

# More OWL

Read

- Semantic Web Programming: chapter 4, 5
- Foundations of Semantic Web Technologies: chapter 4, 5.

## 1 Exercises from lecture

1. What do we mean when we say ABox, TBox, and Rbox axioms
2. What is the difference between unique name assumption and non-unique name assumption?
3. Why do OWL make a non-unique name assumption?

## 2 The Simpsons in OWL 2

In this exercise we will extend the family ontology by using OWL 2 vocabulary. OWL 2 adds expressivity to OWL DL without becoming undecidable. For an overview of the new features of OWL 2 see OWL 2 Web Ontology Language: New Features and Rationale.

### 2.1 Exercise

Create a new ontology which imports the family OWL ontology we made in last week's exercises.

### 2.2 Exercise

State that a person have exactly one gender.

### 2.3 Exercise

Explain what owl keys are.

How would you state that `foaf:name` is key for `Person`?

## 2.4 Exercise

Identify the correct characteristics of all properties in the ontology, i.e., which properties are

- asymmetric
- reflexive
- irreflexive

Explain what asymmetric, reflexive and irreflexive properties are. Use examples.

### 2.4.1 Tip

Note that the exercise says "identify" and not "add". The reason for this is that not all combinations of property characteristics are admissible in OWL 2. Adding more property characteristics carelessly will cause reasoners to report errors. For more information see the paper *The even more irresistible SROIQ* by Horrocks et al. SROIQ is the description logic which is the logical basis for OWL 2.

## 2.5 Exercise

Explain what it means for two properties to be disjoint. Add the natural disjoint axioms for properties in the ontology.

## 2.6 Exercise

Explain what property chaining is. Using property chaining, define the properties

- `fam:hasAunt`
- `fam:hasGrandParent`
- `fam:hasUncle`

as equivalent to the correct property chain.

Create two new properties

- `:hasGrandMother`
- `:hasGrandFather`

in the same manner.

## 2.7 Exercise

Define a class **Minor** as a **Person** which is under the age of 18.

Define a class **Juvenile** as a **Person** which is under the age of 16.

## 2.8 Exercise

Add a class **Job** and a object property **hasJob** with domain **Person** and range **Job**.

Define a class **Retiree** as a **Person** which is over 66 years old and has no **Job**.