Learn OWL and RDFS

OWL References for Humans

This lesson provides an introduction to the most useful stuff in OWL. It's not intended to be comprehensive, but it does contain quite a few of the constructs that you're likely run into.

Note: many of the examples in this lesson came from the more technical OWL Primer (http://www.w3.org/TR/2009/REC-owl2-primer-20091027/). We have repackaged them in the hopes of making them more accessible to those newer to OWL.

Today's Lesson

- Classes and Resources
- Properties
- Restrictions

Classes and Resources

One area in which OWL goes significantly beyond RDFS (https://www.cambridgesemantics.com/semantic-university/rdfs-vs-owl) is that it allows you to construct some fairly complex, but useful, relationships among classes. Some of the most common building blocks for doing so are listed below.

Classes and Resources		
Property	**Used to say that…**	**Example**
intersectionOf	…any instance of the first class is also an instances of all classes in the specified list	:Mother owl:intersectionOf (:Woman :Parent)
unionOf	…any instance of the first class is an instance of at least one of the classes in the specified list	:Parent owl:unionOf (:Mother :Father)
complementOf	…the first class is equivalent to everything not in the second class	:Parent owl:complementOf :NonParent
disjointWith	…the first class and second class have no members in common	:Man owl:disjointWith :Woman
equivalentClass	…the first class and the second class contain all the same members	:AdultFemaleHuman owl:equivalentClass :Woman
sameAs	…the first resource refers to the exact same thing as the second resource	:JimFromWork owl:sameAs :MyNeighborJim
differentFrom	…the first resource refers to something different from the second resource	:BobFromWork owl:differentFrom :MyNeighborBob

Properties

As with RDFS, properties in OWL are used to link things together. OWL provides a rich and complex vocabulary for saying things about these links.

Basic Property Types			
Kind of Property	**Used to say…**	**Example**	**Explanation**
DatatypeProperty	…that this property links to simple data values	ex:hasBirthday	This property links to a date, which is a simple data value
ObjectProperty	…that this property links to another resource	ex:hasSpouse	This property links to a person, which is another resource

Logical Relationships			
Kind of Property	**Used to say…**	**Example**	**Explanation**
TransitiveProperty	…that if this property links A to B, and B to C, then it also links A to C.	ex:tallerThan	If Ann is taller than Bob, and Bob is taller than Chuck, then Ann is taller than Chuck
SymmetricProperty	…that if the property relates A to B, then it always relates B to A as well.	ex:hasSpouse	If Ann is Bob's spouse, then Bob is Ann's spouse too
AsymmetricProperty	…that if the property relates A to B, then it never relates B to A.	ex:tallerThan	If Ann is taller than Bob, then Bob can't be taller than Ann
ReflexiveProperty	…that this property always links something to itself.	ex:livesWith	Everybody lives with themselves
IrreflexiveProperty	…that this property never links something to itself.	ex:hasSpouse	Nobody is their own spouse
FunctionalProperty	…that this property only ever links to at most one thing.	ex:hasBirthday	You only h…
InverseFunctionalProperty	…that the subject of this property is uniquely identified by the value of this property.	ex:hasDLNumber	I am the o… drivers lic…
Properties Linking Properties			

Property	Used to say that...	Example
inverseOf	...the two properties are the inverse of each other. For example, if Ann's child is Bob, then Bob's parent is Ann.	:hasChild owl:inverseOf :hasParent
equivalentProperty	...two properties are exactly the same	:hasBirthPlace owl:equivalentProperty :hasBirthLocation

Restrictions

In RDFS, you could impose constraints on properties simply by specifying the domain and range. For example, if you asserted the range of `:hasBirthday` is `xsd:date`, then all statements using `:hasBirthday` should have an `xsd:date` as their object.

OWL lets you do this too, but it also introduces the concepts of restrictions, enumerations, and dataranges which are much more powerful.

(Note: In the Turtle RDF syntax, these constructs are usually specified using the bracketed blank node syntax, which we'll use below.)

Restrictions and Enumerations			
Parameter	**Used to say...**	**Example**	**Explanation**
cardinality min-cardinality max-cardinality	...that the property can have a certain number of values (objects).	:Automobile owl:equivalentclass [rdf:type owl:Restriction ; owl:cardinality "4"^^xsd:int ; owl:onProperty :hasWheel] .	All automobiles have 4 wheels (e.g., as opposed to a bicycle).
oneOf	...that all instances of a class come from the specified list	:BobsChildren owl:equivalentClass [rdf:type owl:Class ; owl:oneOf (:Bill :John :Mary)] .	The class 'BobsChildren' has the three items: Bill, John, and Mary
hasValue	...that all objects of that property have the specified value	:BobsChildren owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :hasParent ; owl:hasValue :Bob] .	Each instance of BobsChildren has 'Bob' as the object of its :hasParent property.
someValuesFrom	...that at least one object of that property is a member of the specified class.	:Parent owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :hasChild ; owl:someValuesFrom :Person] .	Any instance of the 'Parent' class has at least one child that is a Person
allValuesFrom	...that all objects of that property are members of the specified class	:Vegetarian owl:equivalentClass [rdf:type owl:Restriction ; owl:onProperty :eats ; owl:allValuesFrom :NonMeat] .	The class 'Vegetarian' is equivalent to the class of things that only eat non-meat.

Conclusion

That's the end of our general overview of some basic OWL constructs. We hope this tutorial has been helpful.

‹ PREVIOUS LESSON (HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/LEARN-OWL-RDFS/FLAVORS-OF-OWL/)

(https://www.cambridgesemantics.com)

f-service)
ement &
ML and
AI (/solutions/anzo/#data-management)
Industry Solutions (/solutions/#industry-use-cases)
- Financial Services (/solutions/financial/)
- Government (/solutions/government-solutions/)
- Healthcare (/solutions/healthcare/)
- Insurance (/solutions/insurance/)
- Life Sciences (/solutions/life-sciences/)
- Material Sciences (/solutions/material-sciences/)
- Retail (/solutions/retail/)

In The News (/category/news/)
Events (/category/events/)
Press Releases (/category/press-releases/)
Awards & Recognition (/category/awards-recognition/)
Culture & Careers (/about-us/careers/)
Management Team (/about-us/#team)

White Papers (/category/white-papers/)
Articles (/category/articles/)
Videos (/videos/)
Documentation (https://docs.cambridgesemantics.com/)
Semantic University (/blog/semantic-university/)
- Getting Started: Understanding Semantic Technologies (/blog/semantic-university/intro-semantic-web/)
- Semantic Technologies Compared (/blog/semantic-university/comparing-semantic-technologies/)
- Semantic Technologies Applied (/blog/semantic-university/semantic-technologies-applied/)

(https://www.linkedin.com/company/cambridge-semantics/)
(https://www.youtube.com/channel/UCu6WgFr6nmwtMsFpqDOY
(https://www.slideshare.net/camsemantics)
Request a Demo (/request-a-demo/)

1 Beacon St | 15th Floor | Boston, MA 02108