge

T-boxes and A-boxes

- DL logical axioms and assertions are stated in T-boxes and A-boxes
  - The T-Box: contain *terminological knowledge* expressing the vocabulary
    - it is independent from the instances
    - It generally consists of subsumption ($\sqsubseteq$) and equivalence ($\equiv$) axioms
      - $\mathtt{LuxuryKitchenApartment} \sqsubseteq \forall\, \mathtt{hasKitchen.LuxuryKitchen}$
      - $\mathtt{LuxuryApartment} \equiv \mathtt{Apartment} \sqcap \forall\, \mathtt{hasKitchen.LuxuryKitchen}$
  - The A-Box: contain *assertional knowledge,* i.e. facts about the instances structured according to the T-box
    - Facts about individuals a, b, c
    - a set of concept membership assertions C(a)
    - and role assertions
      - $\mathtt{LuxuryApartment(BaronWayApartment)}$
      - $\mathtt{locatedIn(BaronWayApartment, Amsterdam)}$

OWL ontology header

OWL namespace

Assertion for housekeeping purposes

```
<rdf:RDF xmlns:owl=”http://www.w3.org/2002/07/owl#”
         xmlns:xsd =”http://www.w3.org/2001/XMLSchema#”
         xmlns:rdf =”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
         xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”
         xml:base=”http://www.example.org">

<owl:Ontology rdf:about=””””>
   <rdfs:comment>An example OWL ontology</rdfs:comment>
   <owl:priorVersion rdf:resource=”http://www.example.org/onto-ns”/>
   <owl:imports rdf:resource=”http://www.example.org/person”/>
   <rdfs:label>Simpson Ontology</rdfs:label>
</owl:Ontology>
...

</rdf:RDF>
```

Namespaces vs import

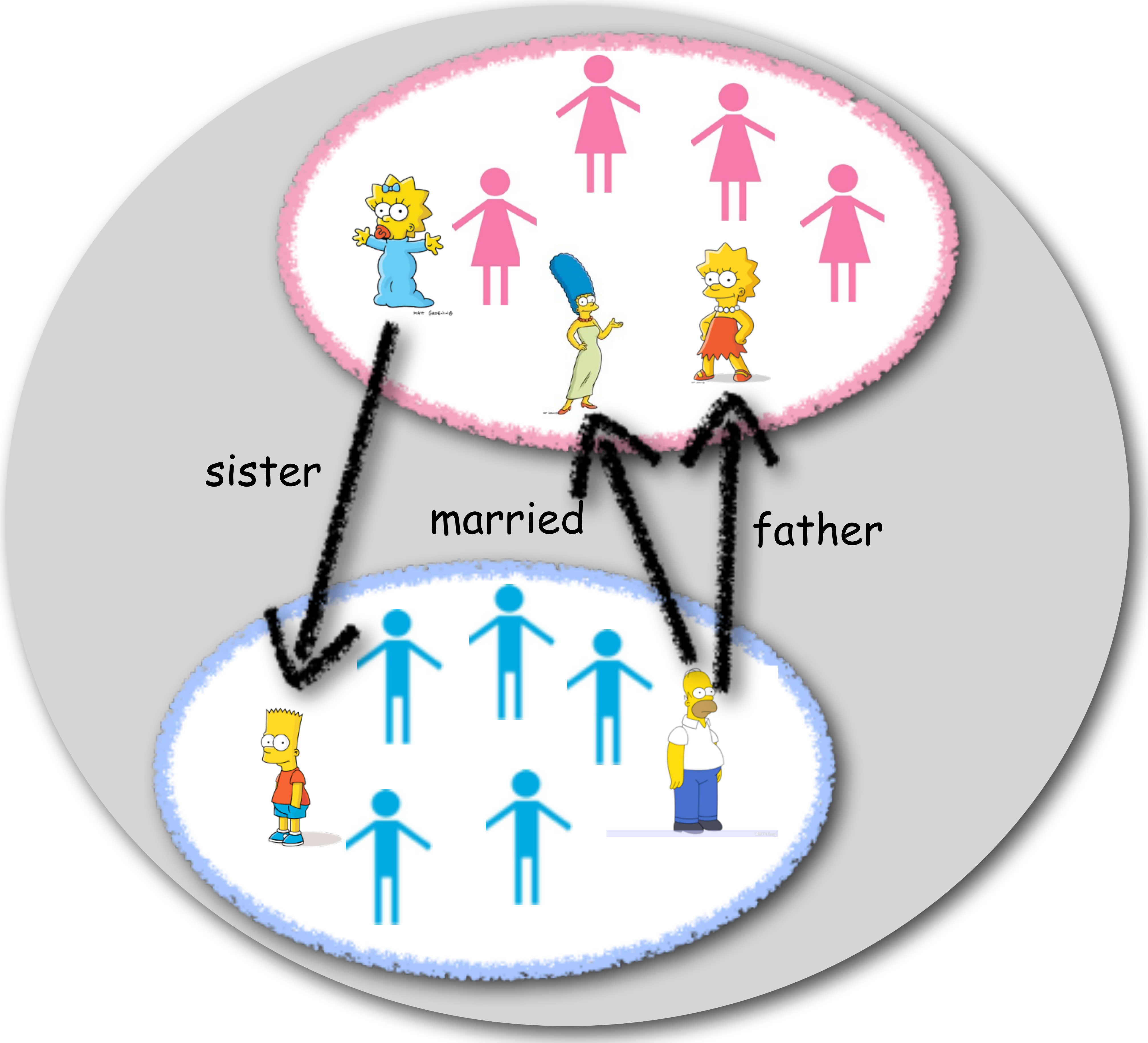
OWL namespace

```
<rdf:RDF xmlns:owl=”http://www.w3.org/2002/07/owl#”
         xmlns:xsd =”http://www.w3.org/2001/XMLSchema#”
         xmlns:rdf =”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
         xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”
         xml:base=”http://www.example.org">

<owl:Ontology rdf:about=””””>
   <rdfs:comment>An example OWL ontology</rdfs:comment>
   <owl:priorVersion rdf:resource=”http://www.example.org/onto-ns”/>
   <owl:imports rdf:resource=”http://www.example.org/person”/>
   <rdfs:label>Simpson Ontology</rdfs:label>
</owl:Ontology>

...

</rdf:RDF>
```

Import of an ontology

What do we describe with OWL

- OWL (we assume DL) ontologies describe a world in terms of:
  - individuals (constants): *homer*, *lisa*, ...
  - classes (unary predicates): *man(x)*, *woman(x)*, *lazy(x)*, *clever(x)*, ...
  - properties/roles (binary predicates): *sister_of(x,y)*, *works_for(x,y)*...



Assertional knowledge (instances)

- Instances assert information about named individuals
- It is restricted to what can be stated in RDF
  - class membership: *female(marge)*
  - property membership: *married(marge, homer)*
    - use **`rdf:ID`** and **`rdf:about`** ***almost*** interchangeably

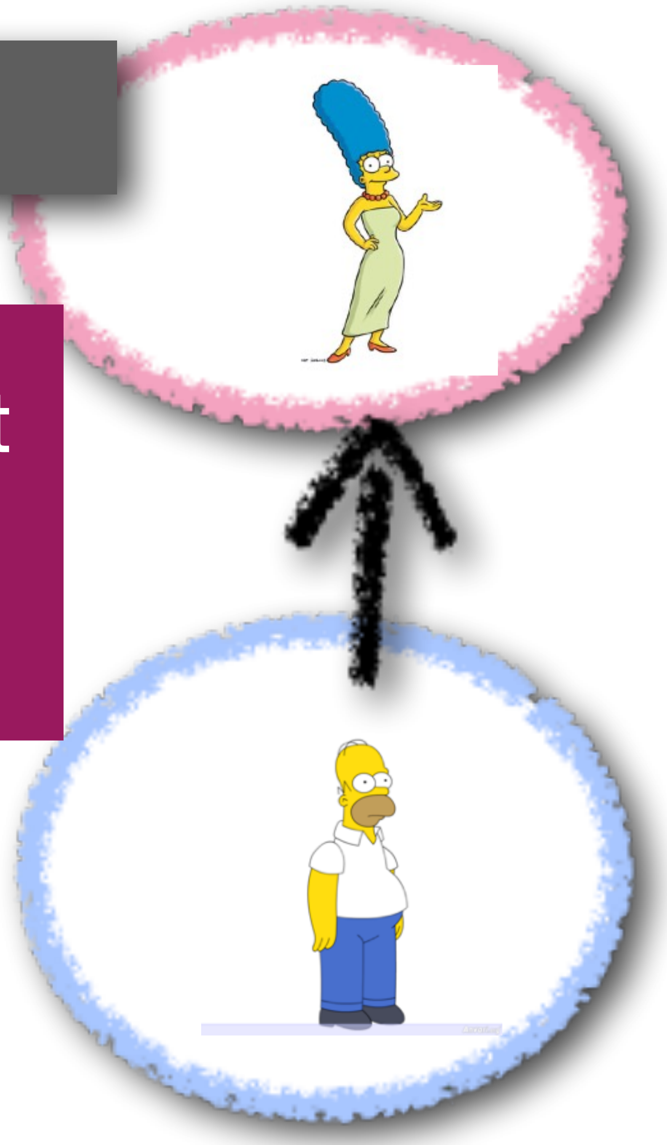
```
<rdf:Description rdf:ID=”marge”>
  <rdf:type rdf:resource=”#woman”/ >
</rdf:Description>
```

It *declares* the individual, it can be used only once in the document

```
<woman rdf:about=”#marge”/>
```

It *references* the individual, it can be used as many times as needed

```
<rdf:Description rdf:about=”marge”>
  <married rdf:resource=”homer”/ >
</rdf:Description>
```



Unique Name Assumption

- In logics with the unique name assumption, different names always refer to different entities in the world.

- Despite being based on description logic, for which UNA holds, OWL does not make this assumption
  - explicit constructs are used to express whether two names refer to the same or different entities
    - `owl:sameAs` - URIs refer to the same entity or individual
    - `owl:differentFrom` - URIs refer to different entities or individual

```
<rdf:Description rdf:about=”#marge”/>
  <owl:sameAs rdf:resource=”#margeSimpson”>
</rdf:Description>

<rdf:Description rdf:about="#homer">
  <owl:differentFrom rdf:resource=“#marge”/>
<rdf:Description>
```

owl:Thing and owl:Nothing

- OWL has two predefined classes
    - `owl:Thing`
        - ⊤ (in DL formalism)
        - class containing all individuals
    - `owl:Nothing`
        - ⊥ (in DL formalism)
        - “empty” class containing no individuals

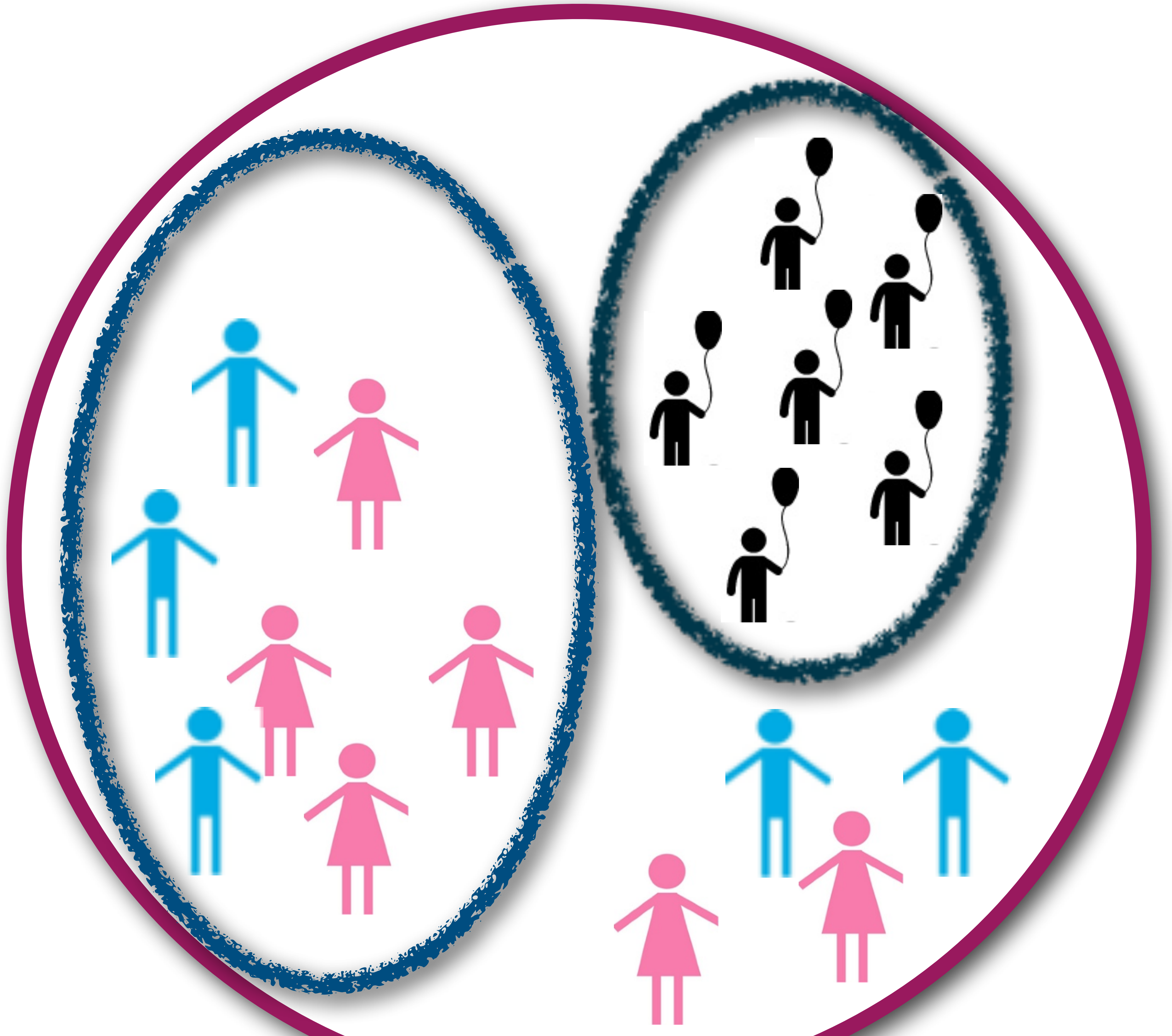
- For every class C
    - `owl:Nothing` is a subclass of C
    - C is a subclass of `owl:Thing`

Terminological knowledge: classes and subclasses

- Classes are defined using owl:class
  - subclass of `rdfs:class`

```
<owl:Class rdf:ID=”parents”>
  <rdfs:subClassOf rdf:resource=”#people”/>
</owl:Class>
```

```
<owl:Class rdf:about=”#children”>
  <owl:disjointWith rdf:resource=”#parents”/>
</owl:Class>
<owl:Class rdf:ID=”offspring”>
  <owl:equivalentClass rdf:resource=”#children”/>
</owl:Class>
```



Recap

- OWL preliminaries
- OWL class constructors

- `https://www.w3.org/TR/owl2-primer/`

COMP318

Ontologies and Semantic Web

End of OWL - Part 3

Dr Valentina Tamma
V.Tamma@liverpool.ac.uk