Quick Reference Guide

Authors

Jie Bao, Rensselaer Polytechnic Institute
Elisa F. Kendall, Sandpiper Software, Inc.
Deborah L. McGuinness, Rensselaer Polytechnic Institute

Contributors

Li Ding, Rensselaer Polytechnic Institute
Ankesh Khandelwal, Rensselaer Polytechnic Institute
Peter F. Patel-Schneider, Bell Labs Research, Alcatel-Lucent

1 Prefixes

The The standard prefixes in OWL 2 are:

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

2 OWL 2 constructs and axioms

Features in bold are only available in OWL 2. We use the following notation conventions: "C" is a class expression, "D" is a data range, "P" is an object property expression, "R" is a data property, "A" is an annotation property, "a" is an OWL individual, "v" is a literal, "n" is a non-negative integer, "_:x" is a blank node, "O" is an ontology, and U is an IRI. All of the previous can have subscripts. "(a_1 … a_n)" is an RDF list.

For an OWL 2 DL ontology, there are some global restrictions on axioms.

Classes

Predefined and Named Classes

universal class	owl:Thing	owl:Thing
empty class	owl:Nothing	owl:Nothing
named class	U	U

Boolean Connectives and Enumeration of Individuals

intersection	ObjectIntersectionOf(C_1 … C_n)	_:x rdf:type owl:Class. _:x owl:intersectionOf (C_1 … C_n).
union	ObjectUnionOf(C_1 … C_n)	_:x rdf:type owl:Class. _:x owl:unionOf (C_1 … C_n).
complement	ObjectComplementOf(C)	_:x rdf:type owl:Class. _:x owl:complementOf C.
enumeration	ObjectOneOf(a_1 … a_n)	_:x rdf:type owl:Class. _:x owl:oneOf (a_1 … a_n).

Object Property Restrictions

universal	ObjectAllValuesFrom(P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:allValuesFrom C
existential	ObjectSomeValuesFrom(P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:someValuesFrom C
individual value	ObjectHasValue(P a)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasValue a.
local reflexivity (N)	ObjectHasSelf(P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasSelf "true"^^xsd:boolean.
exact cardinality	ObjectExactCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:cardinality n.
qualified exact cardinality (N)	ObjectExactCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:qualifiedCardinality n. _:x owl:onClass C.
maximum cardinality	ObjectMaxCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minCardinality n.
qualified maximum cardinality (N)	ObjectMaxCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minQualifiedCardinality n. _:x owl:onClass C.
minimum cardinality	ObjectMinCardinality(n P)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxCardinality n.
qualified minimum cardinality (N)	ObjectMinCardinality(n P C)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxQualifiedCardinality n. _:x owl:onClass C.

Data Property Restrictions

universal	DataAllValuesFrom(R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:allValuesFrom D.
existential	DataSomeValuesFrom(R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:someValuesFrom D.
individual value	DataHasValue(R u)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:hasValue u.
exact cardinality	DataExactCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:cardinality n.
qualified exact cardinality (N)	DataExactCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:qualifiedCardinality n. _:x owl:onDataRange D.
maximum cardinality	DataMaxCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxCardinality n.
qualified maximum cardinality (N)	DataMaxCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxQualifiedCardinality n. _:x owl:onDataRange D.
minimum cardinality	DataMinCardinality(n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minCardinality n.
qualified minimum cardinality (N)	DataMinCardinality(n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minQualifiedCardinality n. _:x owl:onDataRange D.

Restrictions Using n-ary Data Range

"D^n" is an n-ary data range.

n-ary universal (N)	DataAllValuesFrom(R_1 … R_n D^n)	_:x rdf:type owl:Restriction. _:x owl:onProperties (R_1 … R_n). _:x owl:allValuesFrom D^n.
n-ary existential (N)	DataSomeValuesFrom(R_1 … R_n D^n)	_:x rdf:type owl:Restriction. _:x owl:onProperties (R_1 … R_n). _:x owl:someValuesFrom D^n.

Class Axioms

subclasses	SubClassOf(C_1 C_2)	C_1 rdfs:subClassOf C_2.
equivalent classes	EquivalentClasses(C_1 … C_n)	C_i owl:equivalentClass C_{i+1}. i=1…n-1
disjoint classes	DisjointClasses(C_1 C_2)	C_1 owl:disjointWith C_2.
pairwise disjoint classes (N)	DisjointClasses(C_1 … C_n)	_:x rdf:type owl:AllDisjointClasses. _:x owl:members (C_1 … C_n).
disjoint union (N)	DisjointUnionOf(C C_1 … C_n)	C owl:disjointUnionOf (C_1 … C_n).

Properties

Object Property Expressions

named object property	U	U
universal object property (N)	owl:topObjectProperty	owl:topObjectProperty
empty object property (N)	owl:bottomObjectProperty	owl:bottomObjectProperty
inverse property	ObjectInverseOf(P)	_:x owl:inverseOf P

Datatype Property Expressions

named datatype property	U	U
universal datatype property (N)	owl:topDataProperty	owl:topDataProperty
empty datatype property (N)	owl:bottomDataProperty	owl:bottomDataProperty

Object Property Axioms

subproperty	SubObjectPropertyOf(P_1 P_2)	P_1 rdfs:subPropertyOf P_2.
property chain inclusion (N)	SubObjectPropertyOf(ObjectProperty Chain(P_1 … P_n) P)	P owl:propertyChainAxiom (P_1 … P_n).
property domain	ObjectPropertyDomain(P C)	P rdfs:domain C.
property range	ObjectPropertyRange(P C)	P rdfs:range C.
equivalent properties	EquivalentObjectProperties(P_1 … P_n)	P_i owl:equivalentProperty P_{i+1}. j=1…n-1
disjoint properties (N)	DisjointObjectProperties(P_1 P_2)	P_1 owl:propertyDisjointWith P_2.
pairwise disjoint properties (N)	DisjointObjectProperties(P_1 … P_n)	_:x rdf:type owl:AllDisjointProperties. _:x owl:members (P_1 … P_n).
inverse properties	InverseObjectProperties(P_1 P_2)	P_1 owl:inverseOf P_2.
functional property	FunctionalObjectProperty(P)	P rdf:type owl:FunctionalProperty.
inverse functional property	InverseFunctionalObjectProperty(P)	P rdf:type owl:InverseFunctionalProperty.
reflexive property (N)	ReflexiveObjectProperty(P)	P rdf:type owl:ReflexiveProperty.
irreflexive property (N)	IrreflexiveObjectProperty(P)	P rdf:type owl:IrreflexiveProperty.
symmetric property	SymmetricObjectProperty(P)	P rdf:type owl:SymmetricProperty.
asymmetric property (N)	AsymmetricObjectProperty(P)	P rdf:type owl:AsymmetricProperty.
transitive property	TransitiveObjectProperty(P)	P rdf:type owl:TransitiveProperty.

Datatype Property Axioms

subproperty	SubDataPropertyOf(R_1 R_2)	R_1 rdfs:subPropertyOf R_2.
property domain	DataPropertyDomain(R C)	R rdfs:domain C.
property range	DataPropertyRange(R D)	R rdfs:range D.
equivalent properties	EquivalentDataProperties(R_1 … R_n)	R_i owl:equivalentProperty R_{i+1}. j=1…n-1
disjoint properties (N)	DisjointDataProperties(R_1 R_2)	R_1 owl:propertyDisjointWith R_2.
pairwise disjoint properties (N)	DisjointDataProperties(R_1 … R_n)	_:x rdf:type owl:AllDisjointProperties. _:x owl:members (R_1 … R_n).
functional property	FunctionalDataProperty(R)	R rdf:type owl:FunctionalProperty.

Individuals & Literals

named individual	U	U
anonymous individual	_:a	_:a
literal	"abc"^^datatypeIRI	"abc"^^datatypeIRI

Data Ranges

named data type	U	U
data range complement (N)	DataComplementOf(D)	_:x rdf:type rdfs:Datatype. _:x owl:datatypeComplementOf D.
data range intersection (N)	DataIntersectionOf($D_1 \ldots D_n$)	_:x rdf:type rdfs:Datatype. _:x owl:intersectionOf ($D_1 \ldots D_n$).
data range union (N)	DataUnionOf($D_1 \ldots D_n$)	_:x rdf:type rdfs:Datatype. _:x owl:unionOf ($D_1 \ldots D_n$).
literal enumeration	DataOneOf($v_1 \ldots v_n$)	_:x rdf:type rdfs:Datatype. _:x owl:oneOf ($v_1 \ldots v_n$).
datatype restriction (N)	DatatypeRestriction(D f_1 v_1 … f_n v_n) *D a built-in datatype* *f_i a constraining facet,* *v_j a restriction value*	_:x rdf:type rdfs:Datatype. _:x owl:onDatatype D. _:x owl:withRestrictions (y1 ... yn). y_i f_j v_j. j=1…n

Keys

Keys (N)	HasKey(C ($P_1 \ldots P_m$) ($R_1 \ldots R_n$))	C owl:hasKey ($P_1 \ldots P_m$ $R_1 \ldots R_n$).

Assertions

individual equality	SameIndividual(a_1 a_2)	a_1 owl:sameAs a_2.
n-ary individual equality	SameIndividual($a_1 \ldots a_n$)	a_i owl:sameAs a_{i+1}. j=1…n-1
individual inequality	DifferentIndividuals(a_1 a_2)	a_1 owl:differentFrom a_2.
pairwise individual inequality	DifferentIndividuals($a_1 \ldots a_n$)	_:x rdf:type owl:AllDifferent. _:x **owl:members** ($a_1 \ldots a_n$).
class assertion	ClassAssertion(C a)	a rdf:type C.
positive object property assertion	ObjectPropertyAssertion(P a_1 a_2) P is a named object property	a_1 P a_2.
positive inverse object property assertion	ObjectPropertyAssertion(ObjectInverseOf(P) a_1 a_2)	a_2 P a_1.
positive data property assertion	DataPropertyAssertion(P a v)	a P v.
negative object property assertion (N)	NegativeObjectPropertyAssertion(P a_1 a_2)	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a_1. _:x owl:assertionProperty P. _:x owl:targetIndividual a_2
negative datatype property assertion (N)	NegativeDataPropertyAssertion(R a u)	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a. _:x owl:assertionProperty R. _:x owl:targetValue u

Declarations

C, D, P, R, R, A, a are all named.

class (N)	Declaration(Class(C))	C rdf:type owl:Class.
datatype (N)	Declaration(Datatype(D))	D rdf:type rdfs:Datatype.
object property (N)	Declaration(ObjectProperty(P))	P rdf:type owl:ObjectProperty.
datatype property (N)	Declaration(DataProperty(R))	R rdf:type owl:DatatypeProperty.
annotation property (N)	Declaration(AnnotationProperty(A))	A rdf:type owl:AnnotationProperty.
named individual (N)	Declaration(NamedIndividual(a))	a rdf:type owl:NamedIndividual.

Annotations

Annotation of an object s the annotated object, v a resource.

annotation assertions	AnnotationAssertion(A s v)	s A v. or (if the assertion itself has annotation) **s A v.** **_:x rdf:type owl:Annotation.** **_:x owl:subject s.** **_:x owl:predicate A.** **_:x owl:object v.**

Note: an annotated object can be a named ontology, a named ontology entity (class, datatype, property or individual), an anonymous individual, or another annotation.

Annotation of an axiom y the annotated object, v a resource

axiom annotations (N)	AXIOM(Annotation(A v))	s p o. _:x rdf:type owl:Axiom. _:x owl:subject s. _:x owl:predicate p. _:x owl:object o. _:x A v. *If AXIOM(...) becomes s p o.* _:x p o _:x A v. *If AXIOM(...) becomes _:x p o.*

Note: for n-ary axioms of type EquivalentClasses, EquivalentProperties or SameIndividual, they will first be broken up into several RDF triples of their binary forms, then each triple is reified using the above rule.

Annotation Properties

human-readable name	Label	rdfs:label
human-readable description	Comment	rdfs:comment
additional information		rdfs:seeAlso
defining agent		rdfs:isDefinedBy
version information		owl:versionInfo
deprecation	Deprecated	owl:deprecated

Annotation Axioms

annotation subproperties (N)	SubAnnotationPropertyOf(A_1 A_2)	A_1 rdfs:subPropertyOf A_2.
annotation property domain (N)	AnnotationPropertyDomain(A U)	A rdfs:domain U.
annotation property range (N)	AnnotationPropertyRange(A U)	A rdfs:range U.

Deprecation

C a named class or a named datatype, P an named object property, named datatype property or annotation property;

deprecated class	C owl:deprecated "true"^^xsd:boolean	C rdf:type owl:DeprecatedClass.
deprecated property	P owl:deprecated "true"^^xsd:boolean	P rdf:type owl:DeprecatedProperty.
deprecated IRI	U owl:deprecated "true"^^xsd:boolean	U owl:deprecated "true"^^xsd:boolean.

Ontologies

Ontologies U an ontology IRI, V an IRI.

OWL ontology (importing)	Ontology(O [V] Import(U)... Annotation(A, v)...)	O rdf:type owl:Ontology. [O owl:versionIRI V.] O owl:imports U. ... O A v. ...

Note: if O is unnamed, then O is mapped to _:x (an anonymous individual).

Ontology Properties

backwards	O owl:backwardCompatibleWith U.
compatibility	
incompatibility	O owl:incompatibleWith U.
prior version	O owl:priorVersion U.

Deprecated Vocabulary in OWL 2

owl:DataRange	replaced by rdfs:Datatype
owl:distinctMembers	replaced by owl:members

Built-in Datatypes and Facets

Built-in Datatypes

The *value space* is a set determining the set of values of the datatype. A literal value "abc" of the datatype DT can be given in the form "abc"^^DT.

Numbers	**owl:rational**		**owl:real** (N)	
	xsd:double	xsd:float	xsd:decimal	xsd:integer
	xsd:long	xsd:int	xsd:short	xsd:byte
	xsd:nonNegativeInteger		xsd:nonPositiveInteger	
	xsd:positiveInteger		xsd:negativeInteger	
	xsd:unsignedLong		xsd:unsignedInt	
	xsd:unsignedShort		xsd:unsignedByte	
Strings	**rdf:text** (internationalized strings)			
	xsd:string	xsd:NCName	xsd:Name	xsd:NMTOKEN
	xsd:token	xsd:language	xsd:normalizedString	
Boolean Values	xsd:Boolean (value space: *true* and *false*)			
Binary Data	xsd:base64Binary		xsd:hexBinary	
IRIs	xsd:anyURI			
Time Instants	xsd:dateTime (with time zone offset) **xsd:dateTimeStamp** (without time zone offset)			
XML Literals	rdf:XMLLiteral (Note: at Risk in OWL 2)			

Facets

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 a value. Each such pair is mapped to a subset of the value space of the datatype.

Notations: Numeric Datatype, String Datatype and Binary datatype refer to a set of datatypes based on the classification done in the prior section.

Facet (N) f	Datatype	Value v	Explanation
xsd:minInclusive, xsd:maxInclusive, xsd:minExclusive, xsd:maxExclusive	Numeric Datatype DT, Time instant DT	Literal in DT	Restricts the value-space to greater than (equal to) or lesser than (equal to) a value
xsd:minLength, xsd:maxLength, xsd:length	String Datatype, Binary Datatype, xsd:anyURI	Non-negative integer	Restricts the value-space based on the lengths of the literals
xsd:pattern	String Datatype, xsd:anyURI	xsd:string literal whose value is a regular expression	Restricts the value space to literals that match the regular expression
rdf:langRange	rdf:text	xsd:string literal whose value is a regular expression	Restricts the value space to literals with language tags that match the regular expression