An OWL Full Interpretation

Jeremy J. Carroll
HP Laboratories

HPL-2008-60
May 21, 2008*

OWL,
semantics,
Herbrand,
OWL Full

This report is an appendix to report HPL-2008-59. It gives a worked example of the construction used in the proof from that report. For finiteness, a reduced datatype map consisting of only xsd:boolean is used. Each of the graphs in the construction is listed explicitly, with some redundancy eliminated. The final Herbrand graph contains about 15,000 triples.

An OWL Full Interpretation

Jeremy J.Carroll
HP Labs, Bristol, UK

Abstract

This report is an appendix to report HPL-2008-59. It gives a worked example of the construction used in the proof from that report. For finiteness, a reduced datatype map consisting of only xsd:boolean is used. Each of the graphs in the construction is listed explicitly, with some redundancy eliminated. The final Herbrand graph contains about 15,000 triples.

1 Introduction

This is an appendix to [1, 2]. The construction in those reports builds a Herbrand graph in which the property extension of every property is totally explicit. This appendix presents a complete worked example.

The construction is essentially an infinite one. We make it finite in the following ways:

- We have a very small datatype map D, being { `xsd:boolean` }
- We pretend $L_{\text{plain}} = \{\texttt{""}\}$
- We use the ter Horst technique of ignoring every `rdf:_`*i* except the first; these only appear because of their presence in the RDF and RDFS axioms.

The first of these two are non-conformant. Datatype maps are required to include `xsd:string`, `xsd:integer` and `XMLLiteral`. L_{plain} is infinite. The differences are of no great consequence here.

Each section presents one of the graphs in the construction of section 9 of the two papers. Each graph is given by listing the new triples in that graph that were not in the previous triples.

1.1 Notation

In addition to the abbreviations specified in [2], we have the following: `disjoint`*With*, `intersect`*ectionOf*, `equiv`*alent*`C`*lass*, `minCard`*inality*, `maxCard`*inality*, `card`*inality*, `different`*From*, `AllDiff`*erent*, `d`*i*`st`*i*`nctMem`*ber*`s`, `Fun`*ctional*`Prop`*erty*, `subClass`*of*, `allVal`*ues*`From`, `someVal`*ues*`From`, `C`*ontainer*`Mem`*bership*`Prop`*erty*, `vers`*ion*`Info`. and ^^*xsd:*`b`*oolean*.

Some of the triples in the first few sections (5 to 8), are shown with a wavy line under, such as $\utilde{\texttt{("0"\^{}\^{}b, type, eg:c)}}$. This means that the triple is not included in the graphs H_i, i.e. sections 9 to 18.

To avoid too much redundancy we also use the following sets of nodes, identified by their first element, with an overline.

Relating to classes	
$\overline{b_{10}}$	b_{10}, b_9,
$\overline{\texttt{Nothing}}$	Nothing, b_2, DataRange, DataProp, DeprClass, DeprProp, Ontology, Restrict, Alt, Bag, Seq, Statement, XMLLiteral, Container,
$\overline{b_3}$	b_3, b_6, b_7,
$\overline{\texttt{owl:Class}}$	owl:Class, rdfs:Class,
$\overline{\texttt{ObjProp}}$	ObjProp, Prop,
$\overline{\texttt{Thing}}$	Thing, Resource,
Relating to properties	
$\overline{b_{11}}$	b_{11}, b_9,
$\overline{\texttt{priorVers}}$	priorVers, allVals, backComp, card, hasValue, imports, incompat, maxCard, minCard, onProperty, someVals, versInfo, _1, object, predicate, subject, value, comment, isDefinedBy, label, member, seeAlso,

These are used in *metatriples* like ($\overline{\texttt{owl:Class}}$, equivC, $\overline{\texttt{owl:Class}}$) in the listing. This one stands for four triples in the graph, by taking each member of the set for both subject and object. Some of these may have occurred earlier in the construction.

2 The initial graph

This is given in section 5. It has several features intended to illustrate the action of the construction on literals, which while somewhat artificial in this example, are part of the D-semantics [3], and covered by our method.

- We use non-canonical lexical forms. The notion of 'canonical' used in $\mathcal{VL}_D$ is not, in general, the same as that in XML Schema [4], which provides several different 'canonical' forms for the same value depending on which derived datatype is used. So in general, the literal replacement step between G_3 and H_0 is necessary. In our example, it is artificial.

- We use URI and blank nodes which implicitly represent literals; and then use some of these in predicate position.

3 The Function ψ

The function ψ, see section 9.2 of [2], is determined by some D-interpretation of G_3. We take ψ to be:

$$\psi(x) = \begin{cases} \texttt{"true"\^{}\^{}b} & x = \texttt{"1"\^{}\^{}b} \\ \texttt{"false"\^{}\^{}b} & x = \texttt{"0"\^{}\^{}b} \\ \texttt{"true"\^{}\^{}b} & x = \texttt{eg:v} \\ \texttt{"false"\^{}\^{}b} & x = b_{15} \\ x & \text{otherwise} \end{cases} \tag{1}$$

Others would be found from different D-interpretations.

4 The Interpretation

The interpretation in section 10 of [2], uses the graph presented in this document, along with the empty sting, the boolean datatype, and its values as the domain of discourse. The function χ is then given as:

$$\chi(x) = \begin{cases} \text{the boolean datatype} & x = \texttt{boolean} \\ \texttt{""} & x = \texttt{""} \\ \text{TRUE} & x = \texttt{"true"\^{}\^{}b} \\ \text{FALSE} & x = \texttt{"false"\^{}\^{}b} \\ x & \text{otherwise} \end{cases} \tag{2}$$

With the last line meaning the occurrences of the appropriate text string in triples (and implicitly in metatriples) in sections 5 to 18, except where deleted by the wavy line. The actual interpretation is built as specified in [2], via the function θ, which we modify, following ter Horst [5] to deal with the $\texttt{rdf:}\_i$:

$$\theta(x) = \begin{cases} \chi(\_1) & x \in \{\texttt{rdf:}\_i : i = 2, 3 \ldots\} \\ \chi(x) & x \in \mathrm{nd}(H_9) \\ \chi(\psi(x)) & x \in \mathrm{nd}(G_3) \end{cases} \tag{3}$$

5 G_0

("", type, eg:c)
~~("0"^^b, type, eg:c)~~
~~(eg:a, eg:p, "1"^^b)~~
~~(eg:a, eg:p, eg:v)~~
~~(eg:b, eg:p, b_{15})~~
~~(eg:v, eg:v, eg:v)~~
(eg:p, range, boolean)

6 G_1

(boolean, type, Datatype)
("", type, Literal)
("false"^^b, type, boolean)
("true"^^b, type, boolean)

7 G_2

(nil, type, List)
(_1, type, Prop)
(first, type, Prop)
(object, type, Prop)
(predicate, type, Prop)
(rest, type, Prop)
(subject, type, Prop)
(type, type, Prop)
(value, type, Prop)
(_1, type, CMemProp)
(XMLLiteral, type, Datatype)
(_1, domain, Resource)
(first, domain, List)
(object, domain, Statement)
(predicate, domain, Statement)
(rest, domain, List)
(subject, domain, Statement)
(type, domain, Resource)
(value, domain, Resource)
(comment, domain, Resource)
(domain, domain, Prop)
(isDefinedBy, domain, Resource)
(label, domain, Resource)
(member, domain, Resource)
(range, domain, Prop)
(seeAlso, domain, Resource)
(subClass, domain, rdfs:Class)
(subPropOf, domain, Prop)
(_1, range, Resource)
(first, range, Resource)

(object, range, Resource)
(predicate, range, Resource)
(rest, range, List)
(subject, range, Resource)
(type, range, rdfs:Class)
(value, range, Resource)
(comment, range, Literal)
(domain, range, rdfs:Class)
(isDefinedBy, range, Resource)
(label, range, Literal)
(member, range, Resource)
(range, range, rdfs:Class)
(seeAlso, range, Resource)
(subClass, range, rdfs:Class)
(subPropOf, range, Prop)
(Alt, subClass, Container)
(Bag, subClass, Container)
(Seq, subClass, Container)
(XMLLiteral, subClass, Literal)
(CMemProp, subClass, Prop)
(Datatype, subClass, rdfs:Class)
(isDefinedBy, subPropOf, seeAlso)

8 G_3

(eg:p, type, Prop)
(eg:v, type, Prop)
(comment, type, Prop)
(domain, type, Prop)
(isDefinedBy, type, Prop)
(label, type, Prop)
(member, type, Prop)
(range, type, Prop)
(seeAlso, type, Prop)
(subClass, type, Prop)

(subPropOf, type, Prop)
(eg:c, type, rdfs:Class)
(Alt, type, rdfs:Class)
(Bag, type, rdfs:Class)
(List, type, rdfs:Class)
(Prop, type, rdfs:Class)
(Seq, type, rdfs:Class)
(Statement, type, rdfs:Class)
(XMLLiteral, type, rdfs:Class)
(rdfs:Class, type, rdfs:Class)

(Container, type, rdfs:Class)
(CMemProp, type, rdfs:Class)
(Datatype, type, rdfs:Class)
(Literal, type, rdfs:Class)
(Resource, type, rdfs:Class)
(boolean, type, rdfs:Class)
("1"^^b, type, Literal)
("false"^^b, type, Literal)
("true"^^b, type, Literal)
(b_{15}, type, Literal)

(eg:v, type, Literal)
("", type, Resource)
("0"^^b, type, Resource)
("1"^^b, type, Resource)
("false"^^b, type, Resource)
("true"^^b, type, Resource)
(b_{15}, type, Resource)
(eg:a, type, Resource)
(eg:b, type, Resource)
(eg:c, type, Resource)

(eg:p, type, Resource)
(eg:v, type, Resource)
(Alt, type, Resource)
(Bag, type, Resource)
(List, type, Resource)
(Prop, type, Resource)
(Seq, type, Resource)
(Statement, type, Resource)
(XMLLiteral, type, Resource)
(_1, type, Resource)

(first, type, Resource)
(nil, type, Resource)
(object, type, Resource)
(predicate, type, Resource)
(rest, type, Resource)
(subject, type, Resource)
(type, type, Resource)
(value, type, Resource)
(rdfs:Class, type, Resource)
(Container, type, Resource)

(CMemProp, type, Resource)
(Datatype, type, Resource)
(Literal, type, Resource)
(Resource, type, Resource)
(comment, type, Resource)
(domain, type, Resource)
(isDefinedBy, type, Resource)
(label, type, Resource)
(member, type, Resource)
(range, type, Resource)

(seeAlso, type, Resource)
(subClass, type, Resource)
(subPropOf, type, Resource)
(boolean, type, Resource)
("1"^^b, type, boolean)
(b_{15}, type, boolean)
(eg:v, type, boolean)
(eg:c, subClass, eg:c)
(Alt, subClass, Alt)
(Bag, subClass, Bag)

(List, subClass, List)
(Prop, subClass, Prop)
(Seq, subClass, Seq)
(Statement, subClass, Statement)
(XMLLiteral, subClass, XMLLiteral)
(rdfs:Class, subClass, rdfs:Class)
(Container, subClass, Container)
(CMemProp, subClass, CMemProp)
(Datatype, subClass, Datatype)
(Literal, subClass, Literal)

(Resource, subClass, Resource)
(boolean, subClass, Literal)
(boolean, subClass, boolean)
(eg:p, subPropOf, eg:p)
(eg:v, subPropOf, eg:v)
(_1, subPropOf, _1)
(_1, subPropOf, member)
(first, subPropOf, first)

(object, subPropOf, object)
(predicate, subPropOf, predicate)
(rest, subPropOf, rest)
(subject, subPropOf, subject)
(type, subPropOf, type)
(value, subPropOf, value)
(comment, subPropOf, comment)
(domain, subPropOf, domain)

(isDefinedBy, subPropOf, isDefinedBy)
(label, subPropOf, label)
(member, subPropOf, member)
(range, subPropOf, range)
(seeAlso, subPropOf, seeAlso)
(subClass, subPropOf, subClass)
(subPropOf, subPropOf, subPropOf)

9 H_0

("false"^^b, type, eg:c)
("true"^^b, type, Prop)

("true"^^b, "true"^^b, "true"^^b)
(eg:a, eg:p, "true"^^b)

(eg:b, eg:p, "false"^^b)
("true"^^b, subPropOf, "true"^^b)

10 H_1

(rdfs:Class, type, $\overline{b_3}$)
(b_2, type, b_8)
(b_2, type, b_9)
(b_3, type, b_9)
(b_1, type, AllDiff)
(versInfo, type, AnnProp)
(comment, type, AnnProp)
(isDefinedBy, type, AnnProp)
(label, type, AnnProp)
(seeAlso, type, AnnProp)

(priorVers, type, FunProp)
(priorVers, type, InvFunProp)
(backComp, type, OntProp)
(imports, type, OntProp)
(incompat, type, OntProp)
(priorVers, type, OntProp)
(equivProp, type, SymProp)
(inverseOf, type, SymProp)
(subClass, type, TransProp)
(subPropOf, type, TransProp)

(b_9, type, priorVers)
(allVals, type, Prop)
(card, type, Prop)
(complmntOf, type, Prop)
(different, type, Prop)
(disjoint, type, Prop)
(dstnctMems, type, Prop)
(hasValue, type, Prop)
(intersect, type, Prop)
(inverseOf, type, Prop)

(maxCard, type, Prop)
(minCard, type, Prop)
(onProperty, type, Prop)
(oneOf, type, Prop)
(someVals, type, Prop)
(unionOf, type, Prop)
(b_2, type, rdfs:Class)
(owl:Class, type, rdfs:Class)
(DataRange, type, rdfs:Class)
(DeprClass, type, rdfs:Class)

(DeprProp, type, rdfs:Class)
(Nothing, type, rdfs:Class)
(Ontology, type, rdfs:Class)
(Literal, type, Datatype)
(b_2, b_8, b_4)
(b_3, b_8, b_2)
(b_8, b_8, b_2)
(b_8, b_8, b_3)
(b_2, b_9, b_3)
(b_4, b_{10}, b_5)

(b_2, b_{11}, b_3)
(b_3, b_{12}, b_2)
(Nothing, complmntOf, $\overline{\text{Thing}}$)
($\overline{\text{Thing}}$, complmntOf, Nothing)
(Thing, different, Nothing)
(Nothing, disjoint, Nothing)
(Nothing, disjoint, Resource)
(Resource, disjoint, Nothing)
(b_1, dstnctMems, b_2)
(b_1, dstnctMems, b_3)

($\overline{\text{Thing}}$, equivC, $\overline{\text{Thing}}$)
(b_8, equivProp, b_8)
(incompat, equivProp, backComp)
(priorVers, equivProp, backComp)
(priorVers, equivProp, priorVers)
(b_3, intersect, b_5)
($\overline{\text{owl:Class}}$, intersect, b_2)
(rdfs:Class, intersect, b_3)
(b_9, inverseOf, b_{12})
(b_{12}, inverseOf, b_9)

(equivProp, inverseOf, equivProp)
(imports, inverseOf, priorVers)
(incompat, inverseOf, backComp)
(inverseOf, inverseOf, inverseOf)
(priorVers, inverseOf, backComp)
(priorVers, inverseOf, priorVers)
(b_2, oneOf, nil)
(b_3, oneOf, b_2)
(b_6, oneOf, b_2)
(b_6, oneOf, b_3)

(b_7, oneOf, b_3)
(Thing, sameAs, Thing)
(b_2, unionOf, nil)
($\overline{\text{owl:Class}}$, unionOf, b_2)
(rdfs:Class, unionOf, b_3)
(b_2, first, rdfs:Class)
(b_3, first, rdfs:Class)
(b_4, first, Nothing)
(b_5, first, b_3)
(b_2, rest, nil)

(b_3, rest, nil)
(b_4, rest, nil)
(b_5, rest, nil)
(b_9, domain, b_{10})
(b_{10}, domain, b_8)
(b_{12}, domain, b_{10})
(b_9, range, b_{10})

(b_{10}, range, b_8)
(AnnProp, subClass, Prop)
(DataProp, subClass, Prop)
(OntProp, subClass, Prop)
(Restrict, subClass, rdfs:Class)
(Prop, subClass, ObjProp)
(Prop, subClass, Thing)

(rdfs:Class, subClass, owl:Class)
(Resource, subClass, Thing)
(b_8, subPropOf, b_8)
(inverseOf, subPropOf, inverseOf)
(priorVers, subPropOf, b_9)
(priorVers, subPropOf, backComp)

11 H_2

(b_4, type, b_8)
(b_5, type, b_8)
(b_2, type, b_{10})
(b_3, type, b_{10})
($\overline{b_3}$, type, $\overline{\text{owl:Class}}$)
(b_8, type, $\overline{\text{owl:Class}}$)
($\overline{b_{10}}$, type, $\overline{\text{owl:Class}}$)
(eg:c, type, owl:Class)
(AllDiff, type, $\overline{\text{owl:Class}}$)
(AnnProp, type, $\overline{\text{owl:Class}}$)

($\overline{\text{owl:Class}}$, type, owl:Class)
(FunProp, type, $\overline{\text{owl:Class}}$)
(InvFunProp, type, $\overline{\text{owl:Class}}$)
($\overline{\text{Nothing}}$, type, $\overline{\text{owl:Class}}$)
($\overline{\text{ObjProp}}$, type, $\overline{\text{owl:Class}}$)
(OntProp, type, $\overline{\text{owl:Class}}$)
(SymProp, type, $\overline{\text{owl:Class}}$)
($\overline{\text{Thing}}$, type, $\overline{\text{owl:Class}}$)
(TransProp, type, $\overline{\text{owl:Class}}$)
(priorVers, type, $\overline{\text{owl:Class}}$)

(List, type, owl:Class)
(CMemProp, type, owl:Class)
(Datatype, type, owl:Class)
(Literal, type, owl:Class)
(boolean, type, owl:Class)
("true"^^b, type, ObjProp)
(b_8, type, $\overline{\text{ObjProp}}$)
(b_{10}, type, $\overline{\text{ObjProp}}$)
($\overline{b_{11}}$, type, $\overline{\text{ObjProp}}$)
(b_{12}, type, $\overline{\text{ObjProp}}$)

(eg:p, type, ObjProp)
(complmntOf, type, ObjProp)
(different, type, ObjProp)
(disjoint, type, ObjProp)
(dstnctMems, type, ObjProp)
(equivC, type, $\overline{\texttt{ObjProp}}$)
(equivProp, type, $\overline{\texttt{ObjProp}}$)
(intersect, type, ObjProp)
(inverseOf, type, ObjProp)
(oneOf, type, ObjProp)

($\overline{\texttt{priorVers}}$, type, $\overline{\texttt{ObjProp}}$)
(sameAs, type, $\overline{\texttt{ObjProp}}$)
(unionOf, type, ObjProp)
(first, type, ObjProp)
(rest, type, ObjProp)
(type, type, ObjProp)
(domain, type, ObjProp)
(range, type, ObjProp)
(subClass, type, ObjProp)
(subPropOf, type, ObjProp)

("", type, Thing)
("false"^^b, type, Thing)
("true"^^b, type, Thing)
(b_1, type, $\overline{\texttt{Thing}}$)
($\overline{b_3}$, type, $\overline{\texttt{Thing}}$)
(b_4, type, $\overline{\texttt{Thing}}$)
(b_5, type, $\overline{\texttt{Thing}}$)
(b_8, type, $\overline{\texttt{Thing}}$)
(b_{10}, type, $\overline{\texttt{Thing}}$)
($\overline{b_{11}}$, type, $\overline{\texttt{Thing}}$)

(b_{12}, type, $\overline{\texttt{Thing}}$)
(eg:a, type, Thing)
(eg:b, type, Thing)
(eg:c, type, Thing)
(eg:p, type, Thing)
(AllDiff, type, $\overline{\texttt{Thing}}$)
(AnnProp, type, $\overline{\texttt{Thing}}$)
($\overline{\texttt{owl:Class}}$, type, $\overline{\texttt{Thing}}$)
(FunProp, type, $\overline{\texttt{Thing}}$)
(InvFunProp, type, $\overline{\texttt{Thing}}$)

($\overline{\texttt{Nothing}}$, type, $\overline{\texttt{Thing}}$)
($\overline{\texttt{ObjProp}}$, type, $\overline{\texttt{Thing}}$)
(OntProp, type, $\overline{\texttt{Thing}}$)
(SymProp, type, $\overline{\texttt{Thing}}$)
($\overline{\texttt{Thing}}$, type, $\overline{\texttt{Thing}}$)
(TransProp, type, $\overline{\texttt{Thing}}$)
(complmntOf, type, $\overline{\texttt{Thing}}$)
(different, type, $\overline{\texttt{Thing}}$)
(disjoint, type, $\overline{\texttt{Thing}}$)
(dstnctMems, type, $\overline{\texttt{Thing}}$)

(equivC, type, $\overline{\texttt{Thing}}$)
(equivProp, type, $\overline{\texttt{Thing}}$)
(intersect, type, $\overline{\texttt{Thing}}$)
(inverseOf, type, $\overline{\texttt{Thing}}$)
(oneOf, type, $\overline{\texttt{Thing}}$)
($\overline{\texttt{priorVers}}$, type, $\overline{\texttt{Thing}}$)
(sameAs, type, $\overline{\texttt{Thing}}$)
(unionOf, type, $\overline{\texttt{Thing}}$)
(List, type, Thing)
(first, type, Thing)

(nil, type, Thing)
(rest, type, Thing)
(type, type, Thing)
(CMemProp, type, Thing)
(Datatype, type, Thing)
(Literal, type, Thing)
(domain, type, Thing)
(range, type, Thing)
(subClass, type, Thing)
(subPropOf, type, Thing)

(boolean, type, Thing)
(b_2, type, List)
(b_3, type, List)
(b_4, type, List)
(b_5, type, List)
("true"^^b, domain, Resource)
(b_8, domain, Resource)
(b_{10}, domain, Resource)
($\overline{b_{11}}$, domain, Resource)
(b_{12}, domain, Resource)

(eg:p, domain, Resource)
(complmntOf, domain, Resource)
(different, domain, Resource)
(disjoint, domain, Resource)
(dstnctMems, domain, Resource)
(equivC, domain, Resource)
(equivProp, domain, Resource)
(intersect, domain, Resource)
(inverseOf, domain, Resource)
(oneOf, domain, Resource)

(priorVers, domain, $\overline{b_3}$)
(priorVers, domain, b_8)
(priorVers, domain, $\overline{b_{10}}$)
(priorVers, domain, eg:c)
(priorVers, domain, AllDiff)
(priorVers, domain, AnnProp)
(priorVers, domain, $\overline{\texttt{owl:Class}}$)
(priorVers, domain, FunProp)
(priorVers, domain, InvFunProp)
(priorVers, domain, $\overline{\texttt{Nothing}}$)

(priorVers, domain, $\overline{\texttt{ObjProp}}$)
(priorVers, domain, OntProp)
(priorVers, domain, SymProp)
(priorVers, domain, Thing)
(priorVers, domain, TransProp)
(priorVers, domain, priorVers)
(priorVers, domain, List)
(priorVers, domain, CMemProp)
(priorVers, domain, Datatype)
(priorVers, domain, Literal)

($\overline{\texttt{priorVers}}$, domain, Resource)
(priorVers, domain, boolean)
(sameAs, domain, Resource)
(unionOf, domain, Resource)
(first, domain, Resource)
(rest, domain, Resource)
(domain, domain, Resource)
(range, domain, Resource)
(subClass, domain, Resource)
(subPropOf, domain, Resource)

("true"^^b, range, Resource)
(b_8, range, Resource)
(b_{10}, range, Resource)
($\overline{b_{11}}$, range, Resource)
(b_{12}, range, Resource)
(eg:p, range, Resource)
(complmntOf, range, Resource)
(different, range, Resource)
(disjoint, range, Resource)
(dstnctMems, range, Resource)

(equivC, range, Resource)
(equivProp, range, Resource)
(intersect, range, Resource)
(inverseOf, range, Resource)
(oneOf, range, Resource)
(priorVers, range, $\overline{b_3}$)
(priorVers, range, b_8)
(priorVers, range, $\overline{b_{10}}$)
(priorVers, range, eg:c)
(priorVers, range, AllDiff)

(priorVers, range, AnnProp)
(priorVers, range, $\overline{\texttt{owl:Class}}$)
(priorVers, range, FunProp)
(priorVers, range, InvFunProp)
(priorVers, range, $\overline{\texttt{Nothing}}$)
(priorVers, range, $\overline{\texttt{ObjProp}}$)
(priorVers, range, OntProp)
(priorVers, range, SymProp)
(priorVers, range, Thing)
(priorVers, range, TransProp)

(priorVers, range, priorVers)
(priorVers, range, List)
(priorVers, range, CMemProp)
(priorVers, range, Datatype)
(priorVers, range, Literal)
($\overline{\texttt{priorVers}}$, range, Resource)
(priorVers, range, boolean)
(sameAs, range, Resource)
(unionOf, range, Resource)
(rest, range, Resource)
(type, range, Resource)
(domain, range, Resource)
(range, range, Resource)
(subClass, range, Resource)
(subPropOf, range, Resource)
(b_2, subClass, b_2)
(b_3, subClass, b_3)
(b_6, subClass, b_6)
(b_7, subClass, b_7)
(b_8, subClass, b_8)
(b_9, subClass, b_9)
(b_{10}, subClass, b_{10})
(AllDiff, subClass, AllDiff)
(AnnProp, subClass, AnnProp)
(owl:Class, subClass, owl:Class)
(DataRange, subClass, DataRange)
(DataProp, subClass, DataProp)
(DeprClass, subClass, DeprClass)
(DeprProp, subClass, DeprProp)
(FunProp, subClass, FunProp)

(InvFunProp, subClass, InvFunProp)
(Nothing, subClass, Nothing)
(ObjProp, subClass, ObjProp)
(Ontology, subClass, Ontology)
(OntProp, subClass, OntProp)
(Restrict, subClass, Restrict)
(SymProp, subClass, SymProp)
(Thing, subClass, Thing)
(TransProp, subClass, TransProp)
(priorVers, subClass, priorVers)
(b_9, subPropOf, b_9)
(b_{10}, subPropOf, b_{10})
(b_{11}, subPropOf, b_{11})
(b_{12}, subPropOf, b_{12})
(allVals, subPropOf, allVals)
(backComp, subPropOf, backComp)
(card, subPropOf, card)
(complmntOf, subPropOf, complmntOf)
(different, subPropOf, different)
(disjoint, subPropOf, disjoint)
(dstnctMems, subPropOf, dstnctMems)
(equivC, subPropOf, equivC)
(equivProp, subPropOf, equivProp)
(hasValue, subPropOf, hasValue)
(imports, subPropOf, imports)
(incompat, subPropOf, incompat)
(intersect, subPropOf, intersect)
(maxCard, subPropOf, maxCard)
(minCard, subPropOf, minCard)
(onProperty, subPropOf, onProperty)

(oneOf, subPropOf, oneOf)
(priorVers, subPropOf, priorVers)
(sameAs, subPropOf, sameAs)
(someVals, subPropOf, someVals)
(unionOf, subPropOf, unionOf)
(versInfo, subPropOf, versInfo)

12 H_3

The nodes of the graph H_2 are: { "", "false"^^b, "true"^^b, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, eg:a, eg:b, eg:c, eg:p, AllDiff, AnnProp, owl:Class, DataRange, DataProp, DeprClass, DeprProp, FunProp, InvFunProp, Nothing, ObjProp, Ontology, OntProp, Restrict, SymProp, Thing, TransProp, allVals, backComp, card, complmntOf, different, disjoint, dstnctMems, equivC, equivProp, hasValue, imports, incompat, intersect, inverseOf, maxCard, minCard, onProperty, oneOf, priorVers, sameAs, someVals, unionOf, versInfo, Alt, Bag, List, Prop, Seq, Statement, XMLLiteral, _1, first, nil, object, predicate, rest, subject, type, value, rdfs:Class, Container, CMemProp, Datatype, Literal, Resource, comment, domain, isDefinedBy, label, member, range, seeAlso, subClass, subPropOf, boolean, }.
For any single node n in this set, we add (n, sameAs, n).
For any pair of distinct nodes n, n' in this set, we add (n, different, n').

13 H_4

("true"^^b, type, FunProp)
(b_{10}, type, FunProp)
($\overline{b_{11}}$, type, FunProp)
(b_{12}, type, FunProp)
(eg:p, type, FunProp)
($\overline{\texttt{priorVers}}$, type, FunProp)
(sameAs, type, FunProp)
(first, type, FunProp)
(rest, type, FunProp)
("true"^^b, type, InvFunProp)
(b_{10}, type, InvFunProp)
($\overline{b_{11}}$, type, InvFunProp)
(b_{12}, type, InvFunProp)
(eg:p, type, InvFunProp)
(dstnctMems, type, InvFunProp)
($\overline{\texttt{priorVers}}$, type, InvFunProp)
(sameAs, type, InvFunProp)
("true"^^b, type, SymProp)
(complmntOf, type, SymProp)
(different, type, SymProp)
(disjoint, type, SymProp)
(equivC, type, SymProp)
($\overline{\texttt{priorVers}}$, type, SymProp)
(sameAs, type, SymProp)
("true"^^b, type, TransProp)
(b_{10}, type, TransProp)
($\overline{b_{11}}$, type, TransProp)
(b_{12}, type, TransProp)
(eg:p, type, TransProp)
(dstnctMems, type, TransProp)

(equivC, type, TransProp)
(equivProp, type, TransProp)
($\overline{\texttt{priorVers}}$, type, TransProp)
(sameAs, type, TransProp)
(rest, type, TransProp)
(eg:c, type, Datatype)
(eg:c, subClass, Literal)

14 H_5

$(\overline{\mathtt{Nothing}}, \mathtt{complmntOf}, \overline{\mathtt{Thing}})$
$(\overline{\mathtt{Thing}}, \mathtt{complmntOf}, \overline{\mathtt{Nothing}})$
$(\overline{b_3}, \mathtt{disjoint}, b_8)$
$(\overline{b_3}, \mathtt{disjoint}, \overline{b_{10}})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{FunProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{InvFunProp})$
$(\overline{b_3}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\overline{b_3}, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{SymProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{TransProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{List})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{Literal})$
$(\overline{b_3}, \mathtt{disjoint}, \mathtt{boolean})$
$(b_8, \mathtt{disjoint}, \overline{b_3})$
$(b_8, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(b_8, \mathtt{disjoint}, \mathtt{AllDiff})$
$(b_8, \mathtt{disjoint}, \mathtt{AnnProp})$
$(b_8, \mathtt{disjoint}, \mathtt{FunProp})$
$(b_8, \mathtt{disjoint}, \mathtt{InvFunProp})$
$(b_8, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(b_8, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(b_8, \mathtt{disjoint}, \mathtt{OntProp})$
$(b_8, \mathtt{disjoint}, \mathtt{SymProp})$

$(b_8, \mathtt{disjoint}, \mathtt{TransProp})$
$(b_8, \mathtt{disjoint}, \mathtt{priorVers})$
$(b_8, \mathtt{disjoint}, \mathtt{CMemProp})$
$(b_8, \mathtt{disjoint}, \mathtt{Datatype})$
$(b_8, \mathtt{disjoint}, \mathtt{Literal})$
$(b_8, \mathtt{disjoint}, \mathtt{boolean})$
$(\overline{b_{10}}, \mathtt{disjoint}, \overline{b_3})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{FunProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{InvFunProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\overline{b_{10}}, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{SymProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{TransProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{Literal})$
$(\overline{b_{10}}, \mathtt{disjoint}, \mathtt{boolean})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \overline{b_3})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, b_8)$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \overline{b_{10}})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{FunProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{InvFunProp})$

$(\mathtt{eg{:}c}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{SymProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{TransProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{List})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\mathtt{eg{:}c}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \overline{b_3})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, b_8)$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \overline{b_{10}})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{FunProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{InvFunProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{SymProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{TransProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{List})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{Literal})$
$(\mathtt{AllDiff}, \mathtt{disjoint}, \mathtt{boolean})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \overline{b_3})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, b_8)$

$(\mathtt{AnnProp}, \mathtt{disjoint}, \overline{b_{10}})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{List})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{Literal})$
$(\mathtt{AnnProp}, \mathtt{disjoint}, \mathtt{boolean})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{Literal})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{disjoint}, \mathtt{boolean})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \overline{b_3})$
$(\mathtt{FunProp}, \mathtt{disjoint}, b_8)$
$(\mathtt{FunProp}, \mathtt{disjoint}, \overline{b_{10}})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \mathtt{List})$
$(\mathtt{FunProp}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \overline{b_3})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, b_8)$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \overline{b_{10}})$

$(\mathtt{InvFunProp}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \mathtt{List})$
$(\mathtt{InvFunProp}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{b_3})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, b_8)$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{b_{10}})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{eg{:}c})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{AllDiff})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{AnnProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{\mathtt{owl{:}Class}})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{FunProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{InvFunProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{\mathtt{Nothing}})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{\mathtt{ObjProp}})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{OntProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{SymProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \overline{\mathtt{Thing}})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{TransProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{priorVers})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{List})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{CMemProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{Datatype})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{Literal})$
$(\overline{\mathtt{Nothing}}, \mathtt{disjoint}, \mathtt{boolean})$
$(\overline{\mathtt{ObjProp}}, \mathtt{disjoint}, \overline{b_3})$
$(\overline{\mathtt{ObjProp}}, \mathtt{disjoint}, b_8)$
$(\overline{\mathtt{ObjProp}}, \mathtt{disjoint}, \overline{b_{10}})$
$(\overline{\mathtt{ObjProp}}, \mathtt{disjoint}, \mathtt{eg{:}c})$

$(\overline{\texttt{ObjProp}}, \texttt{disjoint}, \texttt{AllDiff})$
$(\overline{\texttt{ObjProp}}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\overline{\texttt{ObjProp}}, \texttt{disjoint}, \texttt{List})$
$(\overline{\texttt{ObjProp}}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{OntProp}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{OntProp}, \texttt{disjoint}, b_8)$
$(\texttt{OntProp}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{OntProp}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{List})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{CMemProp})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{Literal})$
$(\texttt{OntProp}, \texttt{disjoint}, \texttt{boolean})$
$(\texttt{SymProp}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{SymProp}, \texttt{disjoint}, b_8)$
$(\texttt{SymProp}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{SymProp}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{SymProp}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{SymProp}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{SymProp}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{SymProp}, \texttt{disjoint}, \texttt{List})$
$(\texttt{SymProp}, \texttt{disjoint}, \texttt{Datatype})$
$(\overline{\texttt{Thing}}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{TransProp}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{TransProp}, \texttt{disjoint}, b_8)$
$(\texttt{TransProp}, \texttt{disjoint}, \overline{b_{10}})$

$(\texttt{TransProp}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{TransProp}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{TransProp}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{TransProp}, \texttt{disjoint}, \texttt{List})$
$(\texttt{TransProp}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{priorVers}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{priorVers}, \texttt{disjoint}, b_8)$
$(\texttt{priorVers}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{priorVers}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{OntProp})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{SymProp})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{List})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{CMemProp})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{Literal})$
$(\texttt{priorVers}, \texttt{disjoint}, \texttt{boolean})$
$(\texttt{List}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{List}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{List}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{List}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{List}, \texttt{disjoint}, \texttt{FunProp})$
$(\texttt{List}, \texttt{disjoint}, \texttt{InvFunProp})$
$(\texttt{List}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{List}, \texttt{disjoint}, \overline{\texttt{ObjProp}})$
$(\texttt{List}, \texttt{disjoint}, \texttt{OntProp})$
$(\texttt{List}, \texttt{disjoint}, \texttt{SymProp})$
$(\texttt{List}, \texttt{disjoint}, \texttt{TransProp})$

$(\texttt{List}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{List}, \texttt{disjoint}, \texttt{CMemProp})$
$(\texttt{List}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{List}, \texttt{disjoint}, \texttt{Literal})$
$(\texttt{List}, \texttt{disjoint}, \texttt{boolean})$
$(\texttt{CMemProp}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{CMemProp}, \texttt{disjoint}, b_8)$
$(\texttt{CMemProp}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{CMemProp}, \texttt{disjoint}, \overline{\texttt{owl:Class}})$
$(\texttt{CMemProp}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{OntProp})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{List})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{Literal})$
$(\texttt{CMemProp}, \texttt{disjoint}, \texttt{boolean})$
$(\texttt{Datatype}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{Datatype}, \texttt{disjoint}, b_8)$
$(\texttt{Datatype}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{eg:c})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{FunProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{InvFunProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{Datatype}, \texttt{disjoint}, \overline{\texttt{ObjProp}})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{OntProp})$

$(\texttt{Datatype}, \texttt{disjoint}, \texttt{SymProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{TransProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{List})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{CMemProp})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{Literal})$
$(\texttt{Datatype}, \texttt{disjoint}, \texttt{boolean})$
$(\texttt{Literal}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{Literal}, \texttt{disjoint}, b_8)$
$(\texttt{Literal}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{Literal}, \texttt{disjoint}, \overline{\texttt{owl:Class}})$
$(\texttt{Literal}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{OntProp})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{List})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{CMemProp})$
$(\texttt{Literal}, \texttt{disjoint}, \texttt{Datatype})$
$(\texttt{boolean}, \texttt{disjoint}, \overline{b_3})$
$(\texttt{boolean}, \texttt{disjoint}, b_8)$
$(\texttt{boolean}, \texttt{disjoint}, \overline{b_{10}})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{AllDiff})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{AnnProp})$
$(\texttt{boolean}, \texttt{disjoint}, \overline{\texttt{owl:Class}})$
$(\texttt{boolean}, \texttt{disjoint}, \overline{\texttt{Nothing}})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{OntProp})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{priorVers})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{List})$
$(\texttt{boolean}, \texttt{disjoint}, \texttt{CMemProp})$

$(\texttt{boolean}, \texttt{disjoint}, \texttt{Datatype})$
$(\overline{b_3}, \texttt{equivC}, \overline{b_3})$
$(b_8, \texttt{equivC}, b_8)$
$(\overline{b_{10}}, \texttt{equivC}, \overline{b_{10}})$
$(\texttt{eg:c}, \texttt{equivC}, \texttt{eg:c})$
$(\texttt{AllDiff}, \texttt{equivC}, \texttt{AllDiff})$
$(\texttt{AnnProp}, \texttt{equivC}, \texttt{AnnProp})$
$(\overline{\texttt{owl:Class}}, \texttt{equivC}, \overline{\texttt{owl:Class}})$
$(\texttt{FunProp}, \texttt{equivC}, \texttt{FunProp})$
$(\texttt{InvFunProp}, \texttt{equivC}, \texttt{InvFunProp})$
$(\overline{\texttt{Nothing}}, \texttt{equivC}, \overline{\texttt{Nothing}})$
$(\overline{\texttt{ObjProp}}, \texttt{equivC}, \overline{\texttt{ObjProp}})$
$(\texttt{OntProp}, \texttt{equivC}, \texttt{OntProp})$
$(\texttt{SymProp}, \texttt{equivC}, \texttt{SymProp})$
$(\texttt{TransProp}, \texttt{equivC}, \texttt{TransProp})$
$(\texttt{priorVers}, \texttt{equivC}, \texttt{priorVers})$
$(\texttt{List}, \texttt{equivC}, \texttt{List})$
$(\texttt{CMemProp}, \texttt{equivC}, \texttt{CMemProp})$
$(\texttt{Datatype}, \texttt{equivC}, \texttt{Datatype})$
$(\texttt{Literal}, \texttt{equivC}, \texttt{Literal})$
$(\texttt{boolean}, \texttt{equivC}, \texttt{boolean})$
$(\overline{b_3}, \texttt{subClass}, \overline{b_3})$
$(\overline{b_3}, \texttt{subClass}, \overline{\texttt{owl:Class}})$
$(\overline{b_3}, \texttt{subClass}, \overline{\texttt{Thing}})$
$(b_8, \texttt{subClass}, \overline{\texttt{Thing}})$
$(b_8, \texttt{subClass}, \texttt{List})$
$(b_9, \texttt{subClass}, b_{10})$
$(b_{10}, \texttt{subClass}, b_9)$
$(\overline{b_{10}}, \texttt{subClass}, \overline{\texttt{owl:Class}})$
$(\overline{b_{10}}, \texttt{subClass}, \overline{\texttt{Thing}})$

$(\overline{b_{10}}, \mathtt{subClass}, \mathtt{List})$
$(\mathtt{eg{:}c}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{AllDiff}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \mathtt{FunProp})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \mathtt{InvFunProp})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \mathtt{ObjProp})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \mathtt{SymProp})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{AnnProp}, \mathtt{subClass}, \mathtt{TransProp})$
$(\overline{\mathtt{owl{:}Class}}, \mathtt{subClass}, \overline{\mathtt{Thing}})$

$(\mathtt{owl{:}Class}, \mathtt{subClass}, \mathtt{rdfs{:}Class})$
$(\mathtt{FunProp}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\mathtt{FunProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{InvFunProp}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\mathtt{InvFunProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{InvFunProp}, \mathtt{subClass}, \mathtt{TransProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{b_3})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, b_8)$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{b_{10}})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{eg{:}c})$

$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{AllDiff})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{AnnProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{\mathtt{owl{:}Class}})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{FunProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{InvFunProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{\mathtt{Nothing}})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{OntProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{SymProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \overline{\mathtt{Thing}})$

$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{TransProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{priorVers})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{List})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{CMemProp})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{Datatype})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{Literal})$
$(\overline{\mathtt{Nothing}}, \mathtt{subClass}, \mathtt{boolean})$
$(\overline{\mathtt{ObjProp}}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{ObjProp}, \mathtt{subClass}, \mathtt{Prop})$
$(\mathtt{OntProp}, \mathtt{subClass}, \mathtt{FunProp})$

$(\mathtt{OntProp}, \mathtt{subClass}, \mathtt{InvFunProp})$
$(\mathtt{OntProp}, \mathtt{subClass}, \mathtt{ObjProp})$
$(\mathtt{OntProp}, \mathtt{subClass}, \mathtt{SymProp})$
$(\mathtt{OntProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{OntProp}, \mathtt{subClass}, \mathtt{TransProp})$
$(\mathtt{SymProp}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\mathtt{SymProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{Thing}, \mathtt{subClass}, \mathtt{Resource})$
$(\mathtt{TransProp}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\mathtt{TransProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$

$(\mathtt{priorVers}, \mathtt{subClass}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{priorVers}, \mathtt{subClass}, \mathtt{FunProp})$
$(\mathtt{priorVers}, \mathtt{subClass}, \mathtt{InvFunProp})$
$(\mathtt{priorVers}, \mathtt{subClass}, \overline{\mathtt{ObjProp}})$
$(\mathtt{priorVers}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{priorVers}, \mathtt{subClass}, \mathtt{TransProp})$
$(\mathtt{List}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{CMemProp}, \mathtt{subClass}, \mathtt{FunProp})$
$(\mathtt{CMemProp}, \mathtt{subClass}, \mathtt{InvFunProp})$
$(\mathtt{CMemProp}, \mathtt{subClass}, \mathtt{ObjProp})$

$(\mathtt{CMemProp}, \mathtt{subClass}, \mathtt{SymProp})$
$(\mathtt{CMemProp}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{CMemProp}, \mathtt{subClass}, \mathtt{TransProp})$

$(\mathtt{Datatype}, \mathtt{subClass}, \mathtt{owl{:}Class})$
$(\mathtt{Datatype}, \mathtt{subClass}, \overline{\mathtt{Thing}})$
$(\mathtt{Literal}, \mathtt{subClass}, \overline{\mathtt{Thing}})$

$(\mathtt{boolean}, \mathtt{subClass}, \overline{\mathtt{Thing}})$

15 H_6

$(b_1, \mathtt{dstnctMems}, b_4)$
$(b_1, \mathtt{dstnctMems}, b_5)$
$(b_1, \mathtt{dstnctMems}, \mathtt{nil})$
$(\overline{b_3}, \mathtt{intersect}, b_5)$
$(\mathtt{owl{:}Class}, \mathtt{intersect}, b_3)$

$(\overline{\mathtt{Nothing}}, \mathtt{intersect}, b_4)$
$(\overline{\mathtt{Thing}}, \mathtt{intersect}, \mathtt{nil})$
$(b_3, \mathtt{oneOf}, b_3)$
$(b_7, \mathtt{oneOf}, b_2)$
$(\overline{\mathtt{Nothing}}, \mathtt{oneOf}, \mathtt{nil})$

$(\overline{b_3}, \mathtt{unionOf}, b_5)$
$(\mathtt{owl{:}Class}, \mathtt{unionOf}, b_3)$
$(\overline{\mathtt{Nothing}}, \mathtt{unionOf}, b_4)$
$(\overline{\mathtt{Nothing}}, \mathtt{unionOf}, \mathtt{nil})$

16 H_7

$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{inverseOf}, \mathtt{"true"^{\wedge\wedge}b})$
$(b_{11}, \mathtt{inverseOf}, b_{12})$
$(b_{12}, \mathtt{inverseOf}, b_{11})$

$(\mathtt{complmntOf}, \mathtt{inverseOf}, \mathtt{complmntOf})$
$(\mathtt{different}, \mathtt{inverseOf}, \mathtt{different})$
$(\mathtt{disjoint}, \mathtt{inverseOf}, \mathtt{disjoint})$

$(\mathtt{equivC}, \mathtt{inverseOf}, \mathtt{equivC})$
$(\overline{\mathtt{priorVers}}, \mathtt{inverseOf}, \overline{\mathtt{priorVers}})$
$(\mathtt{sameAs}, \mathtt{inverseOf}, \mathtt{sameAs})$

17 H_8

$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{FunProp})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{InvFunProp})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \overline{\mathtt{ObjProp}})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{SymProp})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{TransProp})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{Literal})$
$(\mathtt{"true"^{\wedge\wedge}b}, \mathtt{domain}, \mathtt{boolean})$
$(b_8, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$
$(b_8, \mathtt{domain}, \mathtt{Thing})$

$(b_{10}, \mathtt{domain}, \mathtt{Thing})$
$(b_{10}, \mathtt{domain}, \mathtt{List})$
$(\overline{b_{11}}, \mathtt{domain}, b_8)$
$(\overline{b_{11}}, \mathtt{domain}, \overline{b_{10}})$
$(\overline{b_{11}}, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$
$(\overline{b_{11}}, \mathtt{domain}, \mathtt{Thing})$
$(\overline{b_{11}}, \mathtt{domain}, \mathtt{List})$
$(b_{12}, \mathtt{domain}, b_9)$
$(b_{12}, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$
$(b_{12}, \mathtt{domain}, \mathtt{Thing})$

$(b_{12}, \mathtt{domain}, \mathtt{List})$
$(\mathtt{eg{:}p}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{complmntOf}, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{complmntOf}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{different}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{disjoint}, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$
$(\mathtt{disjoint}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{dstnctMems}, \mathtt{domain}, \mathtt{AllDiff})$
$(\mathtt{dstnctMems}, \mathtt{domain}, \mathtt{Thing})$
$(\mathtt{equivC}, \mathtt{domain}, \overline{\mathtt{owl{:}Class}})$

(equivC, domain, Thing)
(equivProp, domain, $\overline{\texttt{ObjProp}}$)
(equivProp, domain, Thing)
(intersect, domain, $\overline{\texttt{owl:Class}}$)
(intersect, domain, Thing)
(inverseOf, domain, $\overline{\texttt{ObjProp}}$)
(inverseOf, domain, Thing)
(oneOf, domain, $\overline{\texttt{owl:Class}}$)
(oneOf, domain, Thing)
($\overline{\texttt{priorVers}}$, domain, $\overline{b_3}$)

($\overline{\texttt{priorVers}}$, domain, b_8)
($\overline{\texttt{priorVers}}$, domain, $\overline{b_{10}}$)
($\overline{\texttt{priorVers}}$, domain, eg:c)
($\overline{\texttt{priorVers}}$, domain, AllDiff)
($\overline{\texttt{priorVers}}$, domain, AnnProp)
($\overline{\texttt{priorVers}}$, domain, $\overline{\texttt{owl:Class}}$)
($\overline{\texttt{priorVers}}$, domain, FunProp)
($\overline{\texttt{priorVers}}$, domain, InvFunProp)
($\overline{\texttt{priorVers}}$, domain, $\overline{\texttt{Nothing}}$)
($\overline{\texttt{priorVers}}$, domain, $\overline{\texttt{ObjProp}}$)

($\overline{\texttt{priorVers}}$, domain, OntProp)
($\overline{\texttt{priorVers}}$, domain, SymProp)
($\overline{\texttt{priorVers}}$, domain, Thing)
($\overline{\texttt{priorVers}}$, domain, TransProp)
($\overline{\texttt{priorVers}}$, domain, priorVers)
($\overline{\texttt{priorVers}}$, domain, List)
($\overline{\texttt{priorVers}}$, domain, CMemProp)
($\overline{\texttt{priorVers}}$, domain, Datatype)
($\overline{\texttt{priorVers}}$, domain, Literal)
($\overline{\texttt{priorVers}}$, domain, boolean)

(sameAs, domain, Thing)
(unionOf, domain, $\overline{\texttt{owl:Class}}$)
(unionOf, domain, Thing)
(first, domain, Thing)
(rest, domain, Thing)
(type, domain, Thing)
(domain, domain, ObjProp)
(domain, domain, Thing)
(range, domain, ObjProp)
(range, domain, Thing)

(subClass, domain, owl:Class)
(subClass, domain, Thing)
(subPropOf, domain, ObjProp)
(subPropOf, domain, Thing)
("true"^^b, range, FunProp)
("true"^^b, range, InvFunProp)
("true"^^b, range, $\overline{\texttt{ObjProp}}$)
("true"^^b, range, SymProp)
("true"^^b, range, Thing)
("true"^^b, range, TransProp)

("true"^^b, range, Literal)
("true"^^b, range, boolean)
(b_8, range, Thing)
(b_8, range, List)
(b_{10}, range, Thing)
(b_{10}, range, List)
($\overline{b_{11}}$, range, $\overline{b_{10}}$)
($\overline{b_{11}}$, range, $\overline{\texttt{owl:Class}}$)
($\overline{b_{11}}$, range, Thing)
($\overline{b_{11}}$, range, List)

(b_{12}, range, b_8)
(b_{12}, range, $\overline{b_{10}}$)
(b_{12}, range, $\overline{\texttt{owl:Class}}$)
(b_{12}, range, Thing)
(b_{12}, range, List)
(eg:p, range, Thing)
(eg:p, range, Literal)
(complmntOf, range, $\overline{\texttt{owl:Class}}$)
(complmntOf, range, Thing)
(different, range, Thing)

(disjoint, range, $\overline{\texttt{owl:Class}}$)
(disjoint, range, Thing)
(dstnctMems, range, Thing)
(dstnctMems, range, List)
(equivC, range, $\overline{\texttt{owl:Class}}$)
(equivC, range, Thing)
(equivProp, range, $\overline{\texttt{ObjProp}}$)
(equivProp, range, Thing)
(intersect, range, Thing)
(intersect, range, List)

(inverseOf, range, $\overline{\texttt{ObjProp}}$)
(inverseOf, range, Thing)
(oneOf, range, Thing)
(oneOf, range, List)
($\overline{\texttt{priorVers}}$, range, $\overline{b_3}$)
($\overline{\texttt{priorVers}}$, range, b_8)
($\overline{\texttt{priorVers}}$, range, $\overline{b_{10}}$)
($\overline{\texttt{priorVers}}$, range, eg:c)
($\overline{\texttt{priorVers}}$, range, AllDiff)
($\overline{\texttt{priorVers}}$, range, AnnProp)

($\overline{\texttt{priorVers}}$, range, $\overline{\texttt{owl:Class}}$)
($\overline{\texttt{priorVers}}$, range, FunProp)
($\overline{\texttt{priorVers}}$, range, InvFunProp)
($\overline{\texttt{priorVers}}$, range, $\overline{\texttt{Nothing}}$)
($\overline{\texttt{priorVers}}$, range, $\overline{\texttt{ObjProp}}$)
($\overline{\texttt{priorVers}}$, range, OntProp)
($\overline{\texttt{priorVers}}$, range, SymProp)
($\overline{\texttt{priorVers}}$, range, Thing)
($\overline{\texttt{priorVers}}$, range, TransProp)
($\overline{\texttt{priorVers}}$, range, priorVers)

($\overline{\texttt{priorVers}}$, range, List)
($\overline{\texttt{priorVers}}$, range, CMemProp)
($\overline{\texttt{priorVers}}$, range, Datatype)
($\overline{\texttt{priorVers}}$, range, Literal)
($\overline{\texttt{priorVers}}$, range, boolean)
(sameAs, range, Thing)
(unionOf, range, Thing)
(unionOf, range, List)
(first, range, $\overline{\texttt{owl:Class}}$)
(first, range, Thing)

(rest, range, Thing)
(type, range, owl:Class)
(type, range, Thing)
(domain, range, owl:Class)
(domain, range, Thing)
(range, range, owl:Class)
(range, range, Thing)
(subClass, range, owl:Class)
(subClass, range, Thing)
(subPropOf, range, ObjProp)

(subPropOf, range, Thing)

18 H_9

("true"^^b, equivProp, "true"^^b)
(b_{10}, equivProp, b_{10})
($\overline{b_{11}}$, equivProp, $\overline{b_{11}}$)
(b_{12}, equivProp, b_{12})
(eg:p, equivProp, eg:p)
(complmntOf, equivProp, complmntOf)
(different, equivProp, different)
(disjoint, equivProp, disjoint)
(dstnctMems, equivProp, dstnctMems)
(equivC, equivProp, equivC)

(equivProp, equivProp, equivProp)
(intersect, equivProp, intersect)
(inverseOf, equivProp, inverseOf)
(oneOf, equivProp, oneOf)
($\overline{\texttt{priorVers}}$, equivProp, $\overline{\texttt{priorVers}}$)
(sameAs, equivProp, sameAs)
(unionOf, equivProp, unionOf)
(first, equivProp, first)
(rest, equivProp, rest)
(type, equivProp, type)

(domain, equivProp, domain)
(range, equivProp, range)
(subClass, equivProp, subClass)
(subPropOf, equivProp, subPropOf)
("true"^^b, subPropOf, equivProp)
("true"^^b, subPropOf, inverseOf)
("true"^^b, subPropOf, sameAs)
("true"^^b, subPropOf, subPropOf)
(b_8, subPropOf, different)
(b_9, subPropOf, b_{11})

$(b_{10}, \texttt{subPropOf}, \texttt{different})$
$(b_{11}, \texttt{subPropOf}, b_9)$
$(\overline{b_{11}}, \texttt{subPropOf}, \texttt{different})$
$(\overline{b_{11}}, \texttt{subPropOf}, \texttt{disjoint})$
$(\overline{b_{11}}, \texttt{subPropOf}, \texttt{subClass})$
$(b_{12}, \texttt{subPropOf}, b_8)$
$(b_{12}, \texttt{subPropOf}, \texttt{different})$
$(b_{12}, \texttt{subPropOf}, \texttt{disjoint})$
$(b_{12}, \texttt{subPropOf}, \texttt{oneOf})$
$(\texttt{eg:p}, \texttt{subPropOf}, \texttt{different})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{oneOf})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \overline{\texttt{priorVers}})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{sameAs})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{unionOf})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{first})$
$(\texttt{complmntOf}, \texttt{subPropOf}, \texttt{different})$
$(\texttt{complmntOf}, \texttt{subPropOf}, \texttt{disjoint})$
$(\texttt{dstnctMems}, \texttt{subPropOf}, \texttt{different})$
$(\texttt{equivC}, \texttt{subPropOf}, \texttt{subClass})$
$(\texttt{equivProp}, \texttt{subPropOf}, \texttt{subPropOf})$
$(\texttt{intersect}, \texttt{subPropOf}, \texttt{different})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{"true"\^{}\^{}b})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, b_8)$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, b_{10})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \overline{b_{11}})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{rest})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{type})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{domain})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{range})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{subClass})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, b_{12})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{eg:p})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{complmntOf})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{different})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{disjoint})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{dstnctMems})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{equivC})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{equivProp})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{intersect})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{inverseOf})$
$(\overline{\texttt{priorVers}}, \texttt{subPropOf}, \texttt{subPropOf})$
$(\texttt{unionOf}, \texttt{subPropOf}, \texttt{different})$
$(\texttt{first}, \texttt{subPropOf}, \texttt{different})$
$(\texttt{rest}, \texttt{subPropOf}, \texttt{different})$

References

[1] Carroll, J.J., Turner, D.: The Consistency of OWL Full. Technical Report, HP Labs (2008) HPL-2008-58.

[2] Carroll, J.J., Turner, D.: The Consistency of OWL Full (with proofs). Technical Report, HP Labs (2008) HPL-2008-59.

[3] Hayes, P.: RDF Semantics. W3C recommendation, W3C (February 2004) http://www.w3.org//TR/2004/REC-rdf-mt-20040210/.

[4] Malhotra, A., Biron, P.V.: XML Schema Part 2: Datatypes Second Edition. W3C Recommendation, W3C (October 2004) http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/.

[5] ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. J. Web Semantics **3**(2-3) (2005) 79–115