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, see here for examples
- 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#>)
Description logics reason with definitions
- They prefer to have complete descriptions
- A complete definition includes both necessary conditions and sufficient conditions

Often impractical or impossible, e.g. natural kinds

Primitive definition is partial or incomplete
- Limits classification that can be done automatically

Example:
- Primitive: a Person
- Defined: Parent = Person with at least one child
Partial and complete definitions in Owl

- Partial definitions typically made using one or more `rdfs:subClassOf` relations
  - `:Parent rdfs:subClassOf :Person`
  - Knowing that John is a parent, it is necessary that he is a person

- Complete definitions are made with `owl:equivalentClass`
  - `:Parent owl:equivalentClass [a owl:intersection (:Person [owl:restriction ...]))]
  - Knowing that John is a person and has a child is sufficient to conclude he is a parent
Definition vs. Assertion

- A **definition** is used to describe *intrinsic* properties of an object. The parts of a description have meaning as a part of a composite description of an object.

- An **assertion** is used to describe an *incidental* property of an object. Asserted facts have meaning on their own.

- Example: “a black telephone”
  Could be either a description or an assertion, depending on the meaning and import of “blackness” on the concept telephone.
Definition versus Assertion

- In English, “a black telephone” is ambiguous
  (1) A black telephone is a common sight in an office
  (2) A black telephone is on the corner of my desk
- KR languages should not be ambiguous so typically distinguish between descriptions of classes and descriptions of individuals
- KR languages often also allow additional assertions to be made that are not part of the definition (In OWL called annotation properties)
Classification is very useful

- Classification is a powerful kind of reasoning that is very useful
- Many expert systems can be usefully thought of as doing “heuristic classification”
- Logical classification over structured descriptions and individuals is also quite useful.
- But… can classification ever deduce something about an individual other than what classes it belongs to?
- And what does that tell us?
Incidental properties

- If we allow incidental properties (e.g., ones that don’t participate in the description mechanism) then these can be deduced via classification.
- This is the purpose of owl’s annotationProperty.
- An annotationProperty can be associated with a definition (partial or complete).
- It is not checked when reasoning about subsumption or instance checking.
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)
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:NYPost)
  value(pp:drives pp:Fiat_500)
  value(pp:name "Mick"^^xsd:string))
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)))))
Every old lady must have a pet cat.

(Because she must have some pet and all her pets must be cats.)
What can be said about mad cows?
There can be no mad cows.

(Because cows, as vegetarians, don’t eat anything that is a part of an animal.)
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))
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?
Walt must be an animal lover. Note that stating that Walt is a person is redundant.
Quiz #5

What are Mick and the National Enquirer?

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:National_Enquirer) value(pp:drives pp:Q123+ABC))
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 National Enquirer is a tabloid.