OWL 2 DL

First Part

Valentina Bartalesi Lenzi

Istituto di Scienza e Tecnologie dell'Informazione - Consiglio Nazionale delle Ricerche – Pisa

A.A. 2024-2025

OWL 2 DL Ontologies

An OWL 2 DL ontology is a **formal description of a domain of interest**

OWL 2 DL ontologies consist of **three different syntactic categories**:

1. **Entities**: classes, properties, and individuals, identified by IRIs. They form the primitive terms and the basic elements of an ontology
2. **Expressions**: complex notions capturing the intensions of classes and properties
3. **Axioms**: statements that are asserted to be true

OWL reserves some IRIs, written using the classical prefixes:

rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs: <http://www.w3.org/2000/01/rdf-schema#>
xsd: <http://www.w3.org/2001/XMLSchema#>
owl: <http://www.w3.org/2002/07/owl#>

OWL 2 DL Ontologies

An **ontology** is a **resource identified by an IRI** with an **optional version**

Logically, an ontology consists of a **set of axioms**

An ontology is associated with an **ontology document**, which physically contains the ontology

The ontology document should be accessible via the **ontology IRI** (if any), or via the **ontology version** (if any)

Import Closure

For modularization, an ontology can **import** other ontologies, specified via their document IRIs

The import closure of an ontology O is a set containing O and all the ontologies that O imports

The import closure of O should not contain ontologies O1 and O2 such that:

- O1 and O2 are **different ontology versions** from the same ontology series, or
- O1 contains an ontology annotation **owl:incompatibleWith** with the value equal to either the ontology IRI or the version IRI of O2

Axioms Closure

The **axiom closure** of an ontology O is the **set of axioms that constitutes the ontology**

Since an ontology O can be spread over several imported ontologies, the **axiom closure of O** is defined to be the **smallest set that contains all the axioms from each ontology in the import closure of O**, with all anonymous individuals (i.e., blank nodes) standardized apart. That is, the anonymous individuals from different ontologies in the import closure of O are treated as being different

Ontology Document

According to the OWL DL abstract syntax, an **ontology document** consists of the following parts:

- **zero or more prefix declarations**, each given by the keyword Prefix (in Turtle syntax @prefix) followed by a prefix name and an IRI;
- **the keyword "Ontology",** optionally followed by an ontology IRI, optionally followed by a version IRI; this part of the ontology document is serialized in RDF Turtle as follows:
  - ontologyIRI rdf:type owl:Ontology .
  - ontologyIRI owl:versionIRI versionIRI .

If no version IRI is given, then, the second triple above is omitted.

If no ontology IRI is given, then the triples are the same as those above, except that a **blank node _:x is used in place of ontologyIRI**

Ontology Document

- **zero or more import statements**, each given by the keyword *Import* followed by an IRI; each import statement is serialized as follows:

  ontologyIRI owl:imports importedOntologyIRI .

- **zero or more ontology annotations**

- **zero or more axioms**

Annotations

OWL DL ontologies include two main kinds of annotations:

- **contextual annotations** (now), and
- **annotation axioms** (later)

Contextual annotations are embedded in ontologies, axioms or other annotation statements

The distinctive feature of a contextual annotation is that it is **about something that emerges from the context in which the annotation is embedded**

A contextual annotation has the abstract syntax:

Annotation(AP v)

where **AP** is an annotation property, and **v** is an annotation value, which can be a named or anonymous individual or a literal

Annotations

For instance, the annotation of an axiom:

Annotation(AP v)

SubClassOf(Annotation(rdfs:comment "Male people are people") :Man :Person)

Annotation Properties

OWL DL offers some **built-in annotation properties to ease and standardize the expression of contextual annotations**

- Built-in annotation properties are **rdfs:label**, **rdfs:comment**, **rdfs:seeAlso** and **rdfs:isDefinedBy**, all imported from the already examined RDF and RDF Schema vocabularies
- The **owl:versionInfo** annotation property associates any entity with a string literal that gives version information about that entity

Annotation Properties

An ontology can have a set of annotations, which are:

- **owl:priorVersion** annotation property specifies the IRI of a prior version of the containing ontology
- **owl:backwardCompatibleWith** annotation property specifies the IRI of a prior version of the containing ontology that is compatible with the current version of the containing ontology
- **owl:incompatibleWith** annotation property specifies the IRI of a prior version of the containing ontology that is incompatible with the current version of the containing ontology

Datatypes

Datatypes are provided in OWL **the same way they are provided in RDF**, to allow using standardized values such as numbers and strings

Each datatype is identified by an IRI and is defined by the following components:

- The **value space** is the set of values of the datatype
- The **lexical space** is a set of strings that can be used to refer to data values. Each member of the lexical space is called a lexical form, and it is mapped to a particular data value
- The **facet space** is a set of pairs of the form (F, v) where F is an IRI called a **constraining facet**, and v is an arbitrary data value called the **constraining value**. Each such pair is mapped to a subset of the value space of the datatype

Facet Space

- For the XML Schema datatypes xsd:double, xsd:float, and xsd:decimal, the allowed **constraining facets** are: xsd:minInclusive, xsd:maxInclusive, xsd:minExclusive and xsd:maxExclusive
- Example: The pair (xsd:minInclusive,v) of the facet space denotes the set of all numbers *x* from the value space of the datatype such that x=v or x>v
- Similarly for other datatypes

Datatypes

A set of datatypes supported by a **reasoner** is called a **datatype map**

The **OWL 2 datatype map** lists the datatypes that can be used in OWL 2 ontologies

Most datatypes are taken from the set of **XML Schema Datatypes**, the RDF specification, or the specification for plain literals

In addition, the OWL 2 datatype map adds:

- **owl:real**, whose value space is the set of real numbers and it does not directly provide any lexical forms
- **owl:rational**, whose value space is the set of rational numbers, and whose lexical space is given by

  numerator '/' denominator

where the numerator is an xsd:integer and the denominator is a positive, non-zero xsd:integer

Datatypes

The complete OWL 2 datatype map consists of the following datatypes:

owl:real
owl:rational
xsd:decimal
xsd:integer
xsd:nonNegativeInteger
xsd:nonPositiveInteger
xsd:positiveInteger
xsd:negativeInteger
xsd:long
xsd:int
xsd:short
xsd:byte
xsd:unsignedLong
xsd:unsignedInt
xsd:unsignedShort
xsd:unsignedByte

xsd:double
xsd:float

xsd:string
xsd:normalizedString
xsd:token
xsd:language
xsd:Name
xsd:NCName
xsd:NMTOKEN

xsd:boolean

xsd:hexBinary
xsd:base64Binary

xsd:anyURI

xsd:dateTime
xsd:dateTimeStamp

rdf:XMLLiteral