Abstract

OWL 2 extends the W3C OWL Web Ontology Language with a small but useful set of features that have been requested by users, for which effective reasoning algorithms are now available, and that OWL tool developers are willing to support. The new features include extra syntactic sugar, additional property and qualified cardinality constructors, extended datatype support, simple metamodeling, and extended annotations.

This document provides the direct model-theoretic semantics for OWL 2, which is compatible with the description logic SROIQ. Furthermore, this document defines the most common inference problems for OWL 2.
Status of this Document

May Be Superseded

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Set of Documents

This document is being published as one of a set of 11 documents:

1. Structural Specification and Functional-Style Syntax
2. Direct Semantics (this document)
3. RDF-Based Semantics
4. Conformance and Test Cases
5. Mapping to RDF Graphs
6. XML Serialization
7. Profiles
8. Quick Reference Guide
9. New Features and Rationale
10. Manchester Syntax
11. rdf:text: A Datatype for Internationalized Text

Last Call

The Working Group believes it has completed its design work for the technologies specified this document, so this is a "Last Call" draft. The design is not expected to change significantly, going forward, and now is the key time for external review, before the implementation phase.

Summary of Changes

This document has been updated to keep in sync with the Syntax document. The most significant update is in the formal definition of the datatype map.

Please Comment By 23 January 2009

The OWL Working Group seeks public feedback on these Working Drafts. Please send your comments to public-owl-comments@w3.org (public archive). If possible, please offer specific changes to the text that would address your concern. You may also wish to check the Wiki Version of this document for internal-review comments and changes being drafted which may address your concerns.
No Endorsement

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Patents

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Contents

• 1 Introduction
  • 2 Direct Model-Theoretic Semantics for OWL 2
    ◦ 2.1 Vocabulary
    ◦ 2.2 Interpretations
      ▪ 2.2.1 Object Property Expressions
      ▪ 2.2.2 Data Ranges
      ▪ 2.2.3 Class Expressions
    ◦ 2.3 Satisfaction in an Interpretation
      ▪ 2.3.1 Class Expression Axioms
      ▪ 2.3.2 Object Property Expression Axioms
      ▪ 2.3.3 Data Property Expression Axioms
      ▪ 2.3.4 Keys
      ▪ 2.3.5 Assertions
      ▪ 2.3.6 Ontologies
    ◦ 2.4 Models
    ◦ 2.5 Inference Problems
      ▪ 3 Independence of the Semantics from the Datatype Map
      ▪ 4 Acknowledgments
      ▪ 5 References

1 Introduction

This document defines the direct model-theoretic semantics of OWL 2. The semantics given here is strongly related to the semantics of description logics
Description Logics and is compatible with the semantics of the description logic SROIQ [SROIQ]. As the definition of SROIQ does not provide for datatypes and punning, the semantics of OWL 2 is defined directly on the constructs of the structural specification of OWL 2 [OWL 2 Specification] instead of by reference to SROIQ. For the constructs available in SROIQ, the semantics of SROIQ trivially corresponds to the one defined in this document.

Since OWL 2 is an extension of OWL DL, this document also provides a direct semantics for OWL Lite and OWL DL; this semantics is equivalent to the official semantics of OWL Lite and OWL DL [OWL Abstract Syntax and Semantics]. Furthermore, this document also provides the direct model-theoretic semantics for the OWL 2 profiles [OWL 2 Profiles].

The semantics is defined for an OWL 2 axioms and ontologies, which should be understood as instances of the structural specification [OWL 2 Specification]. Parts of the structural specification are written in this document using the functional-style syntax.

OWL 2 allows for annotations of ontologies, anonymous individuals, axioms, and other annotations. Annotations of all these types, however, have no semantic meaning in OWL 2 and are ignored in this document. OWL 2 declarations are used only to disambiguate class expressions from data ranges and object property from data property expressions in the functional-style syntax; therefore, they are not mentioned explicitly in this document.

2 Direct Model-Theoretic Semantics for OWL 2

This section specifies the direct model-theoretic semantics of OWL 2 ontologies.

2.1 Vocabulary

A datatype map is a 6-tuple \( D = ( N_{DT}, N_{LS}, N_{FS}, \cdot^{DT}, \cdot^{LS}, \cdot^{FS} ) \) with the following components.

- \( N_{DT} \) is a set of datatypes that does not contain the datatype rdfs:Literal.
- \( N_{LS} \) is a function that assigns to each datatype \( DT \in N_{DT} \) a set \( N_{LS}(DT) \) of strings called lexical values. The set \( N_{LS}(DT) \) is called the lexical space of \( DT \).
- \( N_{FS} \) is a function that assigns to each datatype \( DT \in N_{DT} \) a set \( N_{FS}(DT) \) of pairs \( \langle F, v \rangle \), where \( F \) is a constraining facet and \( v \) is an arbitrary object called a value. The set \( N_{FS}(DT) \) is called the facet space of \( DT \).
- For each datatype \( DT \in N_{DT} \), the interpretation function \( \cdot^{DT} \) assigns to \( DT \) a set \( (DT)^{DT} \) called the value space of \( DT \).
- For each datatype \( DT \in N_{DT} \) and each lexical value \( LV \in N_{LS}(DT) \), the interpretation function \( \cdot^{LS} \) assigns to the pair \( \langle LV, DT \rangle \) a data value \( (\langle LV, DT \rangle)^{LS} \in (DT)^{DT} \).
• For each datatype $DT \in N_{DT}$ and each pair $\langle F \, v \rangle \in N_{FS}(DT)$, the interpretation function $\cdot_{FS}$ assigns to $\langle F \, v \rangle$ a facet value $(\langle F \, v \rangle)^{FS} \subseteq (DT)^{DT}$.

A vocabulary $V = (V_C, V_{OP}, V_{DP}, V_I, V_{DT}, V_{LT}, V_{FA})$ over a datatype map $D$ is a 7-tuple consisting of the following elements:

• $V_C$ is a set of classes as defined in the OWL 2 Specification [OWL 2 Specification], containing at least the classes owl:Thing and owl:Nothing.
• $V_{OP}$ is a set of object properties as defined in the OWL 2 Specification [OWL 2 Specification], containing at least the object properties owl:topObjectProperty and owl:bottomObjectProperty.
• $V_{DP}$ is a set of data properties as defined in the OWL 2 Specification [OWL 2 Specification], containing at least the data properties owl:topDataProperty and owl:bottomDataProperty.
• $V_I$ is a set of individuals (named and anonymous) as defined in the OWL 2 Specification [OWL 2 Specification].
• $V_{DT}$ is the set of all datatypes of $D$ extended with the datatype rdfs:Literal; that is, $V_{DT} = N_{DT} \cup \{ rdfs:Literal \}$.
• $V_{LT}$ is a set of literals $LV \wedge DT$ for each datatype $DT \in N_{DT}$ and each lexical value $LV \in N_{LS}(DT)$.
• $V_{FA}$ is the set of pairs $\langle F \, lt \rangle$ for each constraining facet $F$, datatype $DT \in N_{DT}$, and literal $lt \in V_{LT}$ such that $\langle F \, (\langle LV \, DT \rangle)^{LS} \rangle \in N_{FS}(DT)$, where $LV$ is the lexical value of $lt$ and $DT$ is the datatype of $lt$.

Given a vocabulary $V$, the following conventions are used in this document to denote different syntactic parts of OWL 2 ontologies:

• $OP$ denotes an object property;
• $OPE$ denotes an object property expression;
• $DP$ denotes a data property;
• $DPE$ denotes a data property expression;
• $PE$ denotes an object property or a data property expression;
• $C$ denotes a class;
• $CE$ denotes a class expression;
• $DT$ denotes a datatype;
• $DR$ denotes a data range;
• $a$ denotes an individual (named or anonymous);
• $lt$ denotes a literal; and
• $F$ denotes a constraining facet.

2.2 Interpretations

Given a datatype map $D$ and a vocabulary $V$ over $D$, an interpretation $Int = (\Delta_{Int}, \Delta_D, C, OP, DP, I, DT, LT, FA)$ for $D$ and $V$ is a 9-tuple with the following structure.

• $\Delta_{Int}$ is a nonempty set called the object domain.
• $\Delta_D$ is a nonempty set disjoint with $\Delta_{\text{int}}$ called the data domain such that $(DT)^{DT}_{DT} \subseteq \Delta_D$ for each datatype $DT \in V_D$.

• $\cdot^C$ is the class interpretation function that assigns to each class $C \in V_C$ a subset $(C)^C \subseteq \Delta_{\text{int}}$ such that
  - $(\text{owl:Thing})^C = \Delta_{\text{int}}$ and
  - $(\text{owl:Nothing})^C = \emptyset$.

• $\cdot^{OP}$ is the object property interpretation function that assigns to each object property $OP \in V_{OP}$ a subset $(OP)^{OP} \subseteq \Delta_{\text{int}} \times \Delta_{\text{int}}$ such that
  - $(\text{owl:topObjectProperty})^{OP} = \Delta_{\text{int}} \times \Delta_D$ and
  - $(\text{owl:bottomObjectProperty})^{OP} = \emptyset$.

• $\cdot^{DP}$ is the data property interpretation function that assigns to each data property $DP \in V_{DP}$ a subset $(DP)^{DP} \subseteq \Delta_{\text{int}} \times \Delta_D$ such that
  - $(\text{owl:topDataProperty})^{DP} = \Delta_{\text{int}} \times \Delta_{\text{int}}$ and
  - $(\text{owl:bottomDataProperty})^{DP} = \emptyset$.

• $\cdot^I$ is the individual interpretation function that assigns to each individual $a \in V_I$ an element $(a)^I \in \Delta_{\text{int}}$.

• $\cdot^{DT}$ is the datatype interpretation function that is the same as in $D$ for all datatypes $DT \in N_{DT}$ and is extended to rdfsLiteral by setting
  - $(\text{rdfs:LITERAL})^{DT} = \Delta_D$.

• $\cdot^{LT}$ is the literal interpretation function that is defined as $(lt)^{LT} = \langle (LV \ DT) \rangle^{LS}$ for each $lt \in V_{LT}$, where $LV$ is the lexical value of $lt$ and $DT$ is the datatype of $lt$.

• $\cdot^{FA}$ is the facet interpretation function that is defined as $(\langle F \ lt \rangle)^{FA} = \langle F \ (lt)^{LT} \rangle^{FS}$ for each $\langle F \ lt \rangle \in V_{FA}$.

The following sections define the extensions of $\cdot^{OP}$, $\cdot^{DT}$, and $\cdot^C$ to object property expressions, data ranges, and class expressions.

### 2.2.1 Object Property Expressions

The object property interpretation function $\cdot^{OP}$ is extended to object property expressions as shown in Table 1.

<table>
<thead>
<tr>
<th>Object Property Expression</th>
<th>Interpretation $\cdot^{OP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>InverseOf( OP )</td>
<td>${ (x, y)</td>
</tr>
</tbody>
</table>

### 2.2.2 Data Ranges

The datatype interpretation function $\cdot^{DT}$ is extended to data ranges as shown in Table 3. All datatypes in OWL 2 are unary, so each datatype $DT$ is interpreted as a unary relation over $\Delta_D$ — that is, a set $(DT)^{DT}_{DT} \subseteq \Delta_D$. Data ranges, however, can be $n$-ary, as this allows implementations to extend OWL 2 with built-in operations such as comparisons or arithmetic. An $n$-ary data range $DR$ is interpreted as an $n$-ary relation $(DR)^{DT}_{DT}$ over $\Delta_D$. 
### Table 3. Interpreting Data Ranges

<table>
<thead>
<tr>
<th>Data Range</th>
<th>Interpretation ( \cdot DT )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{IntersectionOf}( \text{DR}_1 \ldots \text{DR}_n ) )</td>
<td>((\text{DR}_1)^DT \cap \ldots \cap (\text{DR}_n)^DT)</td>
</tr>
<tr>
<td>( \text{UnionOf}( \text{DR}_1 \ldots \text{DR}_n ) )</td>
<td>((\text{DR}_1)^DT \cup \ldots \cup (\text{DR}_n)^DT)</td>
</tr>
<tr>
<td>( \text{ComplementOf}( \text{DR} ) )</td>
<td>((\Delta\text{D})^D \setminus (\text{DR})^DT) where (n) is the arity of (\text{DR})</td>
</tr>
<tr>
<td>( \text{OneOf}( \text{lt}_1 \ldots \text{lt}_n ) )</td>
<td>({ (\text{lt}_1)^{LT}, \ldots, (\text{lt}_n)^{LT} })</td>
</tr>
<tr>
<td>( \text{DatatypeRestriction}( \text{DT} \text{ F}_1 \text{ lt}_1 \ldots \text{ F}_n \text{ lt}_n ) )</td>
<td>((\text{DT})^DT \cap (\langle \text{ F}_1 \text{ lt}_1 \rangle)^{FA} \cap \ldots \cap (\langle \text{ F}_n \text{ lt}_n \rangle)^{FA})</td>
</tr>
</tbody>
</table>

#### 2.2.3 Class Expressions

The class interpretation function \( \cdot C \) is extended to class expressions as shown in Table 4. For \(S\) a set, \(\#S\) denotes the number of elements in \(S\).

### Table 4. Interpreting Class Expressions

<table>
<thead>
<tr>
<th>Class Expression</th>
<th>Interpretation ( \cdot C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{IntersectionOf}( \text{CE}_1 \ldots \text{CE}_n ) )</td>
<td>((\text{CE}_1)^C \cap \ldots \cap (\text{CE}_n)^C)</td>
</tr>
<tr>
<td>( \text{UnionOf}( \text{CE}_1 \ldots \text{CE}_n ) )</td>
<td>((\text{CE}_1)^C \cup \ldots \cup (\text{CE}_n)^C)</td>
</tr>
<tr>
<td>( \text{ComplementOf}( \text{CE} ) )</td>
<td>(\Delta\text{Int} \setminus (\text{CE})^C)</td>
</tr>
<tr>
<td>( \text{OneOf}( \text{a}_1 \ldots \text{a}_n ) )</td>
<td>({ (\text{a}_1)^I, \ldots, (\text{a}_n)^I })</td>
</tr>
<tr>
<td>( \text{SomeValuesFrom}( \text{OPE} \text{ CE} ) )</td>
<td>({ x \mid \exists y : \langle x, y \rangle \in (\text{OPE})^{OP} \text{ and } y \in (\text{CE})^C })</td>
</tr>
<tr>
<td>( \text{AllValuesFrom}( \text{OPE} \text{ CE} ) )</td>
<td>({ x \mid \forall y : \langle x, y \rangle \in (\text{OPE})^{OP} \text{ implies } y \in (\text{CE})^C })</td>
</tr>
<tr>
<td>( \text{HasValue}( \text{OPE} \text{ a} ) )</td>
<td>({ x \mid \langle x, (\text{a})^I \rangle \in (\text{OPE})^{OP} })</td>
</tr>
<tr>
<td>( \text{HasSelf}( \text{OPE} ) )</td>
<td>({ x \mid \langle x, x \rangle \in (\text{OPE})^{OP} })</td>
</tr>
<tr>
<td>( \text{MinCardinality}( n \text{ OPE} ) )</td>
<td>({ x \mid #{ y \mid \langle x, y \rangle \in (\text{OPE})^{OP} } \geq n })</td>
</tr>
<tr>
<td>( \text{MaxCardinality}( n \text{ OPE} ) )</td>
<td>({ x \mid #{ y \mid \langle x, y \rangle \in (\text{OPE})^{OP} } \leq n })</td>
</tr>
</tbody>
</table>
2.3 Satisfaction in an Interpretation

An interpretation $\text{Int} = (\Delta_{\text{Int}}, \Delta_{O}, C, O^\text{OP}, D \text{P}, I^\text{OP}, I^\text{DP}, I^\text{DT}, I^\text{LT}, I^\text{FA})$ satisfies an axiom w.r.t. an ontology $O$ if the axiom satisfies appropriate conditions listed in the following sections. Satisfaction of axioms in $\text{Int}$ is defined w.r.t. $O$ because satisfaction of key axioms uses the following function:

$$\text{ISNAMED}_O(x) = \text{true} \text{ for } x \in \Delta_{\text{Int}} \text{ if and only if } (a)^I = x \text{ for some named individual } a \text{ occurring in the axiom closure of } O$$
2.3.1 Class Expression Axioms

Satisfaction of OWL 2 class expression axioms in \( \text{Int} \) w.r.t. \( O \) is defined as shown in Table 5.

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubClassOf( ( CE_1 \ CE_2 ) )</td>
<td>( (CE_1)^C \subseteq (CE_2)^C )</td>
</tr>
<tr>
<td>EquivalentClasses( ( CE_1 \ldots CE_n ) )</td>
<td>( (CE_j)^C = (CE_k)^C ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n )</td>
</tr>
<tr>
<td>DisjointClasses( ( CE_1 \ldots CE_n ) )</td>
<td>( (CE_j)^C \cap (CE_k)^C = \emptyset ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n ) such that ( j \neq k )</td>
</tr>
<tr>
<td>DisjointUnion( ( C CE_1 \ldots CE_n ) )</td>
<td>( (C)^C = (CE_1)^C \cup \ldots \cup (CE_n)^C ) and ( (CE_j)^C \cap (CE_k)^C = \emptyset ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n ) such that ( j \neq k )</td>
</tr>
</tbody>
</table>

2.3.2 Object Property Expression Axioms

Satisfaction of OWL 2 object property expression axioms in \( \text{Int} \) w.r.t. \( O \) is defined as shown in Table 6.

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubPropertyOf( ( OPE_1 \ OPE_2 ) )</td>
<td>( (OPE_1)^{OP} \subseteq (OPE_2)^{OP} )</td>
</tr>
<tr>
<td>SubPropertyOf( ( \text{PropertyChain}( \ OPE_1 \ldots OPE_n \ ) \ OPE ) )</td>
<td>( \forall \ y_0, \ldots, y_n : \langle y_0, y_1 \rangle \in (OPE_1)^{OP} ) and ( \ldots ) and ( \langle y_{n-1}, y_n \rangle \in (OPE_n)^{OP} ) imply ( \langle y_0, y_n \rangle \in (OPE)^{OP} )</td>
</tr>
<tr>
<td>EquivalentProperties( ( OPE_1 \ldots OPE_n ) )</td>
<td>( (OPE_j)^{OP} = (OPE_k)^{OP} ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n )</td>
</tr>
<tr>
<td>DisjointProperties( ( OPE_1 \ldots OPE_n ) )</td>
<td>( (OPE_j)^{OP} \cap (OPE_k)^{OP} = \emptyset ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n ) such that ( j \neq k )</td>
</tr>
<tr>
<td>PropertyDomain( ( OPE \ CE ) )</td>
<td>( \forall \ x, y : \langle x, y \rangle \in (OPE)^{OP} ) implies ( x \in (CE)^C )</td>
</tr>
<tr>
<td>PropertyRange( ( OPE \ CE ) )</td>
<td>( \forall \ x, y : \langle x, y \rangle \in (OPE)^{OP} ) implies ( y \in (CE)^C )</td>
</tr>
</tbody>
</table>
2.3.3 Data Property Expression Axioms

Satisfaction of OWL 2 data property expression axioms in $\text{Int}$ w.r.t. $O$ is defined as shown in Table 7.

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubPropertyOf( DPE$<em>{1}$ DPE$</em>{2}$ )</td>
<td>$(\text{DPE}_1)^{\text{DP}} \subseteq (\text{DPE}_2)^{\text{DP}}$</td>
</tr>
<tr>
<td>EquivalentProperties( DPE$<em>{1}$ ... DPE$</em>{n}$ )</td>
<td>$\forall 1 \leq j \leq n \exists 1 \leq k \leq n$ $(\text{DPE}_j)^{\text{DP}} = (\text{DPE}_k)^{\text{DP}}$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$</td>
</tr>
<tr>
<td>DisjointProperties( DPE$<em>{1}$ ... DPE$</em>{n}$ )</td>
<td>$(\text{DPE}_1)^{\text{DP}} \cap (\text{DPE}_2)^{\text{DP}} = \emptyset$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$</td>
</tr>
<tr>
<td>PropertyDomain( DPE CE )</td>
<td>$\forall x, y : \langle x, y \rangle \in (\text{DPE})^{\text{DP}}$ implies $x \in (\text{CE})^{\text{C}}$</td>
</tr>
<tr>
<td>PropertyRange( DPE DR )</td>
<td>$\forall x, y : \langle x, y \rangle \in (\text{DPE})^{\text{DP}}$ implies $y \in (\text{DR})^{\text{DT}}$</td>
</tr>
<tr>
<td>FunctionalProperty( DPE )</td>
<td>$\forall x, y_1, y_2 : \langle x, y_1 \rangle \in (\text{DPE})^{\text{DP}}$ and $\langle x, y_2 \rangle \in (\text{DPE})^{\text{DP}}$ imply $y_1 = y_2$</td>
</tr>
</tbody>
</table>
2.3.4 Keys

Satisfaction of keys in \( \text{Int} \) w.r.t. \( O \) is defined as shown in Table 8.

**Table 8. Satisfaction of Keys in an Interpretation**

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
</table>
| HasKey( CE PE\(_1\) ... PE\(_n\) ) | \( \forall \, x, y, z_1, \ldots, z_n : \)  
  if \( \text{ISNAMED}(x) \) and \( \text{ISNAMED}(y) \) and \( \text{ISNAMED}(z_1) \) and ... and \( \text{ISNAMED}(z_n) \) and \( x \in (CE)^C \) and \( y \in (CE)^C \) and for each \( 1 \leq i \leq n \),  
  if \( PE_i \) is an object property, then \( \langle x, z_i \rangle \in (PE_i)^{OP} \) and \( \langle y, z_i \rangle \in (PE_i)^{OP} \), and  
  if \( PE_i \) is a data property, then \( \langle x, z_i \rangle \in (PE_i)^{DP} \) and \( \langle y, z_i \rangle \in (PE_i)^{DP} \),  
  then \( x = y \) |

2.3.5 Assertions

Satisfaction of OWL 2 assertions in \( \text{Int} \) w.r.t. \( O \) is defined as shown in Table 9.

**Table 9. Satisfaction of Assertions in an Interpretation**

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SameIndividual( a(_1) ... a(_n) )</td>
<td>( (a_j)^I = (a_k)^I ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n )</td>
</tr>
<tr>
<td>DifferentIndividuals( a(_1) ... a(_n) )</td>
<td>( (a_j)^I \neq (a_k)^I ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n ) such that ( j \neq k )</td>
</tr>
<tr>
<td>ClassAssertion( CE a )</td>
<td>( (a)^I \in (CE)^C )</td>
</tr>
<tr>
<td>PropertyAssertion( OPE a(_1) a(_2) )</td>
<td>( \langle (a_1)^I, (a_2)^I \rangle \in (OPE)^{OP} )</td>
</tr>
<tr>
<td>NegativePropertyAssertion( OPE a(_1) a(_2) )</td>
<td>( \langle (a_1)^I, (a_2)^I \rangle \notin (OPE)^{OP} )</td>
</tr>
<tr>
<td>PropertyAssertion( DPE a lt )</td>
<td>( \langle (a)^I, (lt)^{LT} \rangle \in (DPE)^{DP} )</td>
</tr>
<tr>
<td>NegativePropertyAssertion( DPE a lt )</td>
<td>( \langle (a)^I, (lt)^{LT} \rangle \notin (DPE)^{DP} )</td>
</tr>
</tbody>
</table>
2.3.6 Ontologies

Int satisfies an OWL 2 ontology O if all axioms in the axiom closure of O (with anonymous individuals renamed apart as described in Section 5.6.2 of the OWL 2 Specification [OWL 2 Specification]) are satisfied in Int w.r.t. O.

2.4 Models

An interpretation Int = (ΔInt, ΔD, ⋯ C, ⋯ OP, ⋯ DP, ⋯ I, ⋯ DT, ⋯ LT, ⋯ FA) is a model of an OWL 2 ontology O if an interpretation Int1 = (ΔInt, ΔD, ⋯ C, ⋯ OP, ⋯ DP, ⋯ I, ⋯ DT, ⋯ LT, ⋯ FA) exists such that Int1 coincides with Int on all named individuals and Int1 satisfies O. Thus, an interpretation Int satisfying O is also a model of O. In contrast, a model Int of O may not satisfy O directly; however, by modifying the interpretation of anonymous individuals, Int can always be coerced into an interpretation Int1 that satisfies O.

2.5 Inference Problems

Let D be a datatype map and V a vocabulary over D. Furthermore, let O and O1 be OWL 2 ontologies, CE, CE1, and CE2 class expressions, and a a named individual, such that all of them refer only to the vocabulary elements in V. A Boolean conjunctive query Q is a closed formula of the form

\[ \exists x_1, \ldots, x_n, y_1, \ldots, y_m : [ \bigwedge A_1 \land \ldots \land A_k ] \]

where each \( A_i \) is an atom of the form \( C(s) \), \( OP(s, t) \), or \( DP(s, u) \) with \( C \) a class, \( OP \) an object property, \( DP \) a data property, \( s \) and \( t \) individuals or some variable \( x_j \), and \( u \) a literal or some variable \( y_j \).

The following inference problems are often considered in practice.

Ontology Consistency: O is consistent (or satisfiable) w.r.t. D if a model of O w.r.t. D and V exists.

Ontology Entailment: O entails O1 w.r.t. D if every model of O w.r.t. D and V is also a model of O1 w.r.t. D and V.

Ontology Equivalence: O and O1 are equivalent w.r.t. D if O entails O1 w.r.t. D and O1 entails O w.r.t. D.

Ontology Equisatisfiability: O and O1 are equisatisfiable w.r.t. D if O is satisfiable w.r.t. D and only if O1 is satisfiable w.r.t. D.
Class Expression Satisfiability: \( CE \) is satisfiable w.r.t. \( O \) and \( D \) if a model \( Int = ( \Delta_{Int}, \Delta_D, C, DP, OP, I, DT, LT, FA ) \) of \( O \) w.r.t. \( D \) and \( V \) exists such that \((CE)^D \neq \emptyset\).

Class Expression Subsumption: \( CE_1 \) is subsumed by a class expression \( CE_2 \) w.r.t. \( O \) and \( D \) if \((CE_1)^C \subseteq (CE_2)^C\) for each model \( Int = ( \Delta_{Int}, \Delta_D, C, DP, OP, I, DT, LT, FA ) \) of \( O \) w.r.t. \( D \) and \( V \).

Instance Checking: \( a \) is an instance of \( CE \) w.r.t. \( O \) and \( D \) if \( a \in Int(CE) \) for each model \( Int = ( \Delta_{Int}, \Delta_D, C, DP, OP, I, DT, LT, FA ) \) of \( O \) w.r.t. \( D \) and \( V \).

Boolean Conjunctive Query Answering: \( Q \) is an answer w.r.t. \( O \) and \( D \) if \( Q \) is true in each model of \( O \) w.r.t. \( D \) and \( V \).

In order to ensure that ontology entailment, class expression satisfiability, class expression subsumption, and instance checking are decidable, the following restriction w.r.t. \( O \) needs to be satisfied:

Each class expression of type MinObjectCardinality, MaxObjectCardinality, ExactObjectCardinality, and ObjectHasSelf that occurs in \( O \) and \( CE_2 \) can contain only object property expressions that are simple in the axiom closure \( Ax \) of \( O \).

For ontology equivalence to be decidable, \( O_1 \) needs to satisfy this restriction w.r.t. \( O \) and vice versa. These restrictions are analogous to the first condition from Section 11.2 of the OWL 2 Specification [OWL 2 Specification].

3 Independence of the Semantics from the Datatype Map

The semantics of OWL 2 has been defined in such a way that the semantics of an OWL 2 ontology \( O \) does not depend on the choice of a datatype map, as long as the datatype map chosen contains all the datatypes occurring in \( O \). This statement is made precise by the following theorem, which has several useful consequences:

- One can interpret an OWL 2 ontology \( O \) by considering only the datatypes explicitly occurring in \( O \).
- When referring to various reasoning problems, the datatype map \( D \) need not be given explicitly, as it is sufficient to consider an implicit datatype map containing only the datatypes from the given ontology.
- OWL 2 reasoners can provide datatype results not explicitly mentioned in this specification without fear that this will change the semantics of OWL 2 ontologies not using these datatypes.

**Theorem DS1.** Let \( O_1 \) and \( O_2 \) be OWL 2 ontologies over a vocabulary \( V \) and \( D = ( ND_D, NL_D, NF_D, DT_D, LT_D, FA_D ) \) a datatype map such that each datatype mentioned in \( O_1 \) and \( O_2 \) is either rdfs:Literal or it occurs in \( ND_D \). Furthermore, let \( D' = ( ND_D', NL_D', NF_D', DT_D', LT_D', FA_D' ) \) be a datatype map such that \( ND_D \subseteq ND_D' \), \( NL_D(DT) = NL_D'(DT) \), and \( NF_D(DT) = NF_D'(DT) \) for each \( DT \in ND_D \), and \( DT_D' \).
\( \cdot \wedge S \cdot \) and \( \cdot \wedge F \cdot \) are extensions of \( \cdot DT \cdot \), \( \cdot \wedge S \cdot \), and \( \cdot \wedge F \cdot \), respectively. Then, \( \mathcal{O}_1 \) entails \( \mathcal{O}_2 \) w.r.t. \( D \) if and only if \( \mathcal{O}_1 \) entails \( \mathcal{O}_2 \) w.r.t. \( D' \).

**Proof.** Without loss of generality, one can assume \( \mathcal{O}_1 \) and \( \mathcal{O}_2 \) to be in negation-normal form [Description Logics]. The claim of the theorem is equivalent to the following statement: an interpretation \( \text{Int} \) w.r.t. \( D \) and \( V \) exists such that \( \mathcal{O}_1 \) is and \( \mathcal{O}_2 \) is not satisfied in \( \text{Int} \) if and only if an interpretation \( \text{Int}' \) w.r.t. \( D' \) and \( V \) exists such that \( \mathcal{O}_1 \) is and \( \mathcal{O}_2 \) is not satisfied in \( \text{Int}' \). The \((\Leftarrow)\) direction is trivial since each interpretation \( \text{Int} \) w.r.t. \( D' \) and \( V \) is also an interpretation w.r.t. \( D \) and \( V \). For the \((\Rightarrow)\) direction, assume that an interpretation \( \text{Int} = (\Delta \text{Int}, \Delta D, \cdot C', \cdot OP', \cdot DP', \cdot I, \cdot DT', \cdot LT', \cdot FA') \) w.r.t. \( D \) and \( V \) exists such that \( \mathcal{O}_1 \) is and \( \mathcal{O}_2 \) is not satisfied in \( \text{Int} \). Let \( \text{Int}' = (\Delta \text{Int}, \Delta D', \cdot C, \cdot OP, \cdot DP, \cdot I, \cdot DT, \cdot LT, \cdot FA) \) be an interpretation such that

- \( \Delta D' \) is obtained by extending \( \Delta D \) with the value space of all datatypes in \( N_{DT} \setminus N_{DT}' \),
- \( \cdot C \) coincides with \( \cdot C \) on all classes, and
- \( \cdot DP \) coincides with \( \cdot DP \) on all data properties apart from \( \text{owl:topDataProperty} \).

Clearly, \( \text{ComplementOf}(\text{DR})_{DT} \subseteq \text{ComplementOf}(\text{DR})_{DT'} \) for each data range \( DR \) that is is either a datatype, a datatype restriction, or an enumerated data range. The \( \text{owl:topDataProperty} \) property can occur in \( \mathcal{O}_1 \) and \( \mathcal{O}_2 \) only in tautologies. The interpretation of all other data properties is the same in \( \text{Int} \) and \( \text{Int}' \), so \( (CE)^{\mathcal{C}} = (CE)^{\mathcal{C}'} \) for each class expression \( CE \) occurring in \( \mathcal{O}_1 \) and \( \mathcal{O}_2 \). Therefore, \( \mathcal{O}_1 \) is and \( \mathcal{O}_2 \) is not satisfied in \( \text{Int}' \). QED

**4 Acknowledgments**

The starting point for the development of OWL 2 was the [OWL 1.1 member submission](http://www.w3.org/TR/2008/WD-owl-direct-semantics-20081202/), itself a result of user and developer feedback, and in particular of information gathered during the [OWL Experiences and Directions (OWLED) Workshop series](http://www.w3.org/TR/2008/WD-owl-direct-semantics-20081202/). The working group also considered postponed issues from the WebOnt Working Group.

This document is the product of the OWL Working Group (see below) whose members deserve recognition for their time and commitment. The editors extend special thanks to Markus Krötzsch (FZI), Michael Schneider (FZI) and Thomas Schneider (University of Manchester) for their thorough reviews.

The regular attendees at meetings of the OWL Working Group at the time of publication of this document were: Jie Bao (RPI), Diego Calvanese (Free University of Bozen-Bolzano), Bernardo Cuenca Grau (Oxford University), Martin Dzbor (Open University), Achille Fokoue (IBM Corporation), Christine Golbreich (Université de Versailles St-Quentin), Sandro Hawke (W3C/MIT), Ivan Herman (W3C/ERCIM), Rinke Hoekstra (University of Amsterdam), Ian Horrocks (Oxford University), Elisa Kendall (Sandpiper Software), Markus Krötzsch (FZI), Carsten Lutz (Universität Bremen), Boris Motik (Oxford University), Jeff Pan (University of
Aberdeen), Bijan Parsia (University of Manchester), Peter F. Patel-Schneider (Bell Labs Research, Alcatel-Lucent), Alan Ruttenberg (Science Commons), Uli Sattler (University of Manchester), Michael Schneider (FZI), Mike Smith (Clark & Parsia), Evan Wallace (NIST), and Zhe Wu (Oracle Corporation). We would also like to thank past members of the working group: Jeremy Carroll, Jim Hendler and Vipul Kashyap.

5 References

[Description Logics]

[OWL 2 Specification]

[OWL 2 Profiles]

[OWL Abstract Syntax and Semantics]

[SROIQ]

[RFC-4646]