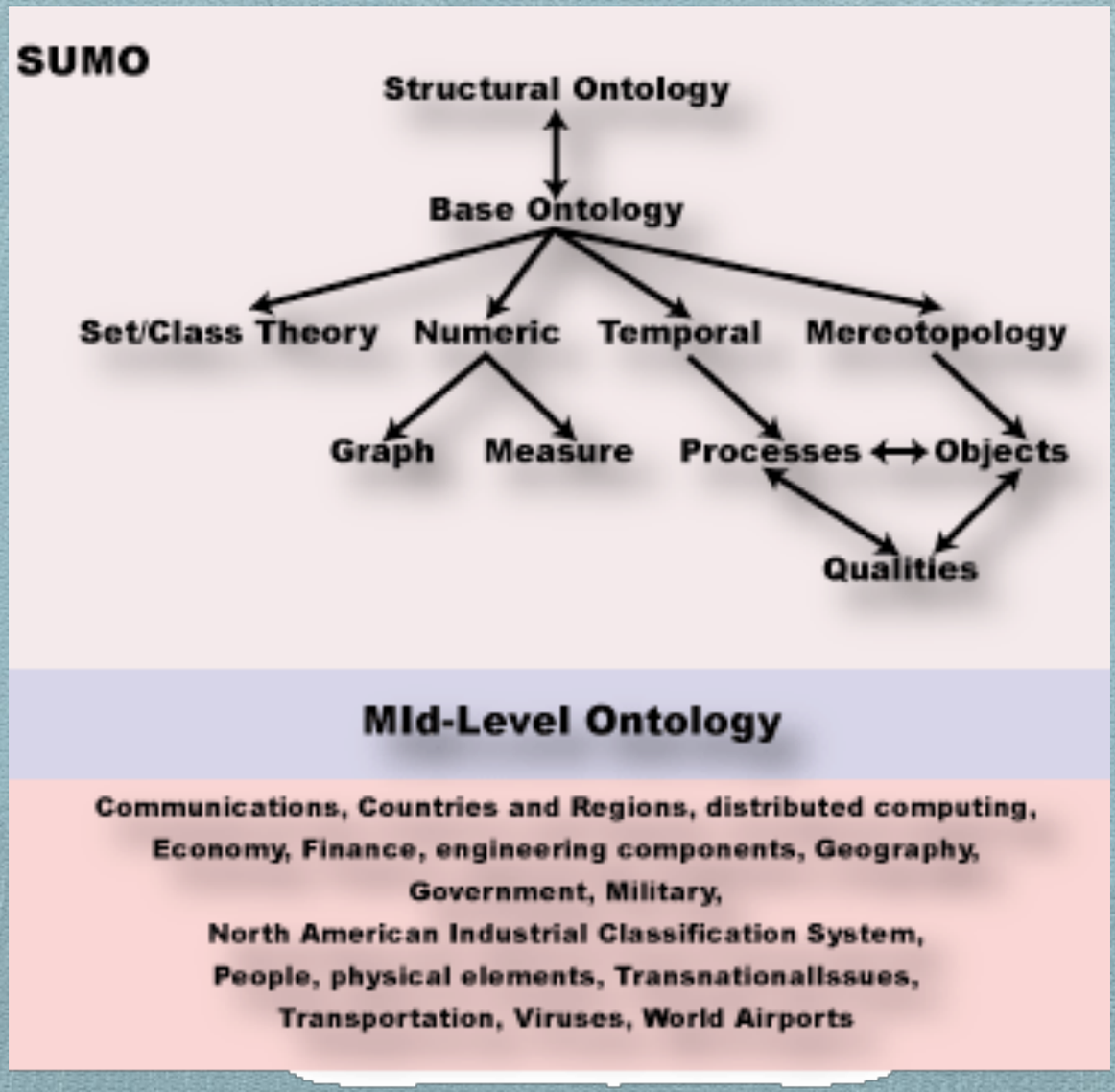


Csaba Veres

Ontologies

Ontology

- Textbook: p. 27 - p. 33
- Gruber
  - “An ontology is an explicit (formal) specification of a conceptualisation.”
- philosophy, where an ontology is a systematic account of existence
- In information science
  - ontology commitment
  - terminological agreement

Making ontologies

- Defining classes
- Arranging the classes in a taxonomic (subclass—superclass) hierarchy
- Defining attributes/relationships and describing allowed values for these.

Ontology Development 101

- Determine the domain and scope of the ontology
  - What is the domain that the ontology will cover?
  - For what we are going to use the ontology?
  - For what types of questions the information in the ontology should provide answers?
  - Who will use and maintain the ontology?

Scope

- Competency questions.
- e.g. wine ontology
- Which wine characteristics should I consider when choosing a wine?
  - Is Bordeaux a red or white wine?
  - Does Cabernet Sauvignon go well with seafood?
  - What is the best choice of wine for grilled meat?
  - Which characteristics of a wine affect its appropriateness for a dish?
  - Does a bouquet or body of a specific wine change with vintage year?
  - What were good vintages for Napa Zinfandel?

Reuse

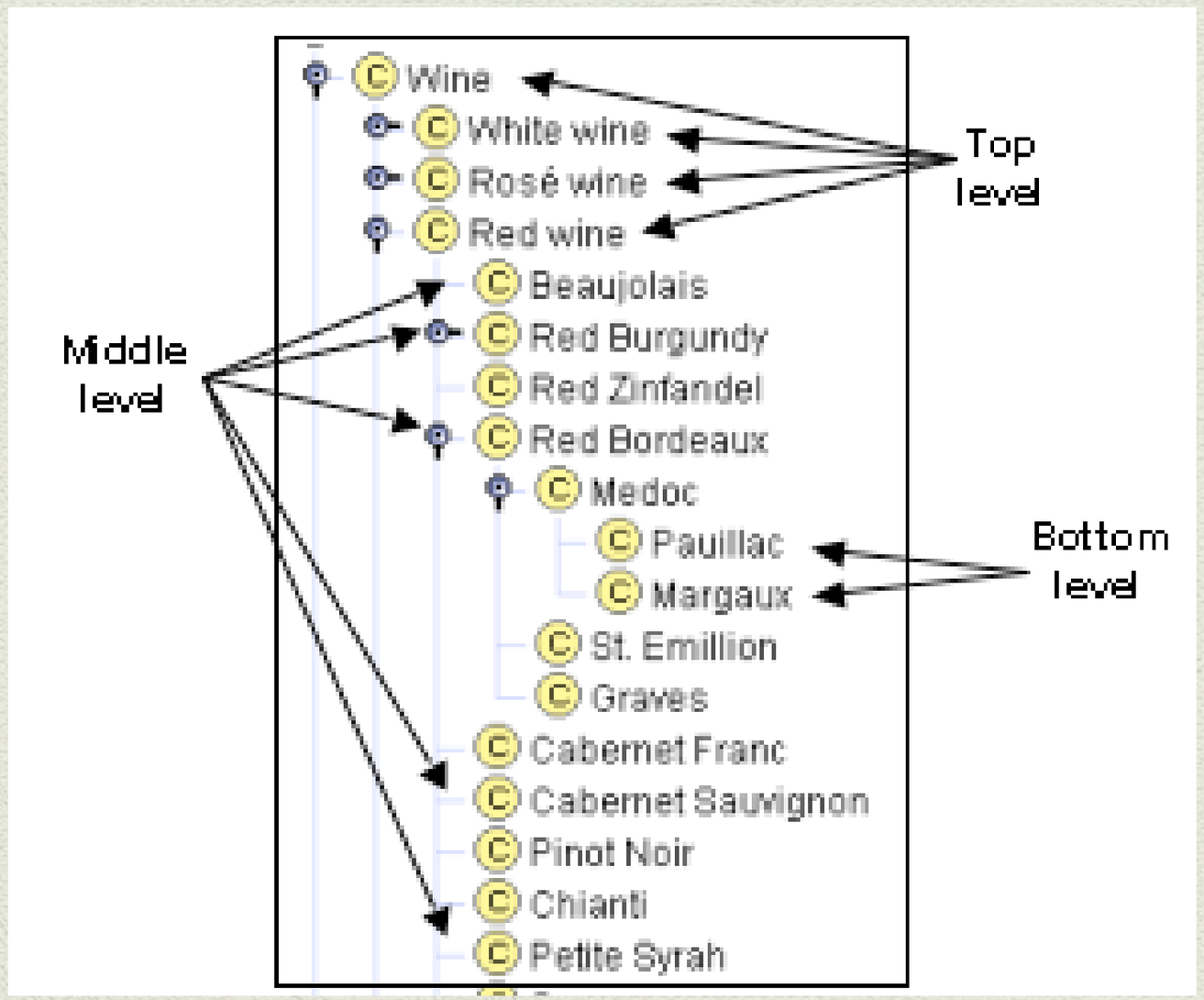
- Consider reusing existing ontologies
  - http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library
  - http://www.daml.org/ontologies/
  - http://onki.fi
  - Google e.g. beer ontology
    - http://www.knoodl.com/ui/groups/Mapping_Ontology_Community/vocab/Beer_Ontology

First Step - List terms

- What are the terms we would like to talk about?
- What properties do those terms have?
- e.g. wine
  - important wine-related terms will include *wine*, *grape*, *winery*, *location*, a wine's *color*, *body*, *flavor* and *sugar content*; different *types of food*, such as *fish* and *red meat*; *subtypes* of wine such as *white wine*, and so on

Define classes

- Top-down
  - danger of starting with too few general concepts
- Bottom-up
  - start with a large set of specific classes (too large?)
- Combination

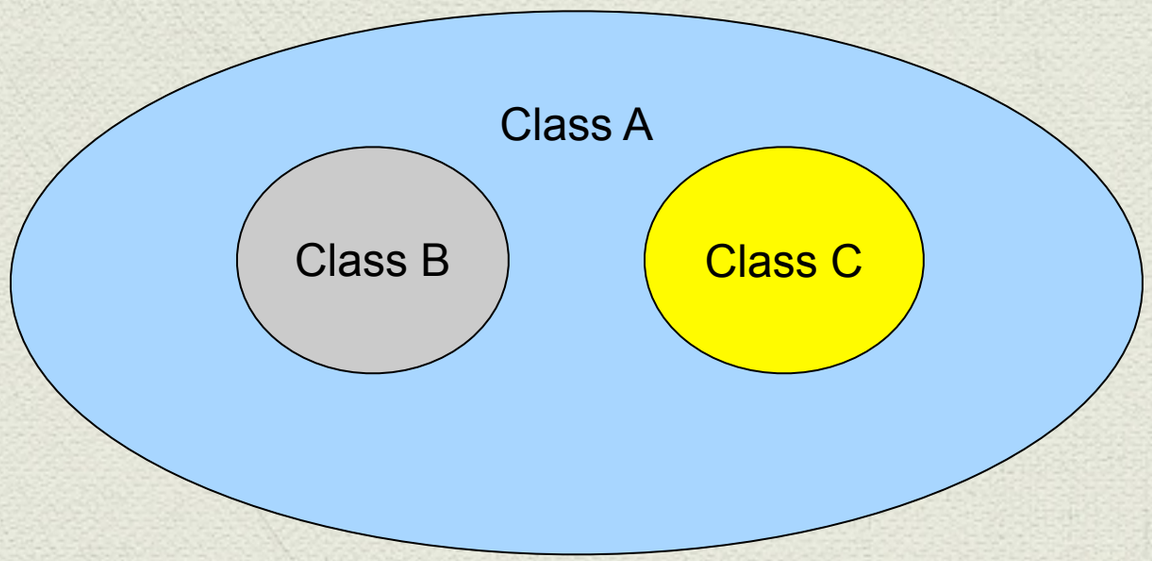


Good classes

- Sortal: (Roughly) the (count) nouns in natural language.
- supplies a principle for distinguishing and counting individual particulars which it collects
  - Substantial: corresponding to types, like person
  - Non-substantial: corresponding to roles, like student
- Non-sortal: (Roughly) adjectives and verbs in natural language.
  - Relationships
- (Mass) Nouns
  - Women, Fire, and Dangerous Things
  - Water, Furniture, ……

Subclass / Superclass

If a class A is a superclass of class B, then every instance of B is also an instance of A



- Class B imply Class A?
- Class C imply Class A?
- Class A imply Class B?

Subclasses

- When do we want new sub classes?
- Subclasses have
  - additional properties
  - participate in different relationships

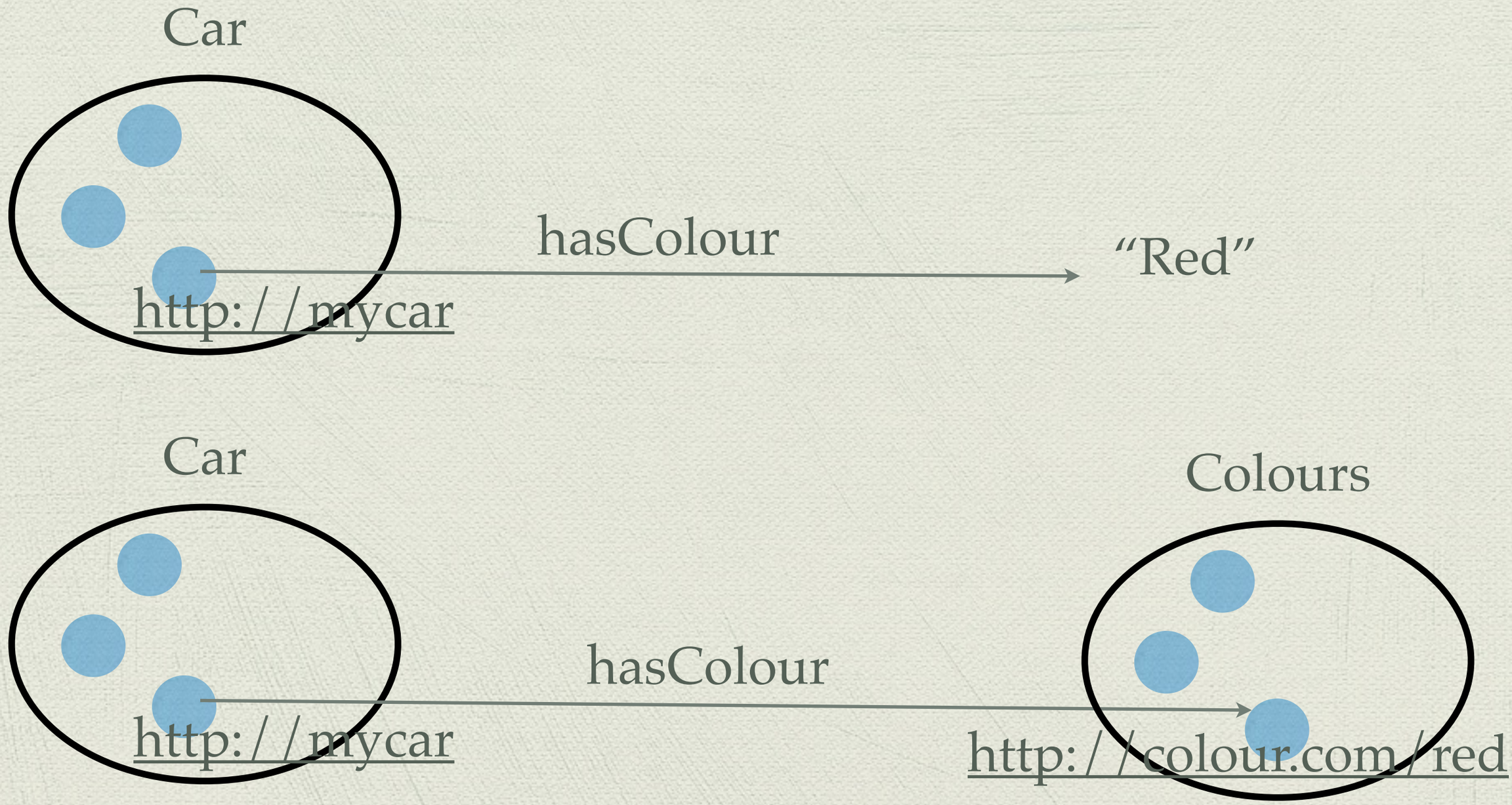
Class properties

- classes alone will not provide enough information to answer the competency questions
- we must describe the internal structure (properties) of concepts
- Properties are *relations between individuals*
- From the wine vocabulary these include color, body, flavour ..

Property values

- What kind of property is it? What kind of values can it have?
- Cardinality
  - min, max, exact
- Type
  - String, Number, Boolean, Enumerated (possible values)
  - Literal value or Instance?
- How do we tell if it should be an instance?

Instance or literal?



Does the range have a “life of its own”?

- Is the property value part of another taxonomy?
- Do you want to be precise about the value?

Class or attribute value?

- Publication
  - publication_type: serial, periodical, single
- OR
  - Periodical_Publication
- Depends:
  - Are there different kinds of Periodical_Publication?
    - newspaper, journal, magazine

Instance vs. Subclass

- Sometimes difficult to decide

  @prefix ex: <http://example.org/>

  ex:Breed rdf:type owl:Class .

  ex:LargeBreed rdf:type owl:Class;

  rdfs:dubClassOf ex:Breed .

ex:SmallBreed rdf:type owl:Class;

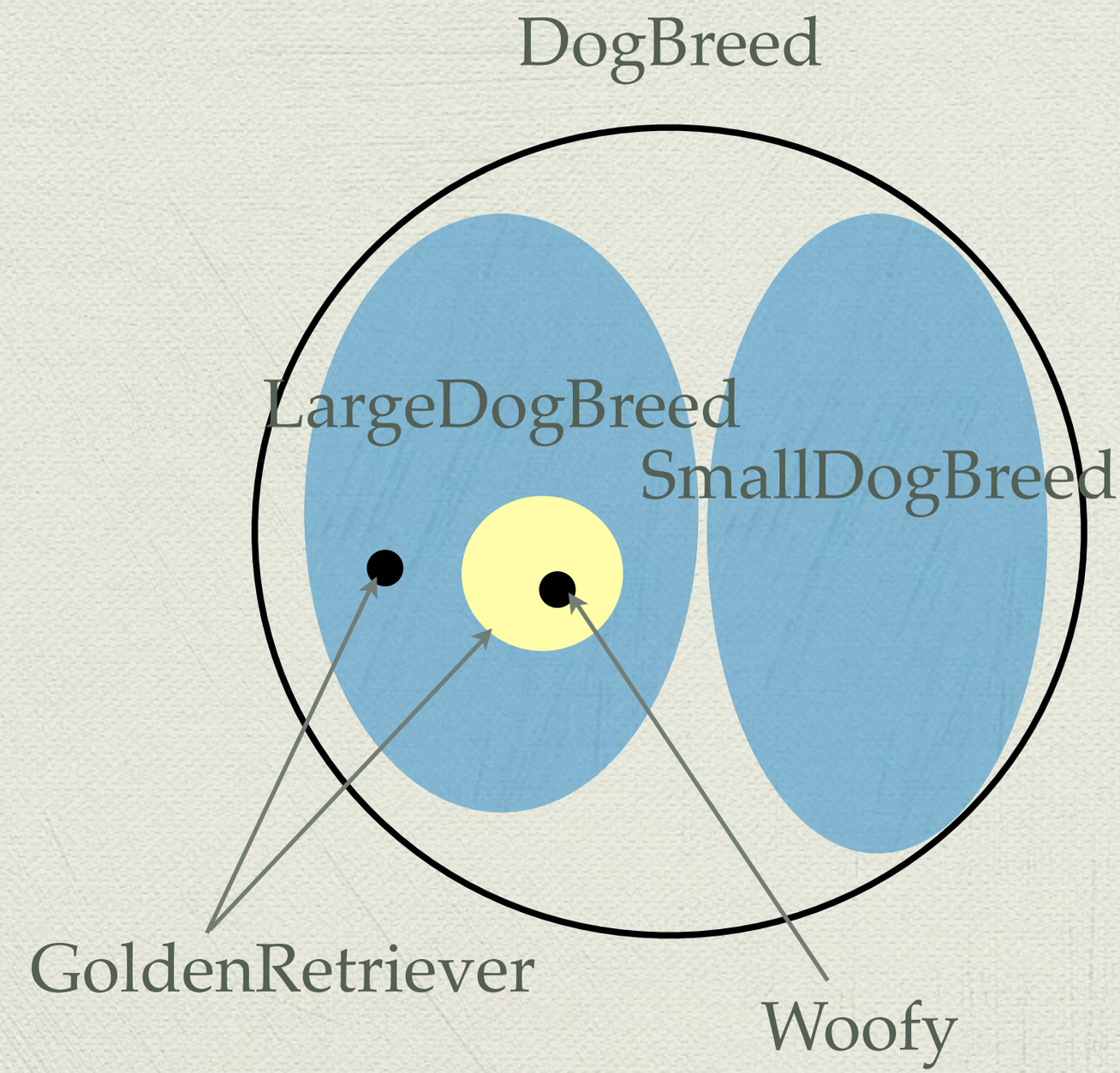
rdfs:dubClassOf ex:Breed .

:GoldenRetriever rdf:type :LargeBreed .

OR

:GoldenRetriever rdf:type owl:Class;

rdfs:subClassOf :LargeBreed .

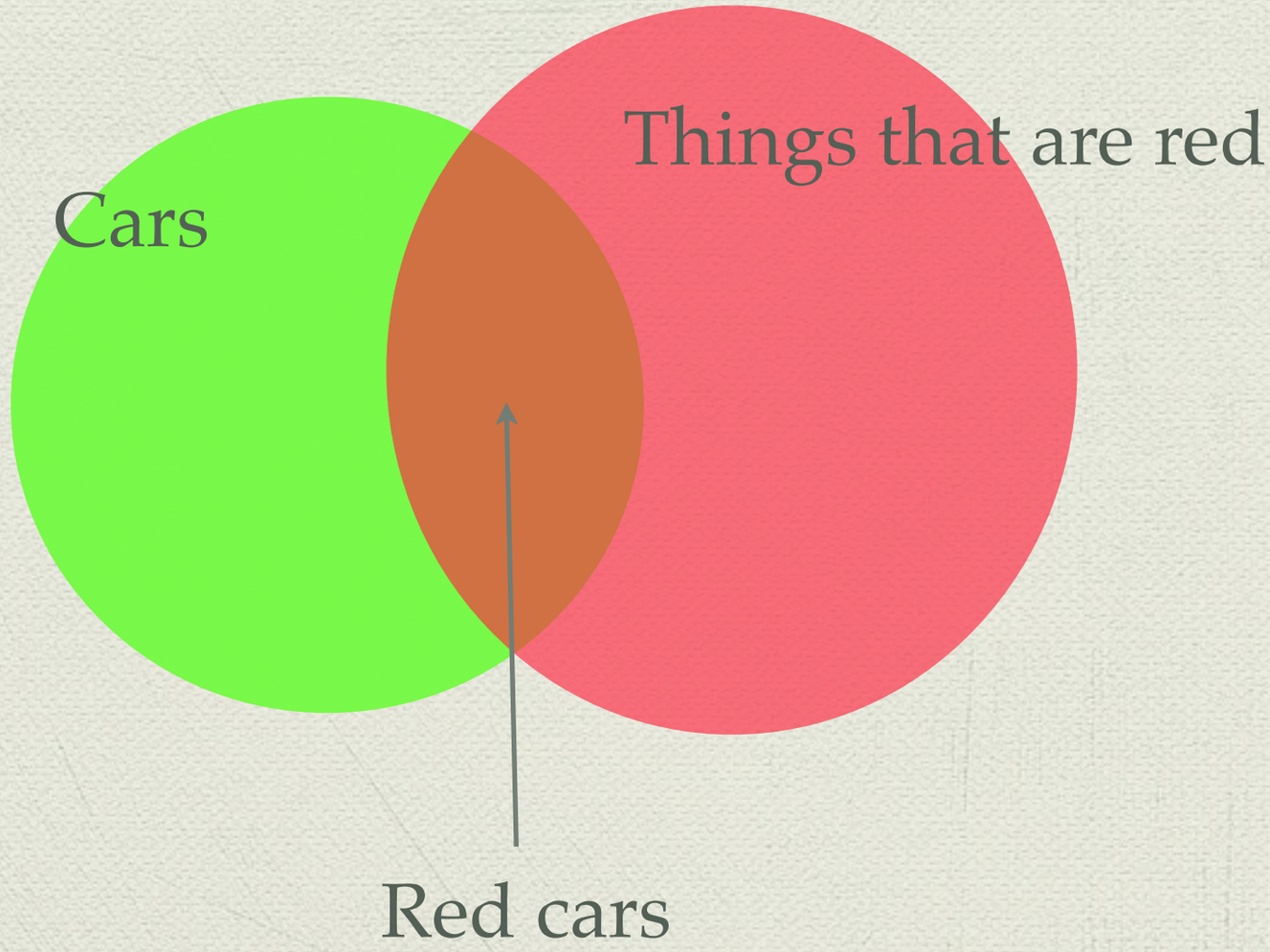


So why do we need an ontology?

- Is Bordeaux a red or white wine?
- Does Cabernet Sauvignon go well with seafood?
- What is the best choice of wine for grilled meat?
- Which characteristics of a wine affect its appropriateness for a dish?
- Does a bouquet or body of a specific wine change with vintage year?
- What were good vintages for Napa Zinfandel?

Some more advanced notions

- Defined classes
- Property restrictions
  - Value restriction
  - Cardinality restriction



All Value restriction

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

Some Value restriction

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

A more complex definition

```
<owl:Class rdf:ID="Operetta">
 <rdfs:subClassOf rdf:resource="#MusicalWork"/>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#hasLibrettist" />
   <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</
owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Class>
   <owl:complementOf rdf:resource="#Opera"/>
  </owl:Class>
 </rdfs:subClassOf>
</owl:Class>
```

Equivalent class

```
<owl:Class rdf:about="#US_President">
  <equivalentClass rdf:resource="#PrincipalResidentOfWhiteHouse"/>
</owl:Class>
```

```
<owl:Class rdf:ID="DaPonteOperaOfMozart">
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <Opera rdf:about="#Nozze_di_Figaro"/>
        <Opera rdf:about="#Don_Giovanni"/>
        <Opera rdf:about="#Cosi_fan_tutte"/>
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Cardinality restrictions

- owl:minCardinality
- owl:maxCardinality
- owl:cardinality

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

Disjoint classes

- owl:disjointWith
- ex:Canine owl:disjointWith ex:Human
- Can solve some problems in inference with open world assumption

Equivalence

- No unique name assumption
- Classes with different names can be equal
  - owl:equivalentClass
- Individuals with different names can be equal
  - owl:sameAs
- Then again, we can't assume that two individuals with different names ARE different:
  - owl:differentFrom