It is useful to keep in mind that OWL2 is essentially a language for describing *sets* of things. These sets are called 'classes.' Any statement we make about a class in OWL2 is meant to differentiate that class from the set of *all* things.

4.4.1 Syntax

OWL2 builds on RDF and RDF Schema and thus can be expressed using all valid RDF syntaxes. However, many syntaxes exist for OWL2, each of which has its own benefits and drawbacks:

Functional-Style Syntax This syntax closely relates to the formal structure of ontologies. It is used in the language specification document, in the definitions of the semantics of OWL2 ontologies, the mappings from and into RDF syntaxes, and the different profiles of OWL2. It is much more compact and readable than many of the other syntaxes. For instance, the above class restriction can be written in this syntax as:

```
EquivalentClasses( :Person ObjectUnionOf( :Man :Woman ) )
```

OWL/XML This is an XML syntax for OWL2 that does not follow the RDF conventions, but closely maps onto the functional-style syntax.[3] The main benefit of this syntax is that it allows us to interact with ontologies using standard off-the-shelf XML authoring tools. For example, the OWL/XML syntax of the equivalent class axiom is:

```
<EquivalentClasses>
  <Class abbreviatedIRI=":Person"/>
  <ObjectUnionOf>
    <Class IRI="#Man"/>
```

[3] This OWL/XML serialization is defined in http://www.w3.org/TR/owl-xml-serialization/. It should not be confused with the older OWL/XML presentation syntax (defined in http://www.w3.org/TR/owl-xmlsyntax), which was based on the Abstract Syntax of the first version of OWL.