OWL abstract syntax and reasoning examples

Adapted from slides by Raphael Volz, AIFB

OWL Abstract Syntax

- Introduced in OWL Web Ontology Language Semantics and Abstract Syntax
- Useful notation, extended to make practical by
  - http://owl.man.ac.uk/2003/concrete/latest/
- Uses a kind of functional notation, e.g.
  - Class(pp:duck partial pp:animal)
  - ObjectProperty(pp:has_pet
    domain(pp:person)
    range(pp:animal))
  - Individual(pp:Walt
    value(pp:has_pet pp:Huey)
    value(pp:has_pet pp:Louie)
    value(pp:has_pet pp:Dewey))

Namespaces

- Namespace(pp = <http://cohse.semanticweb.org/ontologies/people#>)

Declaring classes in OWL

- *Naming a new class “plant”:*
  Class(pp:plant partial)
- *Naming some “special plants”:*
  Class(pp:grass partial pp:plant)
  Class(pp:tree partial pp:plant)
- *Alternative Declaration:*
  Class(pp:grass partial)
  Class(pp:tree partial)
  SubClassOf(pp:grass pp:plant)
  SubClassOf(pp:tree pp:plant)

Declaring Properties in OWL: I

- *A simple property:*
  ```
  ObjectProperty(pp:eaten_by)
  ```
- *Properties may be inverse to each other:*
  ```
  ObjectProperty(pp:eats
  inverseOf(pp:eaten_by))
  ```
- *Domain and Ranges:*
  ```
  ObjectProperty(pp:has_pet
                      domain(pp:person)
                      range(pp:animal))
  ```

Declaring Properties in OWL: II

- *Datatype Properties:*
  ```
  DataProperty(pp:service_number
  range(xsd:integer))
  ```
- *Property Hierarchy:*
  ```
  SubPropertyOf(pp:has_pet pp:likes)
  ```
- *Algebraic properties:*
  ```
  ObjectProperty(pp:married_to Symmetric)
  ObjectProperty(pp:ancestor_of Transitive)
  ObjectProperty(pp:passport_nr Functional)
  ```

Individuals in OWL

```
Individual(pp:Tom type(owl:Person))
Individual(pp:Dewey type(pp:duck))
Individual(pp:Rex type(pp:dog)
             value(pp:is_pet_of pp:Mick))
Individual(pp:Mick type(pp:male)
             value(pp:reads pp:Daily_Mirror)
             value(pp:drives pp:Golf_GTI)
             value(pp:name „Mick"^xsd:string))
```

Entailment Quiz

What follows from
these descriptions?

Quiz # 1

Class(pp:old+lady complete
  intersectionOf(pp:elderly pp:female pp:person))

Class(pp:old+lady partial intersectionOf(
  restriction(pp:has_pet allValuesFrom(pp:cat))
  restriction(pp:has_pet
  someValuesFrom(pp:animal))))

Quiz #1 - Solution

Every old lady must have a pet cat.

(Because she must have some pet and all her pets must be cats.)

Quiz #2

Class(pp:cow partial pp:vegetarian)

Class(pp:mad+cow complete
  intersectionOf(pp:cow restriction(pp:eats
  someValuesFrom(intersectionOf(pp:brain
  restriction(pp:part_of someValuesFrom
  pp:sheep))))))

What can be said about mad cows ?

Quiz # 2 - Solution

There can be no mad cows.

(Because cows, as vegetarians, don’t eat anything that is a part of an animal.)

Quiz #3

What are Minnie and Tom ?

ObjectProperty(pp:has_pet domain(pp:person)
   range(pp:animal))

Class(pp:old+lady complete
   intersectionOf(pp:elderly pp:female pp:person))

Class(pp:old+lady partial
   intersectionOf(restriction(pp:has_pet
   allValuesFrom(pp:cat)) restriction(pp:has_pet
   someValuesFrom(pp:animal))))

Individual(pp:Minnie type(pp:elderly)
   type(pp:female) value(pp:has_pet pp:Tom))

Quiz #3 - Solution

Minnie must be a person (because pet owners are human) and thus is an old lady. Thus Tom must be a cat (because all pets of old ladies are cats).

Quiz #4

Class(pp:animal+lover complete
   intersectionOf(pp:person
   restriction(pp:has_pet minCardinality(3))))

Individual(pp:Walt type(pp:person)
   value(pp:has_pet pp:Huey)
   value(pp:has_pet pp:Louie)
   value(pp:has_pet pp:Dewey))

DifferentIndividuals(pp:Huey pp:Louie
   pp:Dewey)

What is Walt ?

Quiz #4 - Solution

Walt must be an animal lover. Note that stating that Walt is a person is redundant.

Quiz #5

What are Mick and the Daily Mirror ?

Class(pp:van partial pp:vehicle)

Class(pp:driver partial pp:adult)

Class(pp:driver complete intersectionOf(restriction(pp:drives someValuesFrom(pp:vehicle)) pp:person))

Class(pp:white+van+man complete intersectionOf(pp:man restriction(pp:drives someValuesFrom(intersectionOf(pp:white+thing pp:van)))))

Class(pp:white+van+man partial restriction(pp:reads allValuesFrom pp:tabloid))

Individual(pp:Q123+ABC type(pp:white+thing) type(pp:van))

Individual(pp:Mick type(pp:male) value(pp:reads pp:Daily+Mirror) value(pp:drives pp:Q123+ABC))

Quiz #5 - Solution

Mick drives a white van, so he must be an adult (because all drivers are adults). As Mick is male, thus he is a white van man, so any paper he reads must be a tabloid, thus the Daily Mirror is a tabloid.