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May Be Superseded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

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 2009-01-23

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.
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 $\langle Fv \rangle$, 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 $\langle Fv \rangle \in N_{FS}(DT)$, the interpretation function $\cdot_{FS}$ assigns to $\langle Fv \rangle$ a facet value $\langle Fv \rangle^{FS} \subseteq (DT)^{DT}$.

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, $V_{DT} = N_{DT} \cup \{ \text{rdfs:Literal} \}$.

• $V_{LT}$ is a set of literals $L^\wedge DT$ for each datatype $DT \in N_{DT}$ and each lexical value $LV \in N_{LS}(DT)$.

• $V_{FA}$ is the set of pairs $(F, l, t)$ for each constraining facet $F$, datatype $DT \in N_{DT}$, and literal $l, t \in V_{LT}$ such that $(F, (L^\wedge DT l)\wedge LS) \in N_{FS}(DT)$, where $LV$ is the lexical value of $l$ and $DT_1$ is the datatype of $l$.

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 $\Delta_{Int}$ called the data domain such that $(DT)^{DT} \subseteq \Delta_{D}$ for each datatype $DT \in V_{DT}$.
• $\cdot C$ is the class interpretation function that assigns to each class $C \in VC$ a subset $(C)^{C} \subseteq \Delta_{Int}$ such that
  • $(\text{owl:Thing})^{C} = \Delta_{Int}$ 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 $(OP)^{OP} \subseteq \Delta_{Int} \times \Delta_{Int}$ such that
  • $(\text{owl:topObjectProperty})^{OP} = \Delta_{Int} \times \Delta_{Int}$ 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 \( (DP)^{DP} \subseteq \Delta_{Int} \times \Delta_{D} \) such that:
  - \( (owl:topDataProperty)^{DP} = \Delta_{Int} \times \Delta_{D} \)
  - \( (owl:bottomDataProperty)^{DP} = \emptyset \).

• \( \cdot^{I} \) is the **individual interpretation** function that assigns to each individual \( a \in V_{I} \) an element \( (a)^{I} \in \Delta_{Int} \).

• \( \cdot^{DT} \) 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:
  - \( (rdfs:Literal)^{DT} = \Delta_{D} \).

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

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

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 1. Interpreting Object Property Expressions**

<table>
<thead>
<tr>
<th>Object Property Expression</th>
<th>Interpretation ( \cdot^{OP} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>InverseOf( OP )</td>
<td>{ \langle x, y \rangle</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 \( (DT)^{DT} \subseteq \Delta_{D} \). 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 \( (DR)^{DT} \) 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>IntersectionOf( DR1 ... DRn )</td>
<td>( (DR1)^{DT} \cap ... \cap (DRn)^{DT} )</td>
</tr>
<tr>
<td>UnionOf( DR1 ... DRn )</td>
<td>( (DR1)^{DT} \cup ... \cup (DRn)^{DT} )</td>
</tr>
<tr>
<td>ComplementOf( DR )</td>
<td>( (\Delta_{D})^{n} \setminus (DR)^{DT} ) where ( n ) is the arity of ( DR )</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>
<thead>
<tr>
<th>Class Expression</th>
<th>Interpretation ( \cdot^C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IntersectionOf( CE_1 ... CE_n )</td>
<td>((CE_1)^C \cap \ldots \cap (CE_n)^C)</td>
</tr>
<tr>
<td>UnionOf( CE_1 ... CE_n )</td>
<td>((CE_1)^C \cup \ldots \cup (CE_n)^C)</td>
</tr>
<tr>
<td>ComplementOf( CE )</td>
<td>(\Delta \setminus (CE)^C)</td>
</tr>
<tr>
<td>OneOf( a_1 ... a_n )</td>
<td>{ (a_1)^I, \ldots, (a_n)^I }</td>
</tr>
<tr>
<td>SomeValuesFrom( OPE CE )</td>
<td>({ x \mid \exists y: \langle x, y \rangle \in (OPE)^{OP} \text{ and } y \in (CE)^C })</td>
</tr>
<tr>
<td>AllValuesFrom( OPE CE )</td>
<td>({ x \mid \forall y: \langle x, y \rangle \in (OPE)^{OP} \text{ implies } y \in (CE)^C })</td>
</tr>
<tr>
<td>HasValue( OPE a )</td>
<td>({ x \mid \langle x, (a)^I \rangle \in (OPE)^{OP} })</td>
</tr>
<tr>
<td>HasSelf( OPE )</td>
<td>({ x \mid \langle x, x \rangle \in (OPE)^{OP} })</td>
</tr>
<tr>
<td>MinCardinality( n OPE )</td>
<td>({ x \mid # { y \mid \langle x, y \rangle \in (OPE)^{OP} } \geq n })</td>
</tr>
<tr>
<td>MaxCardinality( n OPE )</td>
<td>({ x \mid # { y \mid \langle x, y \rangle \in (OPE)^{OP} } \leq n })</td>
</tr>
<tr>
<td>ExactCardinality( n OPE )</td>
<td>({ x \mid # { y \mid \langle x, y \rangle \in (OPE)^{OP} } = n })</td>
</tr>
<tr>
<td>MinCardinality( n OPE CE )</td>
<td>({ x \mid # { y \mid \langle x, y \rangle \in (OPE)^{OP} \text{ and } y \in (CE)^C } \geq n })</td>
</tr>
<tr>
<td>MaxCardinality( n OPE CE )</td>
<td>({ x \mid # { y \mid \langle x, y \rangle \in (OPE)^{OP} \text{ and } y \in (CE)^C } \leq n })</td>
</tr>
</tbody>
</table>
2.3 Satisfaction in an Interpretation

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

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

2.3.1 Class Expression Axioms

Satisfaction of OWL 2 class expression axioms in $Int$ 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( CE₁ CE₂ )</td>
<td>$(CE₁)^C \subseteq (CE₂)^C$</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( ( \text{OPE}_1 \ \text{OPE}_2 \ ) \ )</td>
<td>( \text{SubPropertyOf}(\text{OPE}_1 \text{OPE}_2) ) (OPE)(^{\text{OP}}) \subseteq (OPE)(^{\text{OP}})</td>
</tr>
<tr>
<td>SubPropertyOf( PropertyChain( ( \text{OPE}_1 \ ... \ \text{OPE}_n \ ) \ OPE ) \ )</td>
<td>( \forall \ y_0, \ ... , y_n : \langle y_0, y_1 \rangle \in (\text{OPE})(^{\text{OP}}) ) and ... and ( \langle y_{n-1}, y_n \rangle \in (\text{OPE})(^{\text{OP}}) ) imply ( \langle y_0, y_n \rangle \in (\text{OPE})(^{\text{OP}})</td>
</tr>
<tr>
<td>EquivalentProperties( ( \text{OPE}_1 \ ... \ \text{OPE}_n \ ) \ )</td>
<td>( \text{EquivalentProperties}(\text{OPE}_1 \text{OPE}_n) ) ( \text{(OPE)}(^{\text{OP}}) = (\text{OPE})(^{\text{OP}}) for each 1 ≤ j ≤ n and each 1 ≤ k ≤ n</td>
</tr>
<tr>
<td>DisjointProperties( ( \text{OPE}_1 \ ... \ \text{OPE}_n \ ) \ )</td>
<td>( \text{DisjointProperties}(\text{OPE}_1 \text{OPE}_n) ) ( \text{(OPE)}(^{\text{OP}}) \cap (\text{OPE})(^{\text{OP}}) = ∅ for each 1 ≤ j ≤ n and each 1 ≤ k ≤ n such that j ≠ k</td>
</tr>
<tr>
<td>PropertyDomain( ( \text{OPE CE} \ ) \ )</td>
<td>( \text{PropertyDomain}(\text{OPE CE}) ) ( \forall \ x, y : \langle x, y \rangle \in (\text{OPE})(^{\text{OP}}) ) implies ( x \in (\text{CE})(^{\text{C}})</td>
</tr>
<tr>
<td>PropertyRange( ( \text{OPE CE} \ ) \ )</td>
<td>( \text{PropertyRange}(\text{OPE CE}) ) ( \forall \ x, y : \langle x, y \rangle \in (\text{OPE})(^{\text{OP}}) ) implies ( y \in (\text{CE})(^{\text{C}})</td>
</tr>
<tr>
<td>InverseProperties( ( \text{OPE}_1 \ \text{OPE}_2 \ ) \ )</td>
<td>( \text{InverseProperties}(\text{OPE}_1 \text{OPE}_2) ) ( \text{(OPE)}(^{\text{OP}}) = { \langle x, y \rangle</td>
</tr>
<tr>
<td>FunctionalProperty( ( \text{OPE} \ ) \ )</td>
<td>( \text{FunctionalProperty}(\text{OPE}) ) ( \forall \ x, y_1, y_2 : \langle x, y_1 \rangle \in (\text{OPE})(^{\text{OP}}) ) and ( \langle x, y_2 \rangle \in (\text{OPE})(^{\text{OP}}) ) imply ( y_1 = y_2 )</td>
</tr>
<tr>
<td>InverseFunctionalProperty( ( \text{OPE} \ ) \ )</td>
<td>( \text{InverseFunctionalProperty}(\text{OPE}) ) ( \forall \ x_1, x_2, y : \langle x_1, y \rangle \in (\text{OPE})(^{\text{OP}}) ) and ( \langle x_2, y \rangle \in (\text{OPE})(^{\text{OP}}) ) imply ( x_1 = x_2 )</td>
</tr>
<tr>
<td>ReflexiveProperty( ( \text{OPE} \ ) \ )</td>
<td>( \text{ReflexiveProperty}(\text{OPE}) ) ( \forall \ x : x \in \Delta_{\text{Int}} ) implies ( \langle x, x \rangle \in (\text{OPE})(^{\text{OP}})</td>
</tr>
<tr>
<td>IrreflexiveProperty( ( \text{OPE} \ ) \ )</td>
<td>( \text{IrreflexiveProperty}(\text{OPE}) ) ( \forall \ x : x \in \Delta_{\text{Int}} ) implies ( \langle x, x \rangle \notin (\text{OPE})(^{\text{OP}})</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 7. Satisfaction of Data Property Expression Axioms in an Interpretation

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubPropertyOf( DPE(_1)...DPE(_n) )</td>
<td>((DPE(_1))^{DP} \subseteq (DPE(_2))^{DP})</td>
</tr>
<tr>
<td>EquivalentProperties( DPE(_1)...DPE(_n) )</td>
<td>((DPE(_j))^{DP} = (DPE(_k))^{DP}) for each (1 \leq j \leq n) and each (1 \leq k \leq n)</td>
</tr>
<tr>
<td>DisjointProperties( DPE(_1)...DPE(_n) )</td>
<td>((DPE(_j))^{DP} \cap (DPE(_k))^{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 : (x, y) \in (DPE)^{DP} \text{ implies } x \in (CE)^{C})</td>
</tr>
<tr>
<td>PropertyRange( DPE DR )</td>
<td>(\forall x, y : (x, y) \in (DPE)^{DP} \text{ implies } y \in (DR)^{DT})</td>
</tr>
<tr>
<td>FunctionalProperty( DPE )</td>
<td>(\forall x, y_1, y_2 : (x, y_1) \in (DPE)^{DP} \text{ and } (x, y_2) \in (DPE)^{DP} \text{ 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>
<tbody>
<tr>
<td>HasKey( PE(_1)...PE(_n) )</td>
<td>(\forall x, y, z_1, \ldots, z_n : )</td>
</tr>
<tr>
<td></td>
<td>if (\text{ISNAMED}_O(x)) and (\text{ISNAMED}_O(y)) and (\text{ISNAMED}_O(z_1)) and ... and (\text{ISNAMED}_O(z_n)) and (x \in (CE)^{C}) and (y \in (CE)^{C}) and for each (1 \leq i \leq n), (z_i \neq y).</td>
</tr>
</tbody>
</table>
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 \( Int \) 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 \ldots a_n ) )</td>
<td>( \langle a_j \rangle^I = \langle a_k \rangle^I ) for each ( 1 \leq j \leq n ) and each ( 1 \leq k \leq n )</td>
</tr>
<tr>
<td>DifferentIndividuals( ( a_1 \ldots a_n ) )</td>
<td>( \langle a_j \rangle^I \neq \langle a_k \rangle^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>( \langle a \rangle^I \in (CE)^C )</td>
</tr>
<tr>
<td>PropertyAssertion( OPE ( a_1 a_2 ) )</td>
<td>( \langle \langle a_1 \rangle^I, \langle a_2 \rangle^I \rangle \in (OPE)^{OP} )</td>
</tr>
<tr>
<td>NegativePropertyAssertion( OPE ( a_1 a_2 ) )</td>
<td>( \langle \langle a_1 \rangle^I, \langle a_2 \rangle^I \rangle \notin (OPE)^{OP} )</td>
</tr>
<tr>
<td>PropertyAssertion( DPE ( a ) )</td>
<td>( \langle \langle a \rangle^I, \langle LL \rangle^T \rangle \in (DPE)^{DP} )</td>
</tr>
<tr>
<td>NegativePropertyAssertion( DPE ( a ) )</td>
<td>( \langle \langle a \rangle^I, \langle LL \rangle^T \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 \( Int = (\Delta_{Int}, \Delta_D, \cdot^C, \cdot^{OP}, \cdot^{DP}, \cdot^I, \cdot^{DT}, \cdot^{LT}, \cdot^{FA}) \) is a model of an OWL 2 ontology \( O \) if an interpretation \( Int_1 = (\Delta_{Int}, \Delta_D, \cdot^C, \cdot^{OP}, \cdot^{DP}, \cdot^I, \cdot^{DT}, \cdot^{LT}, \cdot^{FA}) \) exists such that \( \cdot^I \) coincides with \( \cdot^I_1 \) on all named individuals and \( Int_1 \) satisfies \( O \).
Thus, an interpretation $\text{Int}$ satisfying $\text{O}$ is also a model of $\text{O}$. In contrast, a model $\text{Int}$ of $\text{O}$ may not satisfy $\text{O}$ directly; however, by modifying the interpretation of anonymous individuals, $\text{Int}$ can always be coerced into an interpretation $\text{Int}_1$ that satisfies $\text{O}$.

2.5 Inference Problems

Let $D$ be a datatype map and $V$ a vocabulary over $D$. Furthermore, let $\text{O}$ and $\text{O}_1$ be OWL 2 ontologies, $\text{CE}$, $\text{CE}_1$, and $\text{CE}_2$ 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

$$\exists x_1, \ldots, x_n, y_1, \ldots, y_m : \left[ A_1 \land \ldots \land A_k \right]$$

where each $A_i$ is an atom of the form $C(s)$, $\text{OP}(s,t)$, or $\text{DP}(s,u)$ with $C$ a class, $\text{OP}$ an object property, $\text{DP}$ 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:** $\text{O}$ is consistent (or satisfiable) w.r.t. $D$ if a model of $\text{O}$ w.r.t. $D$ and $V$ exists.

**Ontology Entailment:** $\text{O}$ entails $\text{O}_1$ w.r.t. $D$ if every model of $\text{O}$ w.r.t. $D$ and $V$ is also a model of $\text{O}_1$ w.r.t. $D$ and $V$.

**Ontology Equivalence:** $\text{O}$ and $\text{O}_1$ are equivalent w.r.t. $D$ if $\text{O}$ entails $\text{O}_1$ w.r.t. $D$ and $\text{O}_1$ entails $\text{O}$ w.r.t. $D$.

**Ontology Equisatisfiability:** $\text{O}$ and $\text{O}_1$ are equisatisfiable w.r.t. $D$ if $\text{O}$ is satisfiable w.r.t. $D$ if and only if $\text{O}_1$ is satisfiable w.r.t. $D$.

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

**Class Expression Subsumption:** $\text{CE}_1$ is subsumed by a class expression $\text{CE}_2$ w.r.t. $\text{O}$ and $D$ if $(\text{CE}_1)^C \subseteq (\text{CE}_2)^C$ for each model $\text{Int} = (\Delta_{\text{Int}}, \Delta_D, \cdot C, \cdot \text{OP}, \cdot \text{DP}, \cdot I, \cdot DT, \cdot LT, \cdot FA)$ of $\text{O}$ w.r.t. $D$ and $V$.

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

**Boolean Conjunctive Query Answering:** $Q$ is an answer w.r.t. $\text{O}$ and $D$ if $Q$ is true in each model of $\text{O}$ 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 \( \text{MinObjectCardinality} \), \( \text{MaxObjectCardinality} \), \( \text{ExactObjectCardinality} \), and \( \text{ObjectHasSelf} \) that occurs in \( O_1 \), \( CE \), \( CE_1 \), 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 datatypes 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 \( D = (N_{DT}, N_{LS}, N_{FS}, \cdot^{DT}, \cdot^{LS}, \cdot^{FS}) \) a datatype map such that each datatype mentioned in \( O_1 \) and \( O_2 \) is either \( \text{rdfs:Literal} \) or it occurs in \( N_{DT} \). Furthermore, let \( D' = (N_{DT}', N_{LS}', N_{FS}', \cdot^{DT}', \cdot^{LS}', \cdot^{FS}') \) be a datatype map such that \( N_{DT} \subseteq N_{DT}', N_{LS}(D) = N_{LS}'(DT) \), and \( N_{FS}(DT) = N_{FS}'(DT) \) for each \( DT \in N_{DT} \), and \( \cdot^{DT}, \cdot^{LS}, \cdot^{FS} \) are extensions of \( \cdot^{DT}', \cdot^{LS}', \cdot^{FS}' \), respectively. Then, \( O_1 \) entails \( O_2 \) w.r.t. \( D \) if and only if \( O_1 \) entails \( O_2 \) w.r.t. \( D' \).

**Proof.** Without loss of generality, one can assume \( O_1 \) and \( O_2 \) to be in negation-normal form [Description Logics]. The claim of the theorem is equivalent to the following statement: an interpretation \( Int \) w.r.t. \( D \) and \( V \) exists such that \( O_1 \) is and \( O_2 \) is not satisfied in \( Int \) if and only if an interpretation \( Int' \) w.r.t. \( D' \) and \( V \) exists such that \( O_1 \) is and \( O_2 \) is not satisfied in \( Int' \). The (\( \Rightarrow \)) direction is trivial since each interpretation \( Int \) w.r.t. \( D' \) and \( V \) is also an interpretation w.r.t. \( D \) and \( V \). For the (\( \Leftarrow \)) direction, assume that an interpretation \( Int = (\Delta_{Int}, \Delta_D, \cdot^{C}, \cdot^{OP}, \cdot^{DP}, \cdot^{I}, \cdot^{DT}, \cdot^{LT}, \cdot^{FA}) \) w.r.t. \( D \) and \( V \) exists such that \( O_1 \) is and \( O_2 \) is not satisfied in \( Int \). Let \( Int' = (\Delta_{Int}, \Delta_D', \cdot^{C}, \cdot^{OP}, \cdot^{DP}, \cdot^{I}, \cdot^{DT}, \cdot^{LT}, \cdot^{FA}) \) be an interpretation such that
ΔD′ is obtained by extending ΔD with the value space of all datatypes in NDT \ NDT,
• ⋈D coincides with ⋈C on all classes, and
• ⋈DP coincides with ⋈DP on all data properties apart from owl:topDataProperty.

Clearly, ComplementOf( DR )DT ⊆ ComplementOf( DR )DT′ for each data range DR that is either a datatype, a datatype restriction, or an enumerated data range. The owl:topDataProperty property can occur in O1 and O2 only in tautologies. The interpretation of all other data properties is the same in Int and Int′, so (CE)C = (CE)C′ for each class expression CE occurring in O1 and O2. Therefore, O1 is and O2 is not satisfied in Int′. QED

4 Acknowledgments

The starting point for the development of OWL 2 was the OWL1.1 member submission, itself a result of user and developer feedback, and in particular of information gathered during the OWL Experiences and Directions (OWLED) Workshop series. 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]