

# TEX-OWL: a Latex-Style Syntax for authoring OWL 2 ontologies

Matteo Matassoni  
University of Trento – DISI  
#150629  
<m.matassoni.1@studenti.unitn.it>

## Contents

<b>1</b>	<b>Preliminary Definitions</b>	<b>1</b>
<b>2</b>	<b>General Definitions</b>	<b>1</b>
<b>3</b>	<b>Including additional input files</b>	<b>2</b>
<b>4</b>	<b>Ontologies</b>	<b>2</b>
<b>5</b>	<b>Annotations</b>	<b>2</b>
5.1	Annotations of Ontologies, Axioms, and other Annotations . . .	2
5.2	Annotation Axioms . . . . .	3
5.2.1	Annotation Assertion . . . . .	3
5.2.2	Annotation Subproperties . . . . .	3
5.2.3	Annotation Property Domain . . . . .	3
5.2.4	Annotation Property Range . . . . .	3
<b>6</b>	<b>Entities, Literals, and Anonymous Individuals</b>	<b>3</b>
6.1	Classes . . . . .	3
6.2	Datatype . . . . .	3
6.3	ObjectProperty . . . . .	3
6.4	DataProperty . . . . .	3
6.5	AnnotationProperty . . . . .	4
6.6	Individual . . . . .	4
6.6.1	Named Individuals . . . . .	4
6.6.2	Anonymous Individuals . . . . .	4
6.7	Literals . . . . .	4
6.8	Entity Declarations and Typing . . . . .	4

<b>7</b>	<b>Property Expressions</b>	<b>4</b>
7.1	Object Property Expressions . . . . .	4
7.1.1	Inverse Object Properties . . . . .	5
7.2	Data Property Expressions . . . . .	5
<b>8</b>	<b>Data Ranges</b>	<b>5</b>
8.1	Atomic Data Ranges . . . . .	5
8.1.1	Sequence Intersection of Data Ranges . . . . .	5
8.1.2	Sequence Union of Data Ranges . . . . .	5
8.1.3	Enumeration of Literals . . . . .	5
8.2	Non-atomic Data Ranges . . . . .	5
8.2.1	Intersection of Data Ranges . . . . .	5
8.2.2	Union of Data Ranges . . . . .	5
8.2.3	Complement of Data Ranges . . . . .	6
8.2.4	Datatype Restrictions . . . . .	6
<b>9</b>	<b>Class Expressions</b>	<b>6</b>
9.1	Atomic Class Expression . . . . .	6
9.1.1	Propositional Connectives and Enumeration of Individuals	6
	Sequence Intersection of Class Expressions . . . . .	6
	Sequence Union of Class Expressions . . . . .	6
	Enumeration of Individuals . . . . .	6
9.1.2	Object Property Restrictions . . . . .	7
	Existential Quantification . . . . .	7
	Universal Quantification . . . . .	7
	Individual Value Restriction . . . . .	7
9.1.3	Object Property Cardinality Restrictions . . . . .	7
	Minimum Cardinality . . . . .	7
	Maximum Cardinality . . . . .	7
	Exact Cardinality . . . . .	7
9.1.4	Data Property Restrictions . . . . .	7
	Existential Quantification . . . . .	7
	Universal Quantification . . . . .	7
	Literal Value Restriction . . . . .	7
9.1.5	Data Property Cardinality Restrictions . . . . .	8
	Minimum Cardinality . . . . .	8
	Maximum Cardinality . . . . .	8
	Exact Cardinality . . . . .	8
9.2	Non-atomic Class Expression . . . . .	8
9.2.1	Propositional Connectives . . . . .	8
	Intersection of Class Expressions . . . . .	8
	Union of Class Expressions . . . . .	8
	Complement of Class Expressions . . . . .	8
9.2.2	Object Property Restrictions . . . . .	8
	Self-Restriction . . . . .	8

<b>10 Axiom</b>	<b>8</b>
10.1 Class Expression Axioms . . . . .	9
10.1.1 Non-sequence Class Expression Axioms . . . . .	9
Subclass Axioms . . . . .	9
Equivalent Classes . . . . .	9
Disjoint Classes . . . . .	9
Disjoint Union of Class Expression . . . . .	9
10.1.2 Sequence Class Expression Axioms . . . . .	9
Sequence Equivalent Classes . . . . .	9
Sequence Disjoint Classes . . . . .	9
10.2 Object Property Axioms . . . . .	9
10.2.1 Non-sequence Object Property Axioms . . . . .	10
Object Subproperties . . . . .	10
Equivalent Object Properties . . . . .	10
Disjoint Object Properties . . . . .	10
Inverse Object Properties . . . . .	10
Object Property Domain . . . . .	10
Object Property Range . . . . .	10
Functional Object Properties . . . . .	10
Inverse-Functional Object Properties . . . . .	11
Reflexive Object Properties . . . . .	11
Irreflexive Object Properties . . . . .	11
Symmetric Object Properties . . . . .	11
Asymmetric Object Properties . . . . .	11
Transitive Object Properties . . . . .	11
10.2.2 Sequence Object Property Axioms . . . . .	11
Sequence Equivalent Object Properties . . . . .	11
Sequence Disjoint Object Properties . . . . .	11
10.3 Data Property Axioms . . . . .	11
10.3.1 Non-sequence Object Property Axioms . . . . .	11
Data Subproperties . . . . .	12
Equivalent Data Properties . . . . .	12
Disjoint Data Properties . . . . .	12
Data Property Domain . . . . .	12
Data Property Range . . . . .	12
Functional Data Properties . . . . .	12
10.3.2 Sequence Object Property Axioms . . . . .	12
Equivalent Data Properties . . . . .	12
Disjoint Data Properties . . . . .	12
10.4 Datatype Definitions . . . . .	12
10.5 Keys . . . . .	13
10.6 Assertion . . . . .	13
10.6.1 Non-sequence Assertion . . . . .	13
Individual Equality . . . . .	13
Individual Inequality . . . . .	13
Class Assertions . . . . .	13

	Positive Object Property Assertions . . . . .	13
	Negative Object Property Assertions . . . . .	13
	Positive Data Property Assertions . . . . .	13
	Negative Data Property Assertions . . . . .	14
10.6.2	Sequence Assertion . . . . .	14
	Sequence Individual Equality . . . . .	14
	Individual Inequality . . . . .	14
<b>11</b>	<b>Sample Ontology</b>	<b>14</b>

# 1 Preliminary Definitions

The grammar presented in this document uses the following two “special” terminal symbols, which affect the process of transforming an input sequence of characters into a sequence of regular (i.e., not “special”) terminal symbols:

- whitespace is a nonempty sequence of space (U+20), horizontal tab (U+9), line feed (U+A), or carriage return (U+D) characters, and
- a comment is a sequence of characters that starts with the % (U+25) character and does not contain the line feed (U+A) or carriage return (U+D) characters.

# 2 General Definitions

$\langle nonNegativeInteger \rangle ::=$  a nonempty finite sequence of digits between 0 and 9

$\langle quotedString \rangle ::=$  a finite sequence of characters in which ‘ (U+22) and \ (U+5C) occur only in pairs of the form \‘ (U+5C, U+22) and \\ (U+5C, U+5C), enclosed in a pair of ‘ (U+22) characters

$\langle languageTag \rangle ::=$  @ (U+40) followed a nonempty sequence of characters matching the langtag production from [?]

$\langle nodeID \rangle ::=$  a finite sequence of characters matching the BLANK\_NODE\_LABEL production of [?]

$\langle fullIRI \rangle ::=$  an IRI as defined in [?], enclosed in a pair of < (U+3C) and > (U+3E) characters

$\langle prefixName \rangle ::=$  a finite sequence of characters matching the as PNAME\_NS production of [?] and not matching any of the keyword terminals of the syntax

$\langle abbreviatedIRI \rangle ::=$  a finite sequence of characters matching the PNAME\_LN production of [?] and not matching any of the keyword terminals of the syntax

$\langle simpleIRI \rangle ::=$  a finite sequence of characters matching the PN\_LOCAL production of [?] and not matching any of the keyword terminals of the syntax

$\langle IRI \rangle := \langle fullIRI \rangle \mid \langle abbreviatedIRI \rangle \mid \langle simpleIRI \rangle$

### 3 Including additional input files

Inclusion of additional input files (both local and distributed) allows to organize the ontology in a modular fashion. To specify the input file the following format should be applied:

$\langle includeExternalFile \rangle ::= \text{'\input' '{' } \langle INPUT\_FILE \rangle \text{'}'}$

$\langle INPUT\_FILE \rangle ::= \langle URL \rangle \mid \langle ABSOLUTE\_FILEPATH \rangle$

Please note that this command may occur anywhere inside the ontology document.

### 4 Ontologies

$\langle ontologyDocument \rangle ::= \langle defaultPrefixDeclaration \rangle? \langle prefixDeclaration \rangle^* \langle Ontology \rangle$

$\langle defaultPrefixDeclaration \rangle ::= \text{'\ns' } \langle fullIRI \rangle$

$\langle prefixDeclaration \rangle ::= \text{'\ns' } \langle prefixName \rangle \langle fullIRI \rangle$

$\langle Ontology \rangle ::= \text{'\begin' '{' 'ontology' '}' } \langle ontologyIRIs \rangle ( \langle directlyImportsDocument \rangle \mid \langle ontologyAnnotation \rangle )^* \langle axioms \rangle \text{'\end' '{' 'ontology' '}'}$

$\langle ontologyIRIs \rangle ::= [ \text{'[' } \langle ontologyIRI \rangle [ \text{' ,' } \langle versionIRI \rangle ] \text{' } ]$

$\langle ontologyIRI \rangle ::= \langle IRI \rangle$

$\langle versionIRI \rangle ::= \langle IRI \rangle$

$\langle directlyImportsDocument \rangle ::= \text{'\import' } \langle IRI \rangle$

$\langle ontologyAnnotation \rangle ::= \langle Annotation \rangle$

$\langle axioms \rangle ::= \langle Axiom \rangle^*$

Please note that prefixes: *owl:*, *rdf:*, *rdfs:*, *xml:* and *xsd:* are built-in and should not be explicitly declared.

### 5 Annotations

#### 5.1 Annotations of Ontologies, Axioms, and other Annotations

$\langle Annotation \rangle ::= \text{'\a' '{' } \langle AnnotationProperty \rangle \text{' ,' } \langle AnnotationValue \rangle \text{'}' } \langle annotationAnnotations \rangle$

$\langle annotationAnnotations \rangle ::= [ \text{'[' } \langle Annotation \rangle ( \text{' ,' } \langle Annotation \rangle )^* \text{' } ]$

$\langle AnnotationValue \rangle ::= \langle AnonymousIndividual \rangle \mid \langle IRI \rangle \mid \langle Literal \rangle$

## 5.2 Annotation Axioms

$$\begin{aligned} \langle AnnotationAxiom \rangle &::= \langle AnnotationAssertion \rangle \\ &| \langle SubAnnotationPropertyOf \rangle \\ &| \langle AnnotationPropertyDomain \rangle \\ &| \langle AnnotationPropertyRange \rangle \end{aligned}$$

### 5.2.1 Annotation Assertion

$$\langle AnnotationAssertion \rangle ::= \langle AnnotationSubject \rangle \text{ '\a' } \langle AnnotationProperty \rangle \text{ ', ' } \langle AnnotationValue \rangle \text{ '}' } \langle axiomAnnotations \rangle$$
$$\langle AnnotationSubject \rangle ::= \langle IRI \rangle | \langle AnonymousIndividual \rangle$$

### 5.2.2 Annotation Subproperties

$$\langle SubAnnotationPropertyOf \rangle ::= \langle subAnnotationProperty \rangle \text{ '\aisa' } \langle superAnnotationProperty \rangle \langle axiomAnnotations \rangle$$
$$\langle subAnnotationProperty \rangle ::= \langle AnnotationProperty \rangle$$
$$\langle superAnnotationProperty \rangle ::= \langle AnnotationProperty \rangle$$

### 5.2.3 Annotation Property Domain

$$\langle AnnotationPropertyDomain \rangle ::= \langle AnnotationProperty \rangle \text{ '\adomain' } \langle IRI \rangle \langle axiomAnnotations \rangle$$

### 5.2.4 Annotation Property Range

$$\langle AnnotationPropertyRange \rangle ::= \langle AnnotationProperty \rangle \text{ '\arange' } \langle IRI \rangle \langle axiomAnnotations \rangle$$

## 6 Entities, Literals, and Anonymous Individuals

### 6.1 Classes

$$\langle Class \rangle ::= \langle IRI \rangle$$

### 6.2 Datatype

$$\langle Datatype \rangle ::= \langle IRI \rangle$$

### 6.3 ObjectProperty

$$\langle ObjectProperty \rangle ::= \langle IRI \rangle$$

### 6.4 DataProperty

$$\langle DataProperty \rangle ::= \langle IRI \rangle$$

## 6.5 AnnotationProperty

$\langle AnnotationProperty \rangle ::= \langle IRI \rangle$

## 6.6 Individual

$\langle Individual \rangle ::= \langle NamedIndividual \rangle \mid \langle AnonymousIndividual \rangle$

### 6.6.1 Named Individuals

$\langle NamedIndividual \rangle ::= \langle IRI \rangle$

### 6.6.2 Anonymous Individuals

$\langle AnonymousIndividual \rangle ::= \langle nodeID \rangle$

## 6.7 Literals

$\langle Literal \rangle ::= \langle typedLiteral \rangle \mid \langle stringLiteralNoLanguage \rangle \mid \langle stringLiteralWithLanguage \rangle$

$\langle typedLiteral \rangle ::= \langle lexicalForm \rangle \text{ '}' \langle Datatype \rangle \text{ '}'$

$\langle lexicalForm \rangle ::= \langle quotedString \rangle$

$\langle stringLiteralNoLanguage \rangle ::= \langle quotedString \rangle$

$\langle stringLiteralWithLanguage \rangle ::= \langle quotedString \rangle \text{ '}' \langle languageTag \rangle \text{ '}'$

## 6.8 Entity Declarations and Typing

$\langle Declaration \rangle ::= \langle Entity \rangle \langle axiomAnnotations \rangle$

$\langle Entity \rangle ::= \langle Class \rangle \text{ '}\backslash\text{c'}$   
|  $\langle Datatype \rangle \text{ '}\backslash\text{dt'}$   
|  $\langle ObjectProperty \rangle \text{ '}\backslash\text{o'}$   
|  $\langle DataProperty \rangle \text{ '}\backslash\text{d'}$   
|  $\langle AnnotationProperty \rangle \text{ '}\backslash\text{a'}$   
|  $\langle NamedIndividual \rangle \text{ '}\backslash\text{i'}$

## 7 Property Expressions

### 7.1 Object Property Expressions

$\langle ObjectPropertyExpression \rangle ::= \langle ObjectProperty \rangle$   
|  $\text{ '}' \langle InverseObjectProperty \rangle \text{ '}'$



### 7.1.1 Inverse Object Properties

$\langle \text{InverseObjectProperty} \rangle ::= \text{'\oinvof' } \langle \text{ObjectProperty} \rangle$

## 7.2 Data Property Expressions

$\langle \text{DataPropertyExpression} \rangle ::= \langle \text{DataProperty} \rangle$

## 8 Data Ranges

$\langle \text{DataRange} \rangle ::= \langle \text{Datatype} \rangle$   
|  $\langle \text{AtomicDataRange} \rangle$   
|  $\text{'(' } \langle \text{NonAtomicDataRange} \rangle \text{'}'$

### 8.1 Atomic Data Ranges

$\langle \text{AtomicDataRange} \rangle ::= \langle \text{SequenceDataIntersectionOf} \rangle$   
|  $\langle \text{SequenceDataUnionOf} \rangle$   
|  $\langle \text{DataOneOf} \rangle$

#### 8.1.1 Sequence Intersection of Data Ranges

$\langle \text{SequenceDataIntersectionOf} \rangle ::= \text{'\drandof' } \text{'{' } \langle \text{DataRange} \rangle \text{' , ' } \langle \text{DataRange} \rangle \text{' } + \text{'}'$

#### 8.1.2 Sequence Union of Data Ranges

$\langle \text{SequenceDataUnionOf} \rangle ::= \text{'\drorof' } \text{'{' } \langle \text{DataRange} \rangle \text{' ( ' , ' } \langle \text{DataRange} \rangle \text{' ) + '}'$

#### 8.1.3 Enumeration of Literals

$\langle \text{DataOneOf} \rangle ::= \text{'\droneof' } \text{'{' } \langle \text{Literal} \rangle \text{' ( ' , ' } \langle \text{Literal} \rangle \text{' } * \text{'}'$

## 8.2 Non-atomic Data Ranges

$\langle \text{NonAtomicDataRange} \rangle ::= \langle \text{DataIntersectionOf} \rangle$   
|  $\langle \text{DataUnionOf} \rangle$   
|  $\langle \text{DataComplementOf} \rangle$   
|  $\langle \text{DatatypeRestriction} \rangle$

#### 8.2.1 Intersection of Data Ranges

$\langle \text{DataIntersectionOf} \rangle ::= \langle \text{DataRange} \rangle \text{'\drand' } \langle \text{DataRange} \rangle$

#### 8.2.2 Union of Data Ranges

$\langle \text{DataUnionOf} \rangle ::= \langle \text{DataRange} \rangle \text{'\dror' } \langle \text{DataRange} \rangle$

### 8.2.3 Complement of Data Ranges

$\langle DataComplementOf \rangle ::= \text{'\drnot'} \langle DataRange \rangle$

### 8.2.4 Datatype Restrictions

$\langle DatatypeRestriction \rangle ::= \langle Datatype \rangle \text{'\drres'} \langle DatatypeRestrictionExpression \rangle$

$\langle DatatypeRestrictionExpression \rangle ::= \text{'\{'} \langle constrainingFacet \rangle \langle restrictionValue \rangle$   
 $(\text{'\text{'}} \langle constrainingFacet \rangle \langle restrictionValue \rangle \text{'\text{'}})^* \text{'\}'}$

$\langle constrainingFacet \rangle ::= \langle IRI \rangle$

$\langle restrictionValue \rangle ::= \langle Literal \rangle$

## 9 Class Expressions

$\langle ClassExpression \rangle ::= \langle Class \rangle$   
 $\quad | \langle AtomicClassExpression \rangle$   
 $\quad | \text{'('} \langle NonAtomicClassExpression \rangle \text{'\text{'}}$

### 9.1 Atomic Class Expression

#### 9.1.1 Propositional Connectives and Enumeration of Individuals

$\langle AtomicClassExpression \rangle ::= \langle SequenceObjectIntersectionOf \rangle | \langle SequenceObjectUnionOf \rangle$   
 $\quad | \langle ObjectOneOf \rangle$   
 $\quad | \langle ObjectSomeValueFrom \rangle | \langle ObjectAllValueFrom \rangle | \langle ObjectHasValue \rangle$   
 $\quad | \langle ObjectMinCardinality \rangle | \langle ObjectMaxCardinality \rangle | \langle ObjectExactCardinality \rangle$   
 $\quad | \langle DataSomeValueFrom \rangle | \langle DataAllValueFrom \rangle | \langle DataHasValue \rangle |$   
 $\quad | \langle DataMinCardinality \rangle | \langle DataMaxCardinality \rangle | \langle DataExactCardinality \rangle$

#### Sequence Intersection of Class Expressions

$\langle SequenceObjectIntersectionOf \rangle ::= \text{'\candof'} \text{'\{' } \langle ClassExpression \rangle (\text{'\text{'}} \langle ClassExpression \rangle$   
 $\quad )^+ \text{'\}'}$

#### Sequence Union of Class Expressions

$\langle SequenceObjectUnionOf \rangle ::= \text{'\corof'} \text{'\{' } \langle ClassExpression \rangle (\text{'\text{'}} \langle ClassExpression \rangle$   
 $\quad )^+ \text{'\}'}$

#### Enumeration of Individuals

$\langle ObjectOneOf \rangle ::= \text{'\ooneof'} \text{'\{' } \langle Individual \rangle (\text{'\text{'}} \langle Individual \rangle)^* \text{'\}'}$

### 9.1.2 Object Property Restrictions

#### Existential Quantification

$\langle ObjectSomeValuesFrom \rangle ::= \text{'\oexists' '{' } \langle ObjectPropertyExpression \rangle \text{'}' '{' } \langle ClassExpression \rangle \text{'}' }$

#### Universal Quantification

$\langle ObjectAllValuesFrom \rangle ::= \text{'\forallforall' '{' } \langle ObjectPropertyExpression \rangle \text{'}' '{' } \langle ClassExpression \rangle \text{'}' }$

#### Individual Value Restriction

$\langle ObjectHasValue \rangle ::= \text{'\ohasvalue' '{' } \langle ObjectPropertyExpression \rangle \text{'}' '{' } \langle Individual \rangle \text{'}' }$

### 9.1.3 Object Property Cardinality Restrictions

#### Minimum Cardinality

$\langle ObjectMinCardinality \rangle ::= \text{'\o[>=' } \langle nonNegativeInteger \rangle \text{'}' '{' } \langle ObjectPropertyExpression \rangle \text{'}' [ \text{'{' } \langle ClassExpression \rangle \text{'}' } ]$

#### Maximum Cardinality

$\langle ObjectMaxCardinality \rangle ::= \text{'\o[<=' } \langle nonNegativeInteger \rangle \text{'}' '{' } \langle ObjectPropertyExpression \rangle \text{'}' [ \text{'{' } \langle ClassExpression \rangle \text{'}' } ]$

#### Exact Cardinality

$\langle ObjectExactCardinality \rangle ::= \text{'\o[=' } \langle nonNegativeInteger \rangle \text{'}' '{' } \langle ObjectPropertyExpression \rangle \text{'}' [ \text{'{' } \langle ClassExpression \rangle \text{'}' } ]$

### 9.1.4 Data Property Restrictions

#### Existential Quantification

$\langle DataSomeValuesFrom \rangle ::= \text{'\dexists' '{' } \langle DataPropertyExpression \rangle \text{'}' '{' } \langle DataRange \rangle \text{'}' }$

#### Universal Quantification

$\langle DataAllValuesFrom \rangle ::= \text{'\dforall' '{' } \langle DataPropertyExpression \rangle \text{'}' '{' } \langle DataRange \rangle \text{'}' }$

#### Literal Value Restriction

$\langle DataHasValue \rangle ::= \text{'\dhasvalue' '{' } \langle DataPropertyExpression \rangle \text{'}' '{' } \langle Literal \rangle \text{'}' }$

### 9.1.5 Data Property Cardinality Restrictions

#### Minimum Cardinality

$\langle DataMinCardinality \rangle ::= \text{'\d[>='} \langle nonNegativeInteger \rangle \text{'\d[>='} \text{'\{'} \langle DataPropertyExpression \rangle \text{'\}' [ \text{'\{'} \langle DataRange \rangle \text{'\}' } ]$

#### Maximum Cardinality

$\langle DataMaxCardinality \rangle ::= \text{'\d[<='} \langle nonNegativeInteger \rangle \text{'\d[<='} \text{'\{'} \langle DataPropertyExpression \rangle \text{'\}' [ \text{'\{'} \langle DataRange \rangle \text{'\}' } ]$

#### Exact Cardinality

$\langle DataExactCardinality \rangle ::= \text{'\d[='} \langle nonNegativeInteger \rangle \text{'\d[='} \text{'\{'} \langle DataPropertyExpression \rangle \text{'\}' [ \text{'\{'} \langle DataRange \rangle \text{'\}' } ]$

## 9.2 Non-atomic Class Expression

$\langle NonAtomicClassExpression \rangle ::= \langle ObjectIntersectionOf \rangle \mid \langle ObjectUnionOf \rangle \mid \langle ObjectComplementOf \rangle \mid \langle ObjectHasSelf \rangle$

### 9.2.1 Propositional Connectives

#### Intersection of Class Expressions

$\langle ObjectIntersectionOf \rangle ::= \langle ClassExpression \rangle \text{'\cand'} \langle ClassExpression \rangle$

#### Union of Class Expressions

$\langle ObjectUnionOf \rangle ::= \langle ClassExpression \rangle \text{'\cor'} \langle ClassExpression \rangle$

#### Complement of Class Expressions

$\langle ObjectComplementOf \rangle ::= \text{'\cnot'} \langle ClassExpression \rangle$

### 9.2.2 Object Property Restrictions

#### Self-Restriction

$\langle ObjectHasSelf \rangle ::= \text{'\ohasself'} \langle ObjectPropertyExpression \rangle$

## 10 Axiom

$\langle Axiom \rangle ::= \langle Declaration \rangle \mid \langle ClassAxiom \rangle \mid \langle ObjectPropertyAxiom \rangle \mid \langle DataPropertyAxiom \rangle \mid \langle DatatypeDefinition \rangle \mid \langle HasKey \rangle \mid \langle Assertion \rangle \mid \langle AnnotationAxiom \rangle$

$\langle axiomAnnotations \rangle ::= [ \text{'\['} \langle Annotation \rangle ( \text{'\,'} \langle Annotation \rangle )^* \text{'\]' } ]$

## 10.1 Class Expression Axioms

$\langle ClassAxiom \rangle ::= \langle NonSequenceClassAxiom \rangle \mid \langle SequenceClassAxiom \rangle$

### 10.1.1 Non-sequence Class Expression Axioms

$\langle NonSequenceClassAxiom \rangle ::= \langle SubClassOf \rangle \mid \langle EquivalentClasses \rangle \mid \langle DisjointClasses \rangle$   
 $\mid \langle DisjointUnion \rangle$

#### Subclass Axioms

$\langle SubClassOf \rangle ::= \langle subClassExpression \rangle \text{ '\cisa' } \langle superClassExpression \rangle \langle axiomAnnotations \rangle$

$\langle subClassExpression \rangle ::= \langle ClassExpression \rangle$

$\langle superClassExpression \rangle ::= \langle ClassExpression \rangle$

#### Equivalent Classes

$\langle EquivalentClasses \rangle ::= \langle ClassExpression \rangle \text{ '\ceq' } \langle ClassExpression \rangle \langle axiomAnnotations \rangle$

#### Disjoint Classes

$\langle DisjointClasses \rangle : = \langle ClassExpression \rangle \text{ '\cdisjoint' } \langle ClassExpression \rangle \langle axiomAnnotations \rangle$

#### Disjoint Union of Class Expression

$\langle DisjointUnion \rangle ::= \langle Class \rangle \text{ '\cdisjunction' } \langle disjointClassExpressions \rangle \langle axiomAnnotations \rangle$

$\langle disjointClassExpressions \rangle ::= \text{'\{' } \langle ClassExpression \rangle ( \text{' , ' } \langle ClassExpression \rangle )^*$   
 $\text{'\}'}$

### 10.1.2 Sequence Class Expression Axioms

$\langle SequenceClassAxiom \rangle ::= \langle SequenceEquivalentClasses \rangle \mid \langle SequenceDisjointClasses \rangle$

#### Sequence Equivalent Classes

$\langle SequenceEquivalentClasses \rangle ::= \text{'\calleg' } \text{'\{' } \langle ClassExpression \rangle ( \text{' , ' } \langle ClassExpression \rangle$   
 $\text{'\}' } + \text{'\}' } \langle axiomAnnotations \rangle$

#### Sequence Disjoint Classes

$\langle SequenceDisjointClasses \rangle : = \text{'\calldisjoint' } \text{'\{' } \langle ClassExpression \rangle ( \text{' , ' } \langle ClassExpression \rangle$   
 $\text{'\}' } + \text{'\}' } \langle axiomAnnotations \rangle$

## 10.2 Object Property Axioms

$\langle ObjectPropertyAxiom \rangle ::= \langle NonSequenceObjectPropertyAxiom \rangle \mid \langle SequenceObjectPropertyAxiom \rangle$

### 10.2.1 Non-sequence Object Property Axioms

$\langle NonSequenceObjectPropertyAxiom \rangle ::= \langle SubObjectPropertyOf \rangle \mid \langle EquivalentObjectProperties \rangle$   
 $\mid \langle DisjointObjectProperties \rangle \mid \langle InverseObjectProperties \rangle \mid \langle ObjectPropertyDomain \rangle$   
 $\mid \langle ObjectPropertyRange \rangle \mid \langle FunctionalObjectProperty \rangle \mid \langle InverseFunctionalObjectProperty \rangle$   
 $\mid \langle ReflexiveObjectProperty \rangle \mid \langle IrreflexiveObjectProperty \rangle \mid \langle SymmetricObjectProperty \rangle$   
 $\mid \langle AsymmetricObjectProperty \rangle \mid \langle TransitiveObjectProperty \rangle$

#### Object Subproperties

$\langle SubObjectPropertyOf \rangle ::= \langle subObjectPropertyExpression \rangle \text{ '\oisa' } \langle superObjectPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

$\langle subObjectPropertyExpression \rangle ::= \langle ObjectPropertyExpression \rangle \mid \langle propertyExpressionChain \rangle$

$\langle propertyExpressionChain \rangle ::= \text{ '\ochain' } \{ \langle ObjectPropertyExpression \rangle ( \text{ ',' } \langle ObjectPropertyExpression \rangle )^+ \text{ '}' }$

$\langle superObjectPropertyExpression \rangle ::= \langle ObjectPropertyExpression \rangle$

#### Equivalent Object Properties

$\langle EquivalentObjectProperties \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\oeq' } \langle ObjectPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

#### Disjoint Object Properties

$\langle DisjointObjectProperties \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\odisjoint' } \langle ObjectPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

#### Inverse Object Properties

$\langle InverseObjectProperties \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\oinv' } \langle ObjectPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

#### Object Property Domain

$\langle ObjectPropertyDomain \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\odomain' } \langle ClassExpression \rangle$   
 $\langle axiomAnnotations \rangle$

#### Object Property Range

$\langle ObjectPropertyRange \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\orange' } \langle ClassExpression \rangle$   
 $\langle axiomAnnotations \rangle$

#### Functional Object Properties

$\langle FunctionalObjectProperty \rangle ::= \langle ObjectPropertyExpression \rangle \text{ '\ofunc' } \langle axiomAnnotations \rangle$

### Inverse-Functional Object Properties

$\langle \text{InverseFunctionalObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{oinvfunc} '$   
 $\langle \text{axiomAnnotations} \rangle$

### Reflexive Object Properties

$\langle \text{ReflexiveObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{oreflex} ' \langle \text{axiomAnnotations} \rangle$

### Irreflexive Object Properties

$\langle \text{IrreflexiveObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{oirreflex} ' \langle \text{axiomAnnotations} \rangle$

### Symmetric Object Properties

$\langle \text{SymmetricObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{osym} ' \langle \text{axiomAnnotations} \rangle$

### Asymmetric Object Properties

$\langle \text{AsymmetricObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{oasym} ' \langle \text{axiomAnnotations} \rangle$

### Transitive Object Properties

$\langle \text{TransitiveObjectProperty} \rangle ::= \langle \text{ObjectPropertyExpression} \rangle ' \backslash \text{otrans} ' \langle \text{axiomAnnotations} \rangle$

### 10.2.2 Sequence Object Property Axioms

$\langle \text{SequenceObjectPropertyAxiom} \rangle ::= \langle \text{SequenceEquivalentObjectProperties} \rangle$   
 $| \langle \text{SequenceDisjointObjectProperties} \rangle$

### Sequence Equivalent Object Properties

$\langle \text{SequenceEquivalentObjectProperties} \rangle ::= ' \backslash \text{oalleg} ' \{ ' \langle \text{ObjectPropertyExpression} \rangle$   
 $( ' , ' \langle \text{ObjectPropertyExpression} \rangle ) + ' \} ' \langle \text{axiomAnnotations} \rangle$

### Sequence Disjoint Object Properties

$\langle \text{SequenceDisjointObjectProperties} \rangle ::= ' \backslash \text{oalldisjoint} ' \{ ' \langle \text{ObjectPropertyExpression} \rangle$   
 $( ' , ' \langle \text{ObjectPropertyExpression} \rangle ) + ' \} ' \langle \text{axiomAnnotations} \rangle$

## 10.3 Data Property Axioms

$\langle \text{DataPropertyAxiom} \rangle ::= \langle \text{NonSequenceDataPropertyAxiom} \rangle | \langle \text{SequenceDataPropertyAxiom} \rangle$

### 10.3.1 Non-sequence Object Property Axioms

$\langle \text{NonSequenceDataPropertyAxiom} \rangle ::= \langle \text{SubDataPropertyOf} \rangle | \langle \text{EquivalentDataProperties} \rangle$   
 $| \langle \text{DisjointDataProperties} \rangle$   
 $| \langle \text{DataPropertyDomain} \rangle | \langle \text{DataPropertyRange} \rangle | \langle \text{FunctionalDataProperty} \rangle$

### Data Subproperties

$\langle SubDataPropertyOf \rangle ::= \langle subDataPropertyExpression \rangle ' \backslash disa ' \langle superDataPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

$\langle subDataPropertyExpression \rangle ::= \langle DataPropertyExpression \rangle$

$\langle superDataPropertyExpression \rangle ::= \langle DataPropertyExpression \rangle$

### Equivalent Data Properties

$\langle EquivalentDataProperties \rangle ::= \langle DataPropertyExpression \rangle ' \backslash deq ' \langle DataPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

### Disjoint Data Properties

$\langle DisjointDataProperties \rangle ::= \langle DataPropertyExpression \rangle ' \backslash ddisjoint ' \langle DataPropertyExpression \rangle$   
 $\langle axiomAnnotations \rangle$

### Data Property Domain

$\langle DataPropertyDomain \rangle ::= \langle DataPropertyExpression \rangle ' \backslash ddomain ' \langle ClassExpression \rangle$   
 $\langle axiomAnnotations \rangle$

### Data Property Range

$\langle DataPropertyRange \rangle ::= \langle DataPropertyExpression \rangle ' \backslash drange ' \langle DataRange \rangle \langle axiomAnnotations \rangle$

### Functional Data Properties

$\langle FunctionalDataProperty \rangle ::= \langle DataPropertyExpression \rangle ' \backslash dfunc ' \langle axiomAnnotations \rangle$

### 10.3.2 Sequence Object Property Axioms

$\langle SequenceDataPropertyAxiom \rangle ::= \langle SequenceEquivalentDataProperties \rangle$   
 $| \langle SequenceDisjointDataProperties \rangle$

### Equivalent Data Properties

$\langle SequenceEquivalentDataProperties \rangle ::= ' \backslash dalleg ' ' \{ ' \langle DataPropertyExpression \rangle$   
 $( ' , ' \langle DataPropertyExpression \rangle ) + ' \} ' \langle axiomAnnotations \rangle$

### Disjoint Data Properties

$\langle SequenceDisjointDataProperties \rangle ::= ' \backslash dalldisjoint ' ' \{ ' \langle DataPropertyExpression \rangle$   
 $( ' , ' \langle DataPropertyExpression \rangle ) + ' \} ' \langle axiomAnnotations \rangle$

## 10.4 Datatype Definitions

$\langle DatatypeDefinition \rangle ::= \langle Datatype \rangle ' \backslash dtdef ' \langle DataRange \rangle \langle axiomAnnotations \rangle$



## 10.5 Keys

$\langle HasKey \rangle ::= \langle ClassExpression \rangle \backslash \text{key} \langle HasKeyExpression \rangle \langle axiomAnnotations \rangle$

$\langle HasKeyExpression \rangle ::= \{ [ \langle ObjectPropertyExpression \rangle ( , \langle ObjectPropertyExpression \rangle )^* ] \} \{ [ \langle DataPropertyExpression \rangle ( , \langle DataPropertyExpression \rangle )^* ] \}$

## 10.6 Assertion

$\langle Assertion \rangle ::= \langle NonSequenceAssertion \rangle \mid \langle SequenceAssertion \rangle$

$\langle sourceIndividual \rangle ::= \langle Individual \rangle$

$\langle targetIndividual \rangle ::= \langle Individual \rangle$

$\langle targetValue \rangle ::= \langle Literal \rangle$

### 10.6.1 Non-sequence Assertion

$\langle NonSequenceAssertion \rangle ::= \langle SameIndividual \rangle \mid \langle DifferentIndividuals \rangle \mid \langle ClassAssertion \rangle \mid \langle ObjectPropertyAssertion \rangle \mid \langle NegativeObjectPropertyAssertion \rangle \mid \langle DataPropertyAssertion \rangle \mid \langle NegativeDataPropertyAssertion \rangle$

#### Individual Equality

$\langle SameIndividual \rangle ::= \langle Individual \rangle \backslash \text{ieq} \langle Individual \rangle \langle axiomAnnotations \rangle$

#### Individual Inequality

$\langle DifferentIndividuals \rangle ::= \langle Individual \rangle \backslash \text{idiff} \langle Individual \rangle \langle axiomAnnotations \rangle$

#### Class Assertions

$\langle ClassAssertion \rangle ::= \langle ClassExpression \rangle ( \langle Individual \rangle ) \langle axiomAnnotations \rangle$

#### Positive Object Property Assertions

$\langle ObjectPropertyAssertion \rangle ::= \langle ObjectPropertyExpression \rangle ( \langle sourceIndividual \rangle , \langle targetIndividual \rangle ) \langle axiomAnnotations \rangle$

#### Negative Object Property Assertions

$\langle NegativeObjectPropertyAssertion \rangle ::= \neg \langle ObjectPropertyExpression \rangle ( \langle sourceIndividual \rangle , \langle targetIndividual \rangle ) \langle axiomAnnotations \rangle$

#### Positive Data Property Assertions

$\langle DataPropertyAssertion \rangle ::= \langle DataPropertyExpression \rangle ( \langle sourceIndividual \rangle , \langle targetValue \rangle ) \langle axiomAnnotations \rangle$

## Negative Data Property Assertions

$\langle \text{NegativeDataPropertyAssertion} \rangle ::= \text{'!'} \langle \text{DataPropertyExpression} \rangle \text{'('} \langle \text{sourceIndividual} \rangle \text{' ,' } \langle \text{targetValue} \rangle \text{' )'} \langle \text{axiomAnnotations} \rangle$

### 10.6.2 Sequence Assertion

$\langle \text{SequenceAssertion} \rangle ::= \langle \text{SequenceSameIndividual} \rangle \mid \langle \text{SequenceDifferentIndividuals} \rangle$

## Sequence Individual Equality

$\langle \text{SequenceSameIndividual} \rangle ::= \text{'\ialleq'} \text{'{' } \langle \text{Individual} \rangle \text{' (' } \langle \text{Individual} \rangle \text{' )'} + \text{'}' \langle \text{axiomAnnotations} \rangle$

## Individual Inequality

$\langle \text{SequenceDifferentIndividuals} \rangle ::= \text{'\ialldiff'} \text{'{' } \langle \text{Individual} \rangle \text{' (' } \langle \text{Individual} \rangle \text{' )'} + \text{'}' \langle \text{axiomAnnotations} \rangle$

# 11 Sample Ontology

In this section a sample ontology is outlined.

```
% define base namespace for this ontology
\ns <http://basenamespace.owl#>

% define additional custom namespaces
% to refer to concept/property/object defined in a given
  namespace use a prefix notation ns:name, e.g., owl:Thing,
  owl:Nothing
\ns ns1: <http://www.namespace1.com/ns1#>
\ns ns2: <http://www.namespace2.com/ns2#>

\begin{ontology}
% import ontologies
\import <http://www.firstontology.org/first.owl>
\import <http://www.firstontology.org/second.owl>

% now you can specify the axioms of the ontology
Person \cisa owl:Thing
Person \a{rdfs:label, "Person"}
Person \a{rdfs:comment, "This is a class for representing
  people"}
Person \cisa \candof{\d[>=1]{hasName}, \d[=1]{hasSurname},
  \dexists{hasSex}{Sex}, \d[=1]{hasAge}}

% if some already existing ontologies are required to be
  reused without cutting-and-pasting them into the current
  ontology, the input command can be used
```

```

\input{/input.txt}

% hasName, hasSurname, and hasAge are datatype properties
hasName \ddomain Person
hasName \drange xsd:string
hasName \a{rdfs:comment, "Person's name"}
hasSurname \ddomain Person
hasSurname \drange xsd:string
hasSurname \a{rdfs:comment, "Person's surname"}
hasAge \ddomain Person
hasAge \drange xsd:integer

% let us define another datatype property by enumerating
  possible values
hasHairColor \ddomain Person
hasHairColor \drange \droneof{"Blonde", "Brown", "Red"}

% if we have another property hasLastName, we can define it
  as equal to the property hasSurname
hasLastName \deq hasSurname

% sex can be defined as a concept containing two objects,
  male and female
Sex \ceq \ooneof{M, F}

M \a{rdfs:label, "Male"}
F \a{rdfs:label, "Female"}

Father \cisa Person
Child \cisa (Person \cand \oexists{hasFather}{Father})

% if we want to reuse some concept defined in the other
  namespace
Father \cisa ns1:Father
ns1:isFatherOf \oinv hasFather

% property has father is functional
hasFather \ofunc

% let's populate the ontology with some individual data
Person(john)

% to express the fact that john and johny both refer to the
  same individual one can use the object equality construct
john \ieq johny

hasName(john, "John")
hasSurname(john, "Wild")
hasAge(john, "35"[xsd:integer])
hasSex(john, M)

```

```

Child(katty)
% to express the fact that john is different from katty one
  can use the object difference construct
katty \idiff john
hasName(katty, "Katty")
hasName(katty, "Wild")
hasSex(katty, F)
hasAge(katty, "3"[xsd:integer])

% John is Katty's father
hasFather(katty, john)

\end{ontology}

```

## References

- [1] M. Duerst and M. Suignard. RFC 3987: Internationalized Resource Identifiers (IRIs). RFC 3987 (Proposed Standard), see <http://www.ietf.org/rfc/rfc3987.txt>, January 2005.
- [2] A. Phillips and M. Davis. BCP 47 – Tags for Identifying Languages. BCP 47 Standard, see <http://www.rfc-editor.org/rfc/bcp/bcp47.txt>, September 2006.
- [3] Eric Prud'hommeaux and Andy Seaborne. Sparql query language for rdf. Latest version available as <http://www.w3.org/TR/rdf-sparql-query/>, January 2008.