



OWL 2 RL in RIF

W3C Working Group Note 28 October 2012

This version:

<http://www.w3.org/2005/rules/wg/draft/ED-rif-owl-rl-20121028/>

Latest editor's draft:

<http://www.w3.org/2005/rules/wg/draft/rif-owl-rl/>

Previous version:

<http://www.w3.org/2005/rules/wg/draft/NOTE-rif-owl-rl-20100622/>

Editors:

Dave Reynolds, Hewlett Packard Laboratories

A [color-coded version of this document showing changes made since the previous version](#) is also available.

This document is also available in these non-normative formats: [PDF version](#).

Copyright © 2012 W3C® (MIT, ERCIM, Keio). All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

This document shows how OWL 2 RL can be implemented using RIF. It provides an analysis of how to represent OWL 2 RL inference rules within RIF Core. The OWL 2 RL inference rules can be implemented both via a fixed RIF Core rule set (Appendix 7) and via a mapping algorithm which converts an OWL 2 RL ontology to a customized RIF Core rule set (Appendix 8).

Status of this Document

May Be Superseded

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

Set of Documents

This document is being published as one of a set of 12 documents:

1. [RIF Overview](#)
2. [RIF Core Dialect](#)
3. [RIF Basic Logic Dialect](#)
4. [RIF Production Rule Dialect](#)
5. [RIF Framework for Logic Dialects](#)
6. [RIF Datatypes and Built-Ins 1.0](#)
7. [RIF RDF and OWL Compatibility](#)
8. [OWL 2 RL in RIF](#) (this document)
9. [RIF Combination with XML data](#)
10. [RIF In RDF](#)
11. [RIF Test Cases](#)
12. [RIF Primer](#)

Summary of Changes

There have been no [substantive](#) changes since the [previous version](#). For details on the minor changes see the [change log](#) and [color-coded diff](#).

Please Send Comments

Please send any comments to public-rif-comments@w3.org ([public archive](#)). Although work on this document by the [Rule Interchange Format \(RIF\) Working Group](#) is complete, comments may be addressed in the [errata](#) or in future revisions. Open discussion among developers is welcome at public-rif-dev@w3.org ([public archive](#)).

No Endorsement

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

Patents

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). The group does not expect this document to become a W3C Recommendation. W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

- [1 Introduction](#)
- [2 Summary of issues](#)
- [3 Background OWL 2 RL](#)
- [4 Analysis of OWL 2 RL rules](#)
 - [4.1 Triple pattern rules](#)
 - [4.2 Inconsistency rules](#)
 - [4.3 List rules](#)
 - [4.3.1 Inconsistent pairs rules](#)
 - [4.3.2 Property chain rule](#)
 - [4.3.3 HasKey rule](#)
 - [4.3.4 Forward intersectionOf rule](#)

if	then
T(s ₁ , p ₁ , o ₁)	T(sr ₁ , pr ₁ , or ₁)
...	...
T(s _n , p _n , o _n)	T(sr _m , pr _m , or _m)

where each argument to the *T* predicate may be a variable, an IRI or literal value.

Such rules can be expressed in the RIF Core presentation syntax as:

```
Group (
  Forall ?v1 ... ?v10 (
    sr1[pr1->or1] :- And( s1[p1->o1] ... sn[pn->on] ) )
  ...
  Forall ?vm ... ?vm0 (
    srm[prm->orm] :- And( s1[p1->o1] ... sn[pn->on] ) )
)
```

where ?v₁ ... ?v₁₀ are the variables which occur in the *i*th rule.

4.2 Inconsistency rules

A number of the OWL 2 RL rules detect inconsistencies in the original RDF graph. They express this by deriving a distinguished propositional symbol `false`.

For example:

if	then
T(?p ₁ , owl:propertyDisjointWith, ?p ₂) T(?x, ?p ₁ , ?y) T(?x, ?p ₂ , ?y)	false

We translate such rules using a distinguished nullary predicate `rif:error()`. If this predicate is derived then the original RDF graph is inconsistent under OWL 2 RL semantics.

Note: in practice implementers may wish to replace `rif:error()` with a unary predicate which can be used to convey diagnostic information on the nature and source of the inconsistency that has been detected.

Thus the above rule would be expressed in RIF Core as:

```
Forall ?p1 ?p2 ?x ?y (
  rif:error() :-
    And( ?p1[owl:propertyDisjointWith->?p2] ?x[?p1->?y] ?x[?p2->?y] )
)
```

4.3 List rules

A number of the OWL 2 RL rules involve processing OWL expressions which include RDF lists. Such rules are expressed in OWL 2 RL as templates which, for each possible list length, would correspond to a set of Triple Pattern rules.

There are two ways we can approach this in RIF. We can rewrite such rules as recursive rules that traverse the RDF list links or we can introduce the notion of a preprocessor which takes an OWL2 RL ontology in RDF syntax and translates it to a corresponding RIF rule set in which the templates rules have been instantiated for the lists that actually occur in the source ontology. This is possible because none of the rules is able to deduce new list entries for such lists within the syntactic constraints of OWL 2 RL.

The approach of using a fixed recursive rule set is appropriate to enable a single RIF OWL 2 RL ruleset to be published. The translation approach yields rules which are likely to perform better in practice.

We discuss the direct forms of the List Rules here and then describe in [Appendix: OWL 2 RL to RIF translation](#) the translation algorithm for the pre-processor approach.

There are several different patterns of List Rules in the OWL 2 RL ruleset, we discuss each group of rules in turn organized by the pattern involved.

4.3.1 Inconsistent pairs rules

These rules check whether any pair of entries from a list match a certain criterion and if they do an error is signaled (derivation of the `false` propositional symbol).

eq-diff2	T(?x, rdf:type, owl:AllDifferent) T(?x, owl:members, ?y) LIST[?y, ?z ₁ , ..., ?z _n] T(?z _i , owl:sameAs, ?z _j)	false	for each 1 ≤ i < j ≤ n
prp-adp	T(?x, rdf:type, owl:AllDisjointProperties) T(?x, owl:members, ?y) LIST[?y, ?p ₁ , ..., ?p _n] T(?u, ?p _i , ?v) T(?u, ?p _j , ?v)	false	for each 1 ≤ i < j ≤ n
cax-adc	T(?x, rdf:type, owl:AllDisjointClasses) T(?x, owl:members, ?y) LIST[?y, ?c ₁ , ..., ?c _n] T(?z, rdf:type, ?c _i) T(?z, rdf:type, ?c _j)	false	for each 1 ≤ i < j ≤ n

These rules can be directly translated with the assistance of a utility predicate to select any element from an RDF list.

```
(* <#eq-diff2> *)
Forall ?x ?y ?z1 ?z2 ?iz1 ?iz2 (
  rif:error() :- And (
    ?x[rdf:type -> owl:AllDifferent]
    ?x[owl:members -> ?y]
    External(pred:list-contains(?y ?z1)) ?iz1 = External(func:index-of(?y ?z1))
    External(pred:list-contains(?y ?z2)) ?iz2 = External(func:index-of(?y ?z2))
    External( pred:numeric-not-equal ( ?iz1 ?iz2 ) )
    ?z1[owl:sameAs->?z2] ) )
(* <#eq-diff3> *)
```

```

Forall ?x ?y ?z1 ?z2 ?iz1 ?iz2 (
  rif:error() :- And (
    ?x[rdf:type -> owl:AllDifferent]
    ?x[owl:distinctMembers -> ?y]
    External(pred:list-contains(?y ?z1)) ?iz1 = External(func:index-of(?y ?z1))
    External(pred:list-contains(?y ?z2)) ?iz2 = External(func:index-of(?y ?z2))
    External( pred:numeric-not-equal ( ?iz1 ?iz2 ) )
    ?z1[owl:sameAs->?z2] ) )

(* <#prp-adp> *)
Forall ?x ?y ?o ?v ?l ?ix ?iy ?r (
  rif:error() :- And (
    ?r[rdf:type -> owl:AllDisjointProperties]
    ?r[owl:members -> ?l]
    External(pred:list-contains(?l ?x)) ?ix = External(func:index-of(?l ?x))
    External(pred:list-contains(?l ?y)) ?iy = External(func:index-of(?l ?y))
    External( pred:numeric-not-equal ( ?ix ?iy ) )
    ?o[?x->?v ?y->?v] ) )

(* <#cax-adc> *)
Forall ?x ?y ?o ?l ?ix ?iy ?r (
  rif:error() :- And (
    ?r[rdf:type -> owl:AllDisjointClasses]
    ?r[owl:members -> ?l]
    External(pred:list-contains(?l ?x)) ?ix = External(func:index-of(?l ?x))
    External(pred:list-contains(?l ?y)) ?iy = External(func:index-of(?l ?y))
    External( pred:numeric-not-equal ( ?ix ?iy ) )
    ?o[rdf:type->?x rdf:type->?y] ) )
)
    
```

Note: it might be tempting to refactor the above rules by defining a `_pairFromList` predicate which can select a disjoint pair of elements from a list. However, such a predicate would be unsafe.

4.3.2 Property chain rule

This rule expands the list into a chain of properties that need to be checked in the rule body.

prp-spo2	$T(?p, owl:propertyChainAxiom, ?x)$ $LIST[?x, ?p_1, \dots, ?p_n]$ $T(?u_1, ?p_1, ?u_2)$ $T(?u_2, ?p_2, ?u_3)$ \dots $T(?u_n, ?p_n, ?u_{n+1})$	$T(?u_1, ?p, ?u_{n+1})$
----------	--	-------------------------

Which can be translated by the RIF Core rule set:

```

(* <#prp-spo2> *)
Forall ?p ?last ?pc ?start (
  ?start[?p->?last] :- And (
    ?p[owl:propertyChainAxiom->?pc]
    _checkChain(?start ?pc ?last) ) )

Forall ?start ?pc ?last ?p ?tl (
  _checkChain(?start ?pc ?last) :- And (
    ?pc[rdf:first->?p rdf:rest->?tl]
    ?start[?p->?next]
    _checkChain(?next ?tl ?last) ) )

Forall ?start ?pc ?last ?p (
  _checkChain(?start ?pc ?last) :- And (
    ?pc[rdf:first->?p rdf:rest->rdf:nil]
    ?start[?p->?last] ) )
    
```

When writing logic rules for predicates such as `_checkChain` it would be more common to have the first recursive rule and then a final "terminating" fact:

```

Forall ?start (
  _checkChain(?start rdf:nil ?start) )
    
```

However, this universal fact is not safe and thus not within RIF Core. We can work around this by the trick of unfolding the final fact into the recursive rule to yield the pair of rules shown above. In this way the rule set is still executable by a forward chaining strategy. Note that such a strategy will generate a large number of irrelevant `_checkChain` assertions since it will apply to any RDF List in the source ontology irrespective of whether that list is part of a `owl:propertyChainAxiom`.

An approach which would give better effective performance for a forward chaining strategy would be to recurse down the property chain "marking" list cells that will need to be checked. Such as:

```

Forall ?p ?pc (
  _needCheckChain(?pc) :- And (
    ?p[owl:propertyChainAxiom->?pc] ) )

Forall ?x ?y (
  _needCheckChain(?y) :- And (
    _needCheckChain(?x)
    ?x[rdf:rest->?y] ) )

Forall ?start ?pc ?last ?p ?tl (
  _checkChain(?start ?pc ?last) :- And (
    _needCheckChain(?pc)
    ?pc[rdf:first->?p rdf:rest->?tl]
    ?start[?p->?next]
    _checkChain(?next ?tl ?last) ) )

Forall ?start ?pc ?last ?p (
  _checkChain(?start ?pc ?last) :- And (
    _needCheckChain(?pc)
    ?pc[rdf:first->?p rdf:rest->rdf:nil]
    ?start[?p->?last] ) )
    
```

However, since such optimizations further obscure the rules we will omit them from the rule set described here.

An alternative strategy, if performant OWL 2 RL reasoning is required, would be for a forward chaining engine to provide custom builtins for the list scanning needed here. Detailed description of such an approach is beyond the scope of this note.

We anticipate that practical rule-based OWL 2 RL implementations in RIF will either introduce such optimizations or employ an ontology translation approach, such as that described in appendix B.

4.3.3 HasKey rule

This rule checks that two individuals are the same by virtual of having the same value for each of a list of key properties.

prp-key	<pre>T(?c, owl:hasKey, ?u) LIST[?u, ?p1, ..., ?pn] T(?x, rdf:type, ?c) T(?x, ?p1, ?z1) ... T(?x, ?pn, ?zn) T(?y, rdf:type, ?c) T(?y, ?p1, ?z1) ... T(?y, ?pn, ?zn)</pre>	T(?x, owl:sameAs, ?y)
---------	--	-----------------------

This can be translated by the RIF rule set:

```
(* <#prp-key> *)
Forall ?x ?y ?c ?u (
  ?x[owl:sameAs->?y] :- And (
    ?c[owl:hasKey->?u] ?x[rdf:type->?c] ?y[rdf:type->?c]
    _sameKey(?u ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rdf:first->?key rdf:rest->?tl]
    ?x[?key->?v] ?y[?key->?v]
    _sameKey(?tl ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rdf:first->?key rdf:rest->rdf:nil]
    ?x[?key->?v] ?y[?key->?v] ) )
```

Similar considerations about safety and performance apply here as in the previous section.

4.3.4 Forward intersectionOf rule

This pattern involves a test for all members of a list.

cls-int1	<pre>T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn] T(?y, rdf:type, ?c1) T(?y, rdf:type, ?c2) ... T(?y, rdf:type, ?cn)</pre>	T(?y, rdf:type, ?c)
----------	---	---------------------

This can be translated by the RIF rule set:

```
(* <#cls-int1> *)
Forall ?y ?c ?l (
  ?y[rdf:type->?c] :- And (
    ?c[owl:intersectionOf->?l]
    _allTypes(?l ?y) ) )

Forall ?l ?y ?ty ?tl (
  _allTypes(?l ?y) :- And (
    ?l[rdf:first->?ty rdf:rest->?tl]
    ?y[rdf:type->?ty]
    _allTypes(?tl ?y) ) )

Forall ?l ?y ?ty (
  _allTypes(?l ?y) :- And (
    ?l[rdf:first->?ty rdf:rest->rdf:nil]
    ?y[rdf:type->?ty] ) )
```

Similar considerations on safety and performance apply here as in the previous two sections.

4.3.5 Simple member rules

This pattern involves a test for each member of a list.

cls-uni	<pre>T(?c, owl:unionOf, ?x) LIST[?x, ?c1, ..., ?cn] T(?y, rdf:type, ?c1)</pre>	T(?y, rdf:type, ?c)	for each 1 ≤ i ≤ n
cls-oo	<pre>T(?c, owl:oneOf, ?x) LIST[?x, ?y1, ..., ?yn]</pre>	T(?yi, rdf:type, ?c)	for each 1 ≤ i ≤ n
cls-int2	<pre>T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn] T(?y, rdf:type, ?c)</pre>	<pre>T(?y, rdf:type, ?c1) T(?y, rdf:type, ?c2) ... T(?y, rdf:type, ?cn)</pre>	
scm-int	<pre>T(?c, owl:intersectionOf, ?x) LIST[?x, ?c1, ..., ?cn]</pre>	<pre>T(?c, rdfs:subClassOf, ?c1) T(?c, rdfs:subClassOf, ?c2)</pre>	

		... T(?c, rdfs:subClassOf, ?cn)
scm-uni	T(?c, owl:unionOf, ?x) LIST[?x, ?c1, ..., ?cn]	T(?c1, rdfs:subClassOf, ?c) T(?c2, rdfs:subClassOf, ?c) ... T(?cn, rdfs:subClassOf, ?c)

These can be translated by the RIF rule set:

```
(* <#cls-uni> *)
Forall ?y ?c ?l ?ci (
  ?y[rdf:type->?c] :- And (
    ?c[owl:unionOf->?l]
    External(pred:list-contains(?l ?ci))
    ?y[rdf:type->?ci] ))

(* <#cls-oo> *)
Forall ?yi ?c ?l (
  ?yi[rdf:type->?c] :- And (
    ?c[owl:oneOf->?l]
    External(pred:list-contains(?l ?yi)) ))

(* <#cls-int2> *)
Forall ?y ?c ?ci ?l (
  ?y[rdf:type->?ci] :- And (
    ?c[owl:intersectionOf->?l]
    External(pred:list-contains(?l ?ci))
    ?y[rdf:type->?c] ))

(* <#scm-int> *)
Forall ?c ?ci ?l (
  ?c[rdfs:subClassOf->?ci] :- And (
    ?c[owl:intersectionOf->?l]
    External(pred:list-contains(?l ?ci)) ))

(* <#scm-uni> *)
Forall ?c ?ci ?l (
  ?ci[rdfs:subClassOf->?c] :- And (
    ?c[owl:unionOf->?l]
    External(pred:list-contains(?l ?ci)) ))
```

where the `_member` predicate was defined earlier.

4.4 Datatype rules

OWL 2 RL specifies five groups of rules to capture the semantics of supported datatypes. These are:

if	then		
dt-type1	true	T(df, rdf:type, rdfs:Datatype)	for each datatype dt supported in OWL 2 RL
dt-type2	true	T(lt, rdf:type, dt)	for each literal lt and each datatype dt supported in OWL 2 RL such that the data value of lt is contained in the value space of dt
dt-eq	true	T(lt1, owl:sameAs, lt2)	for all literals lt1 and lt2 with the same data value
dt-diff	true	T(lt1, owl:differentFrom, lt2)	for all literals lt1 and lt2 with different data values
dt-not-type	T(lt, rdf:type, dt)	false	for each literal lt and each datatype dt supported in OWL 2 RL such that the data value of lt is not contained in the value space of dt

4.4.1 Datatypes supported

The RIF and OWL 2 RL working groups have now aligned the set of datatypes so that all the datatypes required by OWL 2 RL are supported by RIF.

4.4.2 Datatype rules and safety

A naive translation of the datatype rules would require unsafe RIF rules. For example, one might be tempted to translate `dt-eq` as:

```
Forall ?l1 ?l2 ( ?l1[owl:sameAs->l2] :- ?l1 = ?l2 )
```

To ensure safety we must ground the variables in the relevant vocabulary. Referring to [Theorem-1](#) we can see that the entailment checks supported in O_2 are limited and do not include direct checks for inequality, type or non-type of literals. Thus we only need to ground the datatype rules in the literal vocabulary of O_1 . While O_2 can include axioms of the form `DataPropertyAssertion(DP a v)` since RIF and OWL 2 have the same notion of identity over those datatypes which both support then such direct equality entailments are automatically satisfied.

Thus a sufficient rendering of `dt-eq` would be:

```
Forall ?l1 ?l2 ?s1 ?s2 ?p1 ?p2
( ?l1[owl:sameAs->l2] :- And(
  ?s1[?p1->?l1] ?s2[?p2->?l2] ?l1 = ?l2 ))
```

However, we can go further than this. Since the entailment checks are limited to only indirect consequences of these rules then we can fold the datatype rules directly into the rules which include `owl:sameAs` in their premise. This avoids the generation of a quadratic number of `owl:sameAs` and `owl:differentFrom` assertions for all literals.

4.4.3 Rules for supported datatypes

The RIF representation of the data type rules for supported datatypes is reasonably straightforward.

Type membership

```
(* <#dt-type2> *)
Group (
  Forall ?lt (
```

```

rif:error() :- And (
  ?lt[rdf:type->rdf:PlainLiteral] External(pred:is-literal-not-PlainLiteral( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:decimal] External(pred:is-literal-not-decimal( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:integer] External(pred:is-literal-not-integer( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:nonNegativeInteger] External(pred:is-literal-not-nonNegativeInteger( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:nonPositiveInteger] External(pred:is-literal-not-nonPositiveInteger( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:positiveInteger] External(pred:is-literal-not-positiveInteger( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:negativeInteger] External(pred:is-literal-not-negativeInteger( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:long] External(pred:is-literal-not-long( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:int] External(pred:is-literal-not-int( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:short] External(pred:is-literal-not-short( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:byte] External(pred:is-literal-not-byte( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedLong] External(pred:is-literal-not-unsignedLong( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedInt] External(pred:is-literal-not-unsignedInt( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedShort] External(pred:is-literal-not-unsignedShort( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedByte] External(pred:is-literal-not-unsignedByte( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:float] External(pred:is-literal-not-float( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:double] External(pred:is-literal-not-double( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:string] External(pred:is-literal-not-string( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:normalizedString] External(pred:is-literal-not-normalizedString( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:token] External(pred:is-literal-not-token( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:language] External(pred:is-literal-not-language( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:Name] External(pred:is-literal-not-Name( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:NCName] External(pred:is-literal-not-NCName( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:NMTOKEN] External(pred:is-literal-not-NMTOKEN( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:boolean] External(pred:is-literal-not-boolean( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:hexBinary] External(pred:is-literal-not-hexBinary( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:base64Binary] External(pred:is-literal-not-base64Binary( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:anyURI] External(pred:is-literal-not-anyURI( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:dateTime] External(pred:is-literal-not-dateTime( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:dateTimeStamp] External(pred:is-literal-not-dateTimeStamp ( ?lt )) )
)

```

Note that these rules assert that each literal is of type `rdfs:Literal`, as well as being of the direct type indicated by the type guard. This is because the value space of `rdfs:Literal` includes the union of all of the separate literal value spaces. RIF does not provide direct support for `rdfs:Literal` and so there is no simple `pred:is-literal` guard available to enable a more transparent ruleset design.

Type checking

```
(* <#dt-not-type> *)
Group (
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->rdf:PlainLiteral] External(pred:is-literal-not-PlainLiteral( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:decimal] External(pred:is-literal-not-decimal( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:integer] External(pred:is-literal-not-integer( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:nonNegativeInteger] External(pred:is-literal-not-nonNegativeInteger( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:nonPositiveInteger] External(pred:is-literal-not-nonPositiveInteger( ?lt )) )
  )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:positiveInteger] External(pred:is-literal-not-positiveInteger( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:negativeInteger] External(pred:is-literal-not-negativeInteger( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:long] External(pred:is-literal-not-long( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:int] External(pred:is-literal-not-int( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:short] External(pred:is-literal-not-short( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:byte] External(pred:is-literal-not-byte( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:unsignedLong] External(pred:is-literal-not-unsignedLong( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:unsignedInt] External(pred:is-literal-not-unsignedInt( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:unsignedShort] External(pred:is-literal-not-unsignedShort( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:unsignedByte] External(pred:is-literal-not-unsignedByte( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:float] External(pred:is-literal-not-float( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:double] External(pred:is-literal-not-double( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:string] External(pred:is-literal-not-string( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:normalizedString] External(pred:is-literal-not-normalizedString( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:token] External(pred:is-literal-not-token( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:language] External(pred:is-literal-not-language( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:Name] External(pred:is-literal-not-Name( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:NCName] External(pred:is-literal-not-NCName( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:NMTOKEN] External(pred:is-literal-not-NMTOKEN( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:boolean] External(pred:is-literal-not-boolean( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:hexBinary] External(pred:is-literal-not-hexBinary( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:base64Binary] External(pred:is-literal-not-base64Binary( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:anyURI] External(pred:is-literal-not-anyURI( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:dateTime] External(pred:is-literal-not-dateTime( ?lt )) )
  )
  Forall ?lt ?dt (
    rif:error() :- And (
      ?lt[rdftype->xsd:dateTimeStamp ] External(pred:is-literal-not-dateTimeStamp ( ?lt )) )
  )
)
```

Equality and inequality As noted above we can fold the literal equality and inequality checks into the rules which refer to them, in this case the only affected rule is eq-diff1:


```
(* <#eq-diff1> *)
Forall ?x ?y (
  rif:error() :- And(
    ?x[owl:sameAs->?y]
    ?x[owl:differentFrom->?y] ) )
```

So after folding the grounded literal rules for dt-eq and dt-diff in we arrive at:

```
(* <#eq-diff1-literal1> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x[owl:sameAs->?y]
    External(pred:literal-not-identical(?x ?y)) ) )

(* <#eq-diff1-literal2> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x = ?y
    ?x[owl:differentFrom->?y] ) )
```

Fixed vocabulary rules These are trivial asserted facts of the form:

```
(* <#dt-type1-PlainLiteral> *) rdf:PlainLiteral [rdf:type -> rdfs:Datatype] etc
```

5 References

[OWL2-Profiles]

[OWL 2 Web Ontology Language: Profiles](http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/), B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, Editors, W3C Candidate Recommendation, 11 June 2009, <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>. Latest version available at <http://www.w3.org/TR/owl2-profiles/>.

[OWL2-RDF-Mapping]

[OWL 2 Web Ontology Language Mapping to RDF Graphs](http://www.w3.org/TR/2009/CR-owl2-mapping-to-rdf-20090611/), P.F. Patel-Schneider, B. Motik, Editors, W3C Candidate Recommendation, 11 June 2009, <http://www.w3.org/TR/2009/CR-owl2-mapping-to-rdf-20090611/>. Latest version available at <http://www.w3.org/TR/owl2-mapping-to-rdf/>.

[OWL2-RDF-Semantics]

[OWL 2 Web Ontology Language RDF-Based Semantics](http://www.w3.org/TR/2009/CR-owl2-rdf-based-semantics-20090611/), M. Schneider, Editor, W3C Candidate Recommendation, 11 June 2009, <http://www.w3.org/TR/2009/CR-owl2-rdf-based-semantics-20090611/>. Latest version available at <http://www.w3.org/TR/owl2-rdf-based-semantics/>.

[RIF RDF and OWL Compatibility]

[RIF RDF and OWL Compatibility](http://www.w3.org/2005/rules/wg/draft/ED-rif-rdf-owl-20121028/) Jos de Bruijn, Chris Welty, eds. W3C Editor's Draft, 28 October 2012, <http://www.w3.org/2005/rules/wg/draft/ED-rif-rdf-owl-20121028/>. Latest version available at <http://www.w3.org/2005/rules/wg/draft/rif-rdf-owl/>.

6 Acknowledgements

This document is the product of the Rules Interchange Format (RIF) Working Group (see below) whose members deserve recognition for their time and commitment. The editor extends special thanks to Jos de Bruijn and Axel Polleres for their thorough reviews and helpful comments; the working group chairs, Chris Welty and Christian de Sainte-Marie, for their invaluable technical help and inspirational leadership; and W3C staff contact Sandro Hawke, a constant source of ideas, help, and feedback.

The regular attendees at meetings of the Rule Interchange Format (RIF) Working Group at the time of the publication were: Adrian Paschke (Freie Universitaet Berlin), Axel Polleres (DERI), Chris Welty (IBM), Christian de Sainte Marie (ILOG), Dave Reynolds (HP), Gary Hallmark (ORACLE), Harold Boley (NRC), Hassan Ait-Kaci (ILOG), Jos de Bruijn (FUB), Leora Morgenstern (IBM), Michael Kifer (Stony Brook), Mike Dean (BBN), Sandro Hawke (W3C/MIT), and Stella Mitchell (IBM).

7 Appendix: OWL 2 RL ruleset - presentation syntax

In this appendix we provide a RIF Core presentation syntax translation of the complete OWL 2 RL ruleset. We divide these rules into three documents - the simple triple rules (mechanically translated from the OWL 2 RL rule format), the List rules and the datatype rules.

Editor's Note: These rules are believed to be correct. However, since the original version was developed various manual edits have been made to conform to changes in RIF and OWL 2 RL. Mechanical verification will be required before final publication.

Simple triple rules

```
Document (
  Prefix(rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(owl <http://www.w3.org/2002/07/owl#>)
  Prefix(xsd <http://www.w3.org/2001/XMLSchema#>)
  Prefix(rif <http://www.w3.org/2007/rif#>)
  Prefix(func <http://www.w3.org/2007/rif-builtin-function#>)
  Prefix(pred <http://www.w3.org/2007/rif-builtin-predicate#>)
  Prefix(dc <http://purl.org/dc/terms/>)
  Group (

    (* <#eq-ref> *)
    Forall ?p ?o ?s (
      ?s[owl:sameAs->?s] :- ?s[?p->?o])

    (* <#eq-ref1> *)
    Forall ?p ?o ?s (
      ?p[owl:sameAs->?p] :- ?s[?p->?o])

    (* <#eq-ref2> *)
    Forall ?p ?o ?s (
      ?o[owl:sameAs->?o] :- ?s[?p->?o])

    (* <#eq-sym> *)
    Forall ?x ?y (
      ?y[owl:sameAs->?x] :- ?x[owl:sameAs->?y])

    (* <#eq-trans> *)
    Forall ?x ?z ?y (
      ?x[owl:sameAs->?z] :- And(
        ?x[owl:sameAs->?y]
        ?y[owl:sameAs->?z] ) )
```

```

(* <#eq-rep-s> *)
Forall ?p ?o ?s ?s2 (
  ?s2[?p->?o] :- And(
    ?s[owl:sameAs->?s2]
    ?s[?p->?o] ))

(* <#eq-rep-p> *)
Forall ?p ?o ?s ?p2 (
  ?s[?p2->?o] :- And(
    ?p[owl:sameAs->?p2]
    ?s[?p->?o] ))

(* <#eq-rep-o> *)
Forall ?p ?o ?s ?o2 (
  ?s[?p->?o2] :- And(
    ?o[owl:sameAs->?o2]
    ?s[?p->?o] ))

(* <#eq-diff1> *)
Forall ?x ?y (
  rif:error() :- And(
    ?x[owl:sameAs->?y]
    ?x[owl:differentFrom->?y] ))

(* <#prp-ap-label> *)
rdfs:label[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-comment> *)
rdfs:comment[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-seeAlso> *)
rdfs:seeAlso[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-isDefinedBy> *)
rdfs:isDefinedBy[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-deprecated> *)
owl:deprecated[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-priorVersion> *)
owl:priorVersion[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-backwardCompatibleWith> *)
owl:backwardCompatibleWith[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-incompatibleWith> *)
owl:incompatibleWith[rdf:type->owl:AnnotationProperty]

(* <#prp-dom> *)
Forall ?p ?c ?x ?y (
  ?x[rdf:type->?c] :- And(
    ?p[rdfs:domain->?c]
    ?x[?p->?y] ))

(* <#prp-rng> *)
Forall ?p ?c ?x ?y (
  ?y[rdf:type->?c] :- And(
    ?p[rdfs:range->?c]
    ?x[?p->?y] ))

(* <#prp-fp> *)
Forall ?p ?y2 ?x ?y1 (
  ?y1[owl:sameAs->?y2] :- And(
    ?p[rdf:type->owl:FunctionalProperty]
    ?x[?p->?y1]
    ?x[?p->?y2] ))

(* <#prp-ifp> *)
Forall ?p ?x1 ?x2 ?y (
  ?x1[owl:sameAs->?x2] :- And(
    ?p[rdf:type->owl:InverseFunctionalProperty]
    ?x1[?p->?y]
    ?x2[?p->?y] ))

(* <#prp-irp> *)
Forall ?p ?x (
  rif:error() :- And(
    ?p[rdf:type->owl:IrreflexiveProperty]
    ?x[?p->?x] ))

(* <#prp-symp> *)
Forall ?p ?x ?y (
  ?y[?p->?x] :- And(
    ?p[rdf:type->owl:SymmetricProperty]
    ?x[?p->?y] ))

(* <#prp-asymp> *)
Forall ?p ?x ?y (
  rif:error() :- And(
    ?p[rdf:type->owl:AsymmetricProperty]
    ?x[?p->?y]
    ?y[?p->?x] ))

(* <#prp-trp> *)
Forall ?p ?x ?z ?y (
  ?x[?p->?z] :- And(
    ?p[rdf:type->owl:TransitiveProperty]
    ?x[?p->?y]
    ?y[?p->?z] ))

```

```

(* <#prp-spo1> *)
Forall ?x ?y ?p2 ?p1 (
  ?x[?p2->?y] :- And(
    ?p1[rdfs:subPropertyOf->?p2]
    ?x[?p1->?y] ))

(* <#prp-eqp1> *)
Forall ?x ?y ?p2 ?p1 (
  ?x[?p2->?y] :- And(
    ?p1[owl:equivalentProperty->?p2]
    ?x[?p1->?y] ))

(* <#prp-eqp2> *)
Forall ?x ?y ?p2 ?p1 (
  ?x[?p1->?y] :- And(
    ?p1[owl:equivalentProperty->?p2]
    ?x[?p2->?y] ))

(* <#prp-pdw> *)
Forall ?x ?y ?p2 ?p1 (
  rif:error() :- And(
    ?p1[owl:propertyDisjointWith->?p2]
    ?x[?p1->?y]
    ?x[?p2->?y] ))

(* <#prp-inv1> *)
Forall ?x ?y ?p2 ?p1 (
  ?y[?p2->?x] :- And(
    ?p1[owl:inverseOf->?p2]
    ?x[?p1->?y] ))

(* <#prp-inv2> *)
Forall ?x ?y ?p2 ?p1 (
  ?y[?p1->?x] :- And(
    ?p1[owl:inverseOf->?p2]
    ?x[?p2->?y] ))

(* <#cls-thing> *)
owl:Thing[rdf:type->owl:Class]

(* <#cls-nothing1> *)
owl:Nothing[rdf:type->owl:Class]

(* <#cls-nothing2> *)
Forall ?x (
  rif:error() :- ?x[rdf:type->owl:Nothing])

(* <#cls-svf1> *)
Forall ?p ?v ?u ?x ?y (
  ?u[rdf:type->?x] :- And(
    ?x[owl:someValuesFrom->?y]
    ?x[owl:onProperty->?p]
    ?u[?p->?v]
    ?v[rdf:type->?y] ))

(* <#cls-svf2> *)
Forall ?p ?v ?u ?x (
  ?u[rdf:type->?x] :- And(
    ?x[owl:someValuesFrom->owl:Thing]
    ?x[owl:onProperty->?p]
    ?u[?p->?v] ))

(* <#cls-avf> *)
Forall ?p ?v ?u ?x ?y (
  ?v[rdf:type->?y] :- And(
    ?x[owl:allValuesFrom->?y]
    ?x[owl:onProperty->?p]
    ?u[rdf:type->?x]
    ?u[?p->?v] ))

(* <#cls-hv1> *)
Forall ?p ?u ?x ?y (
  ?u[?p->?y] :- And(
    ?x[owl:hasValue->?y]
    ?x[owl:onProperty->?p]
    ?u[rdf:type->?x] ))

(* <#cls-hv2> *)
Forall ?p ?u ?x ?y (
  ?u[rdf:type->?x] :- And(
    ?x[owl:hasValue->?y]
    ?x[owl:onProperty->?p]
    ?u[?p->?y] ))

(* <#cls-maxc1> *)
Forall ?p ?u ?x ?y (
  rif:error() :- And(
    ?x[owl:maxCardinality->0]
    ?x[owl:onProperty->?p]
    ?u[rdf:type->?x]
    ?u[?p->?y] ))

(* <#cls-maxc2> *)
Forall ?p ?y2 ?u ?x ?y1 (
  ?y1[owl:sameAs->?y2] :- And(
    ?x[owl:maxCardinality->1]
    ?x[owl:onProperty->?p]
    ?u[rdf:type->?x]
  )

```

```

    ?u[?p->?y1]
    ?u[?p->?y2] ))

(* <#cls-maxqc1> *)
Forall ?p ?c ?u ?x ?y (
  rif:error() :- And(
    ?x[owl:maxQualifiedCardinality->0]
    ?x[owl:onProperty->?p]
    ?x[owl:onClass->?c]
    ?u[rdftype->?x]
    ?u[?p->?y]
    ?y[rdftype->?c] ))

(* <#cls-maxqc2> *)
Forall ?p ?u ?x ?y (
  rif:error() :- And(
    ?x[owl:maxQualifiedCardinality->0]
    ?x[owl:onProperty->?p]
    ?x[owl:onClass->owl:Thing]
    ?u[rdftype->?x]
    ?u[?p->?y] ))

(* <#cls-maxqc3> *)
Forall ?p ?y2 ?c ?u ?x ?y1 (
  ?y1[owl:sameAs->?y2] :- And(
    ?x[owl:maxQualifiedCardinality->1]
    ?x[owl:onProperty->?p]
    ?x[owl:onClass->?c]
    ?u[rdftype->?x]
    ?u[?p->?y1]
    ?y1[rdftype->?c]
    ?u[?p->?y2]
    ?y2[rdftype->?c] ))

(* <#cls-maxqc4> *)
Forall ?p ?y2 ?u ?x ?y1 (
  ?y1[owl:sameAs->?y2] :- And(
    ?x[owl:maxQualifiedCardinality->1]
    ?x[owl:onProperty->?p]
    ?x[owl:onClass->owl:Thing]
    ?u[rdftype->?x]
    ?u[?p->?y1]
    ?u[?p->?y2] ))

(* <#cax-sco> *)
Forall ?x ?c1 ?c2 (
  ?x[rdftype->?c2] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?x[rdftype->?c1] ))

(* <#cax-ecq1> *)
Forall ?x ?c1 ?c2 (
  ?x[rdftype->?c2] :- And(
    ?c1[owl:equivalentClass->?c2]
    ?x[rdftype->?c1] ))

(* <#cax-ecq2> *)
Forall ?x ?c1 ?c2 (
  ?x[rdftype->?c1] :- And(
    ?c1[owl:equivalentClass->?c2]
    ?x[rdftype->?c2] ))

(* <#cax-dw> *)
Forall ?x ?c1 ?c2 (
  rif:error() :- And(
    ?c1[owl:disjointWith->?c2]
    ?x[rdftype->?c1]
    ?x[rdftype->?c2] ))

(* <#scm-cls> *)
Forall ?c (
  ?c[rdfs:subClassOf->?c] :- ?c[rdftype->owl:Class])

(* <#scm-cls1> *)
Forall ?c (
  ?c[owl:equivalentClass->?c] :- ?c[rdftype->owl:Class])

(* <#scm-cls2> *)
Forall ?c (
  ?c[rdfs:subClassOf->owl:Thing] :- ?c[rdftype->owl:Class])

(* <#scm-cls3> *)
Forall ?c (
  owl:Nothing[rdfs:subClassOf->?c] :- ?c[rdftype->owl:Class])

(* <#scm-sco> *)
Forall ?c1 ?c2 ?c3 (
  ?c1[rdfs:subClassOf->?c3] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?c2[rdfs:subClassOf->?c3] ))

(* <#scm-ecq1> *)
Forall ?c1 ?c2 (
  ?c1[rdfs:subClassOf->?c2] :- ?c1[owl:equivalentClass->?c2])

(* <#scm-ecq11> *)
Forall ?c1 ?c2 (
  ?c2[rdfs:subClassOf->?c1] :- ?c1[owl:equivalentClass->?c2])

```

```

(* <#scm-eqc2> *)
Forall ?c1 ?c2 (
  ?c1[owl:equivalentClass->?c2] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?c2[rdfs:subClassOf->?c1] ) )

(* <#scm-op> *)
Forall ?p (
  ?p[rdfs:subPropertyOf->?p] :- ?p[rdftype->owl:ObjectProperty])

(* <#scm-op1> *)
Forall ?p (
  ?p[owl:equivalentProperty->?p] :- ?p[rdftype->owl:ObjectProperty])

(* <#scm-dp> *)
Forall ?p (
  ?p[rdfs:subPropertyOf->?p] :- ?p[rdftype->owl:DatatypeProperty])

(* <#scm-dp1> *)
Forall ?p (
  ?p[owl:equivalentProperty->?p] :- ?p[rdftype->owl:DatatypeProperty])

(* <#scm-spo> *)
Forall ?p3 ?p2 ?p1 (
  ?p1[rdfs:subPropertyOf->?p3] :- And(
    ?p1[rdfs:subPropertyOf->?p2]
    ?p2[rdfs:subPropertyOf->?p3] ) )

(* <#scm-eqp1> *)
Forall ?p2 ?p1 (
  ?p1[rdfs:subPropertyOf->?p2] :- ?p1[owl:equivalentProperty->?p2])

(* <#scm-eqp11> *)
Forall ?p2 ?p1 (
  ?p2[rdfs:subPropertyOf->?p1] :- ?p1[owl:equivalentProperty->?p2])

(* <#scm-eqp2> *)
Forall ?p2 ?p1 (
  ?p1[owl:equivalentProperty->?p2] :- And(
    ?p1[rdfs:subPropertyOf->?p2]
    ?p2[rdfs:subPropertyOf->?p1] ) )

(* <#scm-dom1> *)
Forall ?p ?c1 ?c2 (
  ?p[rdfs:domain->?c2] :- And(
    ?p[rdfs:domain->?c1]
    ?c1[rdfs:subClassOf->?c2] ) )

(* <#scm-dom2> *)
Forall ?c ?p2 ?p1 (
  ?p1[rdfs:domain->?c] :- And(
    ?p2[rdfs:domain->?c]
    ?p1[rdfs:subPropertyOf->?p2] ) )

(* <#scm-rng1> *)
Forall ?p ?c1 ?c2 (
  ?p[rdfs:range->?c2] :- And(
    ?p[rdfs:range->?c1]
    ?c1[rdfs:subClassOf->?c2] ) )

(* <#scm-rng2> *)
Forall ?c ?p2 ?p1 (
  ?p1[rdfs:range->?c] :- And(
    ?p2[rdfs:range->?c]
    ?p1[rdfs:subPropertyOf->?p2] ) )

(* <#scm-hv> *)
Forall ?c1 ?c2 ?i ?p2 ?p1 (
  ?c1[rdfs:subClassOf->?c2] :- And(
    ?c1[owl:hasValue->?i]
    ?c1[owl:onProperty->?p1]
    ?c2[owl:hasValue->?i]
    ?c2[owl:onProperty->?p2]
    ?p1[rdfs:subPropertyOf->?p2] ) )

(* <#scm-svf1> *)
Forall ?p ?y2 ?c1 ?c2 ?y1 (
  ?c1[rdfs:subClassOf->?c2] :- And(
    ?c1[owl:someValuesFrom->?y1]
    ?c1[owl:onProperty->?p]
    ?c2[owl:someValuesFrom->?y2]
    ?c2[owl:onProperty->?p]
    ?y1[rdfs:subClassOf->?y2] ) )

(* <#scm-svf2> *)
Forall ?c1 ?c2 ?y ?p2 ?p1 (
  ?c1[rdfs:subClassOf->?c2] :- And(
    ?c1[owl:someValuesFrom->?y]
    ?c1[owl:onProperty->?p1]
    ?c2[owl:someValuesFrom->?y]
    ?c2[owl:onProperty->?p2]
    ?p1[rdfs:subPropertyOf->?p2] ) )

(* <#scm-avf1> *)
Forall ?p ?y2 ?c1 ?c2 ?y1 (
  ?c1[rdfs:subClassOf->?c2] :- And(
    ?c1[owl:allValuesFrom->?y1]
    ?c1[owl:onProperty->?p]
    ?c2[owl:allValuesFrom->?y2]
  ) )

```

```

?c2[owl:onProperty->?p]
?y1[rdfs:subClassOf->?y2] ))

(* <#scm-avf2> *)
Forall ?c1 ?c2 ?y ?p2 ?p1 (
  ?c2[rdfs:subClassOf->?c1] :- And(
    ?c1[owl:allValuesFrom->?y]
    ?c1[owl:onProperty->?p1]
    ?c2[owl:allValuesFrom->?y]
    ?c2[owl:onProperty->?p2]
    ?p1[rdfs:subPropertyOf->?p2] ))

(* <#prp-npa1> *)
Forall ?x ?i1 ?p ?i2 (
  rif:error() :- And(
    ?x[owl:sourceIndividual->?i1]
    ?x[owl:assertionProperty->?p]
    ?x[owl:targetIndividual->?i2]
    ?i1[?p->?i2] ))

(* <#prp-npa2> *)
Forall ?x ?i ?p ?lt (
  rif:error() :- And(
    ?x[owl:sourceIndividual->?i]
    ?x[owl:assertionProperty->?p]
    ?x[owl:targetValue->?lt]
    ?i[?p->?lt] ))

(* <#cax-dw> *)
Forall ?c1 ?c2 ?x (
  rif:error() :- And(
    ?c1[owl:disjointWith->?c2]
    ?x[rdf:type->?c1]
    ?x[rdf:type->?c2] ))

(* <#cls-com> *)
Forall ?c1 ?c2 ?x (
  rif:error() :- And(
    ?c1[owl:complementOf->?c2]
    ?x[rdf:type->?c1]
    ?x[rdf:type->?c2] ))

))

```

List rules

```

Document (
  Prefix(rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(owl <http://www.w3.org/2002/07/owl#>)
  Prefix(xsd <http://www.w3.org/2001/XMLSchema#>)
  Prefix(rif <http://www.w3.org/2007/rif#>)
  Prefix(func <http://www.w3.org/2007/rif-builtin-function#>)
  Prefix(pred <http://www.w3.org/2007/rif-builtin-predicate#>)
  Prefix(dc <http://purl.org/dc/terms/>)
  Group (

    (* <#eq-diff2> *)
    Forall ?x ?y ?z1 ?z2 ?iz1 ?iz2 (
      rif:error() :- And (
        ?x[rdf:type -> owl:AllDifferent]
        ?x[owl:members -> ?y]
        External(pred:list-contains(?y ?z1)) ?iz1 = External(func:index-of(?y ?z1))
        External(pred:list-contains(?y ?z2)) ?iz2 = External(func:index-of(?y ?z2))
        External( pred:numeric-not-equal ( ?iz1 ?iz2 ) )
        ?z1[owl:sameAs->?z2] ) )

    (* <#eq-diff3> *)
    Forall ?x ?y ?z1 ?z2 ?iz1 ?iz2 (
      rif:error() :- And (
        ?x[rdf:type -> owl:AllDifferent]
        ?x[owl:distinctMembers -> ?y]
        External(pred:list-contains(?y ?z1)) ?iz1 = External(func:index-of(?y ?z1))
        External(pred:list-contains(?y ?z2)) ?iz2 = External(func:index-of(?y ?z2))
        External( pred:numeric-not-equal ( ?iz1 ?iz2 ) )
        ?z1[owl:sameAs->?z2] ) )

    (* <#prp-adp> *)
    Forall ?x ?y ?o ?v ?l ?ix ?iy ?r (
      rif:error() :- And (
        ?r[rdf:type -> owl:AllDisjointProperties]
        ?r[owl:members -> ?l]
        External(pred:list-contains(?l ?x)) ?ix = External(func:index-of(?l ?x))
        External(pred:list-contains(?l ?y)) ?iy = External(func:index-of(?l ?y))
        External( pred:numeric-not-equal ( ?ix ?iy ) )
        ?o[?x->?v ?y->?v] ) )

    (* <#cax-adc> *)
    Forall ?x ?y ?o ?l ?ix ?iy ?r (
      rif:error() :- And (
        ?r[rdf:type -> owl:AllDisjointClasses]
        ?r[owl:members -> ?l]
        External(pred:list-contains(?l ?x)) ?ix = External(func:index-of(?l ?x))
        External(pred:list-contains(?l ?y)) ?iy = External(func:index-of(?l ?y))
        External( pred:numeric-not-equal ( ?ix ?iy ) )
        ?o[rdf:type->?x rdf:type->?y] ) )

    (* <#prp-spo2> *)
    Forall ?p ?last ?pc ?start (

```

```

?start[?p->?last] :- And (
  ?p[owl:propertyChainAxiom->?pc]
  _checkChain(?start ?pc ?last) )

Forall ?start ?pc ?last ?p ?tl (
  _checkChain(?start ?pc ?last) :- And (
    ?pc[rd:first->?p rd:rest->?tl]
    ?start[?p->?next]
    _checkChain(?next ?tl ?last) ) )

Forall ?start ?pc ?last ?p (
  _checkChain(?start ?pc ?last) :- And (
    ?pc[rd:first->?p rd:rest->rdf:nil]
    ?start[?p->?last] ) )

Forall ?x ?y ?c ?u ?c (
  ?x[owl:sameAs->?y] :- And (
    ?c[owl:hasKey->?u] ?x[rd:type->?c] ?y[rd:type->?c]
    _sameKey(?u ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rd:first->?key rd:rest->?tl]
    ?x[?key->?v] ?y[?key->?v]
    _sameKey(?tl ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rd:first->?key rd:rest->rdf:nil]
    ?x[?key->?v] ?y[?key->?v] ) )

(* <#cls-int1> *)
Forall ?y ?c ?l (
  ?y[rd:type->?c] :- And (
    ?c[owl:intersectionOf->?l]
    _allTypes(?l ?y) ) )

Forall ?l ?y ?ty ?tl (
  _allTypes(?l ?y) :- And (
    ?l[rd:first->?ty rd:rest->?tl]
    ?y[rd:type->?ty]
    _allTypes(?tl ?y) ) )

Forall ?l ?y ?ty (
  _allTypes(?l ?y) :- And (
    ?l[rd:first->?ty rd:rest->rdf:nil]
    ?y[rd:type->?ty] ) )

(* <#prp-key> *)
Forall ?x ?y ?c ?u (
  ?x[owl:sameAs->?y] :- And (
    ?c[owl:hasKey->?u] ?x[rd:type->?c] ?y[rd:type->?c]
    _sameKey(?u ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rd:first->?key rd:rest->?tl]
    ?x[?key->?v] ?y[?key->?v]
    _sameKey(?tl ?x ?y) ) )

Forall ?u ?x ?y (
  _sameKey(?u ?x ?y) :- And (
    ?u[rd:first->?key rd:rest->rdf:nil]
    ?x[?key->?v] ?y[?key->?v] ) )

(* <#cls-uni> *)
Forall ?y ?c ?l ?ci (
  ?y[rd:type->?c] :- And (
    ?c[owl:unionOf->?l]
    External(pred:list-contains(?l ?ci))
    ?y[rd:type->?ci] ) )

(* <#cls-oo> *)
Forall ?yi ?c ?l (
  ?yi[rd:type->?c] :- And (
    ?c[owl:oneOf->?l]
    External(pred:list-contains(?l ?yi)) ) )

(* <#cls-int2> *)
Forall ?y ?c ?ci ?l (
  ?y[rd:type->?ci] :- And (
    ?c[owl:intersectionOf->?l]
    External(pred:list-contains(?l ?ci))
    ?y[rd:type->?c] ) )

(* <#scm-int> *)
Forall ?c ?ci ?l (
  ?c[rd:subClassOf->?ci] :- And (
    ?c[owl:intersectionOf->?l]
    External(pred:list-contains(?l ?ci)) ) )

(* <#scm-uni> *)
Forall ?c ?ci ?l (
  ?ci[rd:subClassOf->?c] :- And (
    ?c[owl:unionOf->?l]
    External(pred:list-contains(?l ?ci)) ) )

```

Datatype rules

```

Document (
Prefix(rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
Prefix(owl <http://www.w3.org/2002/07/owl#>)
Prefix(xsd <http://www.w3.org/2001/XMLSchema#>)
Prefix(rif <http://www.w3.org/2007/rif#>)
Prefix(func <http://www.w3.org/2007/rif-builtin-function#>)
Prefix(pred <http://www.w3.org/2007/rif-builtin-predicate#>)
Prefix(dc <http://purl.org/dc/terms/>)
Group (
(* <#dt-type2> *)
Group (
  Forall ?s ?p ?lt ( ?lt[rdf:type->rdf:PlainLiteral rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-PlainLiteral( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->rdf:XMLLiteral rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-XMLLiteral( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:decimal rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-decimal( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:integer rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-integer( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonNegativeInteger rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-nonNegativeInteger( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonPositiveInteger rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-nonPositiveInteger( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:positiveInteger rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-positiveInteger( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:negativeInteger rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-negativeInteger( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:long rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-long( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:int rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-int( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:short rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-short( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:byte rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-byte( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedLong rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-unsignedLong( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedInt rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-unsignedInt( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedShort rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-unsignedShort( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedByte rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-unsignedByte( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:float rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-float( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:double rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-double( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:string rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-string( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:normalizedString rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-normalizedString( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:token rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-token( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:language rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-language( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:Name rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-Name( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NCName rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-NCName( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NMTOKEN rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-NMTOKEN( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:boolean rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-boolean( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:hexBinary rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-hexBinary( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:base64Binary rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-base64Binary( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:anyURI rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-anyURI( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTime rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-dateTime( ?lt ) ) )
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTimeStamp rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-dateTimeStamp ( ?lt ) ) )
)
)
(* <#dt-not-type> *)
Group (
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->rdf:PlainLiteral] External(pred:is-literal-not-PlainLiteral( ?lt ) ) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral( ?lt ) ) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:decimal] External(pred:is-literal-not-decimal( ?lt ) ) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:integer] External(pred:is-literal-not-integer( ?lt ) ) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:nonNegativeInteger] External(pred:is-literal-not-nonNegativeInteger( ?lt ) ) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:nonPositiveInteger] External(pred:is-literal-not-nonPositiveInteger( ?lt ) ) )
  Forall ?lt (

```



```

rif:error() :- And (
  ?lt[rdftype->xsd:positiveInteger] External(pred:is-literal-not-positiveInteger( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:negativeInteger] External(pred:is-literal-not-negativeInteger( ?lt )) )
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdftype->xsd:long] External(pred:is-literal-not-long( ?lt )) )
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdftype->xsd:int] External(pred:is-literal-not-int( ?lt )) )
      Forall ?lt (
        rif:error() :- And (
          ?lt[rdftype->xsd:short] External(pred:is-literal-not-short( ?lt )) )
        Forall ?lt (
          rif:error() :- And (
            ?lt[rdftype->xsd:byte] External(pred:is-literal-not-byte( ?lt )) )
          Forall ?lt (
            rif:error() :- And (
              ?lt[rdftype->xsd:unsignedLong] External(pred:is-literal-not-unsignedLong( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:unsignedInt] External(pred:is-literal-not-unsignedInt( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:unsignedShort] External(pred:is-literal-not-unsignedShort( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:unsignedByte] External(pred:is-literal-not-unsignedByte( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:float] External(pred:is-literal-not-float( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:double] External(pred:is-literal-not-double( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:string] External(pred:is-literal-not-string( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:normalizedString] External(pred:is-literal-not-normalizedString( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:token] External(pred:is-literal-not-token( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:language] External(pred:is-literal-not-language( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:name] External(pred:is-literal-not-name( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:ncName] External(pred:is-literal-not-ncName( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:nmtoken] External(pred:is-literal-not-nmtoken( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:boolean] External(pred:is-literal-not-boolean( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:hexBinary] External(pred:is-literal-not-hexBinary( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:base64Binary] External(pred:is-literal-not-base64Binary( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:anyURI] External(pred:is-literal-not-anyURI( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:dateTime] External(pred:is-literal-not-dateTime( ?lt )) )
            Forall ?lt (
              rif:error() :- And (
                ?lt[rdftype->xsd:dateTimeStamp] External(pred:is-literal-not-dateTimeStamp ( ?lt )) )
          )
        )
      )
    )
  )
)

(* <#eq-diff1-literal1> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x[owl:sameAs->?y]
    External(pred:literal-not-identical(?x ?y)) )

(* <#eq-diff1-literal2> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x = ?y
    ?x[owl:differentFrom->?y] )

(* <#dt-type1-PlainLiteral> *) rdf:PlainLiteral[rdftype -> rdfs:Datatype]
(* <#dt-type1-decimal> *) xsd:decimal[rdftype -> rdfs:Datatype]
(* <#dt-type1-integer> *) xsd:integer[rdftype -> rdfs:Datatype]
(* <#dt-type1-double> *) xsd:double[rdftype -> rdfs:Datatype]
(* <#dt-type1-string> *) xsd:string[rdftype -> rdfs:Datatype]
(* <#dt-type1-dateTime> *) xsd:dateTime[rdftype -> rdfs:Datatype]
(* <#dt-type1-XMLLiteral> *) rdf:XMLLiteral[rdftype -> rdfs:Datatype]
(* <#dt-type1-Literal> *) rdfs:Literal[rdftype -> rdfs:Datatype]

```

```

(* <#dt-type1-nonNegativeInteger> *) xsd:nonNegativeInteger[rdf:type -> rdfs:Datatype]
(* <#dt-type1-nonPositiveInteger> *) xsd:nonPositiveInteger[rdf:type -> rdfs:Datatype]
(* <#dt-type1-positiveInteger> *) xsd:positiveInteger[rdf:type -> rdfs:Datatype]
(* <#dt-type1-negativeInteger> *) xsd:negativeInteger[rdf:type -> rdfs:Datatype]
(* <#dt-type1-long> *) xsd:long[rdf:type -> rdfs:Datatype]
(* <#dt-type1-int> *) xsd:int[rdf:type -> rdfs:Datatype]
(* <#dt-type1-short> *) xsd:short[rdf:type -> rdfs:Datatype]
(* <#dt-type1-byte> *) xsd:byte[rdf:type -> rdfs:Datatype]
(* <#dt-type1-unsignedLong> *) xsd:unsignedLong[rdf:type -> rdfs:Datatype]
(* <#dt-type1-unsignedInt> *) xsd:unsignedInt[rdf:type -> rdfs:Datatype]
(* <#dt-type1-unsignedShort> *) xsd:unsignedShort[rdf:type -> rdfs:Datatype]
(* <#dt-type1-unsignedByte> *) xsd:unsignedByte[rdf:type -> rdfs:Datatype]
(* <#dt-type1-normalizedString> *) xsd:normalizedString[rdf:type -> rdfs:Datatype]
(* <#dt-type1-token> *) xsd:token[rdf:type -> rdfs:Datatype]
(* <#dt-type1-language> *) xsd:language[rdf:type -> rdfs:Datatype]
(* <#dt-type1-Name> *) xsd:Name[rdf:type -> rdfs:Datatype]
(* <#dt-type1-NCName> *) xsd:NCName[rdf:type -> rdfs:Datatype]
(* <#dt-type1-NMTOKEN> *) xsd:NMTOKEN[rdf:type -> rdfs:Datatype]

(* <#dt-type1-float> *) xsd:float[rdf:type -> rdfs:Datatype]
(* <#dt-type1-boolean> *) xsd:boolean[rdf:type -> rdfs:Datatype]
(* <#dt-type1-hexBinary> *) xsd:hexBinary[rdf:type -> rdfs:Datatype]
(* <#dt-type1-base64Binary> *) xsd:base64Binary[rdf:type -> rdfs:Datatype]
(* <#dt-type1-anyURI> *) xsd:anyURI[rdf:type -> rdfs:Datatype]
(* <#dt-type1-dateTimeStamp> *) xsd:dateTimeStamp [rdf:type -> rdfs:Datatype]

))

```

8 Appendix: OWL 2 RL to RIF translation

The static set of rules in the first appendix provides a complete translation of the OWL 2 RL rules into RIF. While that rule set is within the RIF Core dialect it is fairly inefficient, for example in its handling of lists rules.

In practice we would expect many OWL 2 RL implementations to instantiate the ruleset for a particular ontology. The instantiation process only depends upon OWL TBox axioms and the instantiated ruleset can be applied to other ontologies which only differ by virtue of the ABox assertions.

We here define an algorithm for instantiating a RIF Core rule set for a given OWL 2 RL ontology.

Input: An ontology O conforming to the OWL 2 RL profile and the corresponding translation of O into an RDF Graph $RDF(O)$ as specified in the OWL 2 Mapping to RDF Graphs [OWL 2 RDF Mapping].

Output: A RIF Core rule set $R(RDF(O))$ such that the RIF-RDF combination of $R(RDF(O))$ and $RDF(O)$ has the same entailments as the combination R and $RDF(O)$ where R is the static OWL 2 RL rule set defined above.

Algorithm: $R(RDF(O)) = \text{FixedRules} \cup \text{templateRules}(RDF(O))$

Where the set of FixedRules and the two algorithms templateRules are defined below.

Editor's Note: These rules are believed to be correct. However, since the original version was developed various manual edits have been made to conform to changes in RIF and OWL 2 RL. Mechanical verification will be required before final publication.

8.1 FixedRules

The *FixedRule* ruleset comprises the following rules:

```

(* <#eq-ref> *)
Forall ?p ?o ?s (
  ?s[owl:sameAs->?s] :- ?s[?p->?o])

(* <#eq-ref1> *)
Forall ?p ?o ?s (
  ?p[owl:sameAs->?p] :- ?s[?p->?o])

(* <#eq-ref2> *)
Forall ?p ?o ?s (
  ?o[owl:sameAs->?o] :- ?s[?p->?o])

(* <#eq-sym> *)
Forall ?x ?y (
  ?y[owl:sameAs->?x] :- ?x[owl:sameAs->?y])

(* <#eq-trans> *)
Forall ?x ?z ?y (
  ?x[owl:sameAs->?z] :- And(
    ?x[owl:sameAs->?y]
    ?y[owl:sameAs->?z] ))

(* <#eq-rep-s> *)
Forall ?p ?o ?s ?s2 (
  ?s2[?p->?o] :- And(
    ?s[owl:sameAs->?s2]
    ?s[?p->?o] ))

(* <#eq-rep-p> *)
Forall ?p ?o ?s ?p2 (
  ?s[?p2->?o] :- And(
    ?p[owl:sameAs->?p2]
    ?s[?p->?o] ))

(* <#eq-rep-o> *)
Forall ?p ?o ?s ?o2 (
  ?s[?p->?o2] :- And(
    ?o[owl:sameAs->?o2]
    ?s[?p->?o] ))

```

```

(* <#eq-diff1> *)
Forall ?x ?y (
  rif:error() :- And(
    ?x[owl:sameAs->?y]
    ?x[owl:differentFrom->?y] ))

(* <#prp-ap-label> *)
rdfs:label[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-comment> *)
rdfs:comment[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-seeAlso> *)
rdfs:seeAlso[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-isDefinedBy> *)
rdfs:isDefinedBy[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-deprecated> *)
owl:deprecated[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-priorVersion> *)
owl:priorVersion[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-backwardCompatibleWith> *)
owl:backwardCompatibleWith[rdf:type->owl:AnnotationProperty]

(* <#prp-ap-incompatibleWith> *)
owl:incompatibleWith[rdf:type->owl:AnnotationProperty]

(* <#prp-dom> *)
Forall ?p ?c ?x ?y (
  ?x[rdf:type->?c] :- And(
    ?p[rdfs:domain->?c]
    ?x[?p->?y] ))

(* <#prp-rng> *)
Forall ?p ?c ?x ?y (
  ?y[rdf:type->?c] :- And(
    ?p[rdfs:range->?c]
    ?x[?p->?y] ))

(* <#cls-thing> *)
owl:Thing[rdf:type->owl:Class]

(* <#cls-nothing1> *)
owl:Nothing[rdf:type->owl:Class]

(* <#cls-nothing2> *)
Forall ?x (
  rif:error() :- ?x[rdf:type->owl:Nothing])

(* <#cax-sco> *)
Forall ?x ?c1 ?c2 (
  ?x[rdf:type->?c2] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?x[rdf:type->?c1] ))

(* <#cax-ec1> *)
Forall ?x ?c1 ?c2 (
  ?x[rdf:type->?c2] :- And(
    ?c1[owl:equivalentClass->?c2]
    ?x[rdf:type->?c1] ))

(* <#cax-ec2> *)
Forall ?x ?c1 ?c2 (
  ?x[rdf:type->?c1] :- And(
    ?c1[owl:equivalentClass->?c2]
    ?x[rdf:type->?c2] ))

(* <#cax-dw> *)
Forall ?x ?c1 ?c2 (
  rif:error() :- And(
    ?c1[owl:disjointWith->?c2]
    ?x[rdf:type->?c1]
    ?x[rdf:type->?c2] ))

(* <#scm-cls> *)
Forall ?c (
  ?c[rdfs:subClassOf->?c] :- ?c[rdf:type->owl:Class])

(* <#scm-cls1> *)
Forall ?c (
  ?c[owl:equivalentClass->?c] :- ?c[rdf:type->owl:Class])

(* <#scm-cls2> *)
Forall ?c (
  ?c[rdfs:subClassOf->owl:Thing] :- ?c[rdf:type->owl:Class])

(* <#scm-cls3> *)
Forall ?c (
  owl:Nothing[rdfs:subClassOf->?c] :- ?c[rdf:type->owl:Class])

(* <#scm-sco> *)
Forall ?c1 ?c2 ?c3 (
  ?c1[rdfs:subClassOf->?c3] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?c2[rdfs:subClassOf->?c3] ))

```

```

(* <#scm-eqcl> *)
Forall ?c1 ?c2 (
  ?c1[rdfs:subClassOf->?c2] :- ?c1[owl:equivalentClass->?c2])

(* <#scm-eqcl1> *)
Forall ?c1 ?c2 (
  ?c2[rdfs:subClassOf->?c1] :- ?c1[owl:equivalentClass->?c2])

(* <#scm-eqc2> *)
Forall ?c1 ?c2 (
  ?c1[owl:equivalentClass->?c2] :- And(
    ?c1[rdfs:subClassOf->?c2]
    ?c2[rdfs:subClassOf->?c1] ))

(* <#scm-op> *)
Forall ?p (
  ?p[rdfs:subPropertyOf->?p] :- ?p[rdftype->owl:ObjectProperty])

(* <#scm-op1> *)
Forall ?p (
  ?p[owl:equivalentProperty->?p] :- ?p[rdftype->owl:ObjectProperty])

(* <#scm-dp> *)
Forall ?p (
  ?p[rdfs:subPropertyOf->?p] :- ?p[rdftype->owl:DatatypeProperty])

(* <#scm-dp1> *)
Forall ?p (
  ?p[owl:equivalentProperty->?p] :- ?p[rdftype->owl:DatatypeProperty])

(* <#scm-spo> *)
Forall ?p3 ?p2 ?p1 (
  ?p1[rdfs:subPropertyOf->?p3] :- And(
    ?p1[rdfs:subPropertyOf->?p2]
    ?p2[rdfs:subPropertyOf->?p3] ))

(* <#scm-eqp1> *)
Forall ?p2 ?p1 (
  ?p1[rdfs:subPropertyOf->?p2] :- ?p1[owl:equivalentProperty->?p2])

(* <#scm-eqp11> *)
Forall ?p2 ?p1 (
  ?p2[rdfs:subPropertyOf->?p1] :- ?p1[owl:equivalentProperty->?p2])

(* <#scm-eqp2> *)
Forall ?p2 ?p1 (
  ?p1[owl:equivalentProperty->?p2] :- And(
    ?p1[rdfs:subPropertyOf->?p2]
    ?p2[rdfs:subPropertyOf->?p1] ))

(* <#scm-dom1> *)
Forall ?p ?c1 ?c2 (
  ?p[rdfs:domain->?c2] :- And(
    ?p[rdfs:domain->?c1]
    ?c1[rdfs:subClassOf->?c2] ))

(* <#scm-dom2> *)
Forall ?c ?p2 ?p1 (
  ?p1[rdfs:domain->?c] :- And(
    ?p2[rdfs:domain->?c]
    ?p1[rdfs:subPropertyOf->?p2] ))

(* <#scm-rng1> *)
Forall ?p ?c1 ?c2 (
  ?p[rdfs:range->?c2] :- And(
    ?p[rdfs:range->?c1]
    ?c1[rdfs:subClassOf->?c2] ))

(* <#scm-rng2> *)
Forall ?c ?p2 ?p1 (
  ?p1[rdfs:range->?c] :- And(
    ?p2[rdfs:range->?c]
    ?p1[rdfs:subPropertyOf->?p2] ))

(* <#dt-type2> *)
Group (
  Forall ?s ?p ?lt ( ?lt[rdftype->rdf:PlainLiteral rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-PlainLiteral( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->rdf:XMLLiteral rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-XMLLiteral( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:decimal rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-decimal( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:integer rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-integer( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:nonNegativeInteger rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-nonNegativeInteger( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:nonPositiveInteger rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-nonPositiveInteger( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:positiveInteger rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-positiveInteger( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:negativeInteger rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-negativeInteger( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:long rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-long( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:int rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-int( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:short rdftype->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-short( ?lt ) ) ) )
  Forall ?s ?p ?lt ( ?lt[rdftype->xsd:byte rdftype->rdfs:Literal]

```

```

)
Forall ?s ?p ?lt ( :- And( ?s[?p->?lt] External( pred:is-literal-byte( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedLong rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-unsignedLong( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedInt rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-unsignedInt( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedShort rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-unsignedShort( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedByte rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-unsignedByte( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:float rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-float( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:double rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-double( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:string rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-string( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:normalizedString rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-normalizedString( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:token rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-token( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:language rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-language( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:Name rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-Name( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NCName rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NCName( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NMTOKEN rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NMTOKEN( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:boolean rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-boolean( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:hexBinary rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-hexBinary( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:base64Binary rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-base64Binary( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:anyURI rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-anyURI( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTime rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-dateTime( ?lt ) ) )
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTimeStamp rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-dateTimeStamp ( ?lt ) ) )
)

(* <#dt-not-type> *)
Group (
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->rdf:PlainLiteral] External(pred:is-literal-not-PlainLiteral( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:decimal] External(pred:is-literal-not-decimal( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:integer] External(pred:is-literal-not-integer( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:nonNegativeInteger] External(pred:is-literal-not-nonNegativeInteger( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:nonPositiveInteger] External(pred:is-literal-not-nonPositiveInteger( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:positiveInteger] External(pred:is-literal-not-positiveInteger( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:negativeInteger] External(pred:is-literal-not-negativeInteger( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:long] External(pred:is-literal-not-long( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:int] External(pred:is-literal-not-int( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:short] External(pred:is-literal-not-short( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:byte] External(pred:is-literal-not-byte( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:unsignedLong] External(pred:is-literal-not-unsignedLong( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:unsignedInt] External(pred:is-literal-not-unsignedInt( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:unsignedShort] External(pred:is-literal-not-unsignedShort( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:unsignedByte] External(pred:is-literal-not-unsignedByte( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:float] External(pred:is-literal-not-float( ?lt ) ) )
Forall ?lt (
rif:error() :- And (
?lt[rdf:type->xsd:double] External(pred:is-literal-not-double( ?lt ) ) )
Forall ?lt (

```

```

rif:error() :- And (
  ?lt[rdftype->xsd:string] External(pred:is-literal-not-string( ?lt )) )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:normalizedString] External(pred:is-literal-not-normalizedString( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:token] External(pred:is-literal-not-token( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:language] External(pred:is-literal-not-language( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:Name] External(pred:is-literal-not-Name( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:NCName] External(pred:is-literal-not-NCName( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:NMTOKEN] External(pred:is-literal-not-NMTOKEN( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:boolean] External(pred:is-literal-not-boolean( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:hexBinary] External(pred:is-literal-not-hexBinary( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:base64Binary] External(pred:is-literal-not-base64Binary( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:anyURI] External(pred:is-literal-not-anyURI( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:dateTime] External(pred:is-literal-not-dateTime( ?lt )) )
  )
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->xsd:dateTimeStamp] External(pred:is-literal-not-dateTimeStamp ( ?lt )) )
  )
)

(* <#eq-diff1-literal1> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x[owl:sameAs->?y]
    External(pred:literal-not-identical(?x ?y)) ) )

(* <#eq-diff1-literal2> *)
Forall ?x ?y ?s1 ?s2 ?p1 ?p2 (
  rif:error() :- And(
    ?s1[?p1->?x] ?s2[?p2->?y]
    ?x = ?y
    ?x[owl:differentFrom->?y] ) )

(* <#dt-type1-PlainLiteral> *) rdf:PlainLiteral[rdftype -> rdfs:Datatype]
(* <#dt-type1-decimal> *) xsd:decimal[rdftype -> rdfs:Datatype]
(* <#dt-type1-integer> *) xsd:integer[rdftype -> rdfs:Datatype]
(* <#dt-type1-double> *) xsd:double[rdftype -> rdfs:Datatype]
(* <#dt-type1-string> *) xsd:string[rdftype -> rdfs:Datatype]
(* <#dt-type1-dateTime> *) xsd:dateTime[rdftype -> rdfs:Datatype]
(* <#dt-type1-XMLLiteral> *) rdf:XMLLiteral[rdftype -> rdfs:Datatype]
(* <#dt-type1-Literal> *) rdfs:Literal[rdftype -> rdfs:Datatype]

(* <#dt-type1-nonNegativeInteger> *) xsd:nonNegativeInteger[rdftype -> rdfs:Datatype]
(* <#dt-type1-nonPositiveInteger> *) xsd:nonPositiveInteger[rdftype -> rdfs:Datatype]
(* <#dt-type1-positiveInteger> *) xsd:positiveInteger[rdftype -> rdfs:Datatype]
(* <#dt-type1-negativeInteger> *) xsd:negativeInteger[rdftype -> rdfs:Datatype]
(* <#dt-type1-long> *) xsd:long[rdftype -> rdfs:Datatype]
(* <#dt-type1-int> *) xsd:int[rdftype -> rdfs:Datatype]
(* <#dt-type1-short> *) xsd:short[rdftype -> rdfs:Datatype]
(* <#dt-type1-byte> *) xsd:byte[rdftype -> rdfs:Datatype]
(* <#dt-type1-unsignedLong> *) xsd:unsignedLong[rdftype -> rdfs:Datatype]
(* <#dt-type1-unsignedInt> *) xsd:unsignedInt[rdftype -> rdfs:Datatype]
(* <#dt-type1-unsignedShort> *) xsd:unsignedShort[rdftype -> rdfs:Datatype]
(* <#dt-type1-unsignedByte> *) xsd:unsignedByte[rdftype -> rdfs:Datatype]
(* <#dt-type1-normalizedString> *) xsd:normalizedString[rdftype -> rdfs:Datatype]
(* <#dt-type1-token> *) xsd:token[rdftype -> rdfs:Datatype]
(* <#dt-type1-language> *) xsd:language[rdftype -> rdfs:Datatype]
(* <#dt-type1-Name> *) xsd:Name[rdftype -> rdfs:Datatype]
(* <#dt-type1-NCName> *) xsd:NCName[rdftype -> rdfs:Datatype]
(* <#dt-type1-NMTOKEN> *) xsd:NMTOKEN[rdftype -> rdfs:Datatype]

(* <#dt-type1-float> *) xsd:float[rdftype -> rdfs:Datatype]
(* <#dt-type1-boolean> *) xsd:boolean[rdftype -> rdfs:Datatype]
(* <#dt-type1-hexBinary> *) xsd:hexBinary[rdftype -> rdfs:Datatype]
(* <#dt-type1-base64Binary> *) xsd:base64Binary[rdftype -> rdfs:Datatype]
(* <#dt-type1-anyURI> *) xsd:anyURI[rdftype -> rdfs:Datatype]
(* <#dt-type1-dateTimeStamp> *) xsd:dateTimeStamp [rdftype -> rdfs:Datatype]

```

8.2 templateRules algorithm

We specify the algorithm for instantiating the template rules by means of a translation table.

The first column gives a set of RDF triple patterns (*s p o*) where any of the elements can be a variable (indicated with a '?' prefix).

The second column gives a template to be instantiated. For each match of the triple patterns in *RDF(O)* we generate a binding map which maps each variable in the triple pattern to a corresponding RDF Node in *RDF(O)*. The template should be processed with this binding map, replacing the corresponding variables by their mapped values.

For example the pair:

Pattern	Template
(?p rdf:type owl:SymmetricProperty)	Forall ?x ?y (?y[?p->?x] :- And(?x[?p->?y]))

Applied to an ontology containing the following RDF triples:

```
eg:p rdf:type owl:SymmetricProperty .
eg:q rdf:type owl:SymmetricProperty .
```

Would emit the follow RIF rules:

```
Forall ?x ?y (
  ?y[eg:p->?x] :- And(
    ?x[eg:p->?y] ))

Forall ?x ?y (
  ?y[eg:q->?x] :- And(
    ?x[eg:q->?y] ))
```

In order to specify the list related rules we need to define some addition notation for templates.

Notation	Interpretation
{ \$ rule text \$ }	Emit the <i>rule text</i> substituting any occurrence of variables from the binding map. The outer { \$ \$ } can be omitted in cases where there is no ambiguity.
for(?elt in ?list) { template }	?list is a variable in the pattern which is bound to an RDF List. The <i>for</i> operator iterates over each element of the ?list in turn replacing the binding map for the ?elt variable with the next list entry and processes the enclosed <i>template</i> in the context of that new binding map.
for(?elt in ?list, != ?other) { template }	?list is a variable in the pattern which is bound to an RDF List. The <i>for</i> operator iterates over each element of the ?list in turn, skipping any element identical to ?other, replacing the binding map for the ?elt variable with the next list entry and processes the enclosed <i>template</i> in the context of that new binding map.
\$length(?list)\$	Used within a template this will be replaced by the length the RDFList bound to ?list.
\$i\$	Used within a <i>for(?elt in ?list){ template }</i> this will be replaced by the index of the current ?elt.
\$i+1\$	Used within a <i>for(?elt in ?list){ template }</i> this will be replaced by the index of the current ?elt, plus 1.

The templateRule algorithm is defined using this notation by the following set of pattern/template pairs.

Pattern	Template
(?p rdf:type owl:FunctionalProperty)	(* <#prp-fp> *) Forall ?y2 ?x ?y1 (?y1[owl:sameAs->?y2] :- And(?x[?p->?y1] ?x[?p->?y2]))
(?p rdf:type owl:InverseFunctionalProperty)	(* <#prp-ifp> *) Forall ?x1 ?x2 ?y (?x1[owl:sameAs->?x2] :- And(?x1[?p->?y] ?x2[?p->?y]))
(?p rdf:type owl:IrreflexiveProperty)	(* <#prp-irp> *) Forall ?x (rif:error() :- And(?x[?p->?x]))
(?p rdf:type owl:SymmetricProperty)	(* <#prp-symp> *) Forall ?x ?y (?y[?p->?x] :- And(?x[?p->?y]))
(?p rdf:type owl:AsymmetricProperty)	(* <#prp-asymp> *) Forall ?x ?y (rif:error() :- And(?x[?p->?y] ?y[?p->?x]))

<p>(?p rdf:type owl:TransitiveProperty)</p>	<pre>(* <#prp-trp> *) Forall ?x ?z ?y (?x[?p->?z] :- And(?x[?p->?y] ?y[?p->?z]))</pre>
<p>(?p1 rdfs:subPropertyOf ?p2)</p>	<pre>(* <#prp-spol> *) Forall ?x ?y (?x[?p2->?y] :- And(?x[?p1->?y]))</pre>
<p>(?p1 owl:equivalentProperty ?p2)</p>	<pre>(* <#prp-eqp1> *) Forall ?x ?y (?x[?p2->?y] :- And(?x[?p1->?y])) (* <#prp-eqp2> *) Forall ?x ?y (?x[?p1->?y] :- And(?x[?p2->?y]))</pre>
<p>(?p1 owl:propertyDisjointWith ?p2)</p>	<pre>(* <#prp-pdw> *) Forall ?x ?y (rif:error() :- And(?x[?p1->?y] ?x[?p2->?y]))</pre>
<p>(?p1 owl:inverseOf ?p2)</p>	<pre>(* <#prp-inv1> *) Forall ?x ?y (?y[?p2->?x] :- And(?x[?p1->?y])) (* <#prp-inv2> *) Forall ?x ?y (?y[?p1->?x] :- And(?x[?p2->?y]))</pre>
<p>(?x owl:someValuesFrom ?y) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-svf1> *) Forall ?v ?u (?u[rdf:type->?x] :- And(?u[?p->?v] ?v[rdf:type->?y]))</pre>
<p>(?x owl:someValuesFrom owl:Thing) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-svf2> *) Forall ?v ?u (?u[rdf:type->?x] :- And(?u[?p->?v]))</pre>
<p>(?x owl:allValuesFrom ?y) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-avf> *) Forall ?v ?u (?v[rdf:type->?y] :- And(?u[rdf:type->?x] ?u[?p->?v]))</pre>
<p>(?x owl:hasValue ?y) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-hv1> *) Forall ?u (?u[?p->?y] :- And(?u[rdf:type->?x])) (* <#cls-hv2> *) Forall ?u (?u[rdf:type->?x] :- And(?u[?p->?y]))</pre>
<p>(?x owl:maxCardinality 0) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-maxc1> *) Forall ?u ?y (rif:error() :- And(?u[?p->?y] ?u[rdf:type->?x]))</pre>
<p>(?x owl:maxCardinality 1) (?x owl:onProperty ?p)</p>	<pre>(* <#cls-maxc2> *) Forall ?y2 ?u ?y1 (?y1[owl:sameAs->?y2] :- And(?u[?p->?y1] ?u[?p->?y2] ?u[rdf:type->?x]))</pre>
<p>(?x owl:maxQualifiedCardinality 0) (?x owl:onProperty ?p) (?x owl:onClass ?c)</p>	<pre>(* <#cls-maxqc1> *) Forall ?u ?y (rif:error() :- And(</pre>

	<pre>?u[rdf:type->?x] ?u[?p->?y] ?y[rdf:type->?c]))</pre>
<pre>(?x owl:maxQualifiedCardinality 0) (?x owl:onProperty ?p) (?x owl:onClass owl:Thing)</pre>	<pre>(* <#cls-maxqc2> *) Forall ?y ?u (rif:error() :- And(?u[rdf:type->?x] ?u[?p->?y]))</pre>
<pre>(?x owl:maxQualifiedCardinality 1) (?x owl:onProperty ?p) (?x owl:onClass ?c)</pre>	<pre>(* <#cls-maxqc3> *) Forall ?y2 ?u ?y1 (?y1[owl:sameAs->?y2] :- And(?u[rdf:type->?x] ?u[?p->?y1] ?y1[rdf:type->?c] ?u[?p->?y2] ?y2[rdf:type->?c]))</pre>
<pre>(?x owl:maxQualifiedCardinality 1) (?x owl:onProperty ?p) (?x owl:onClass owl:Thing)</pre>	<pre>(* <#cls-maxqc4> *) Forall ?y2 ?u ?y1 (?y1[owl:sameAs->?y2] :- And(?u[rdf:type->?x] ?u[?p->?y1] ?u[?p->?y2]))</pre>
<pre>(?c1 owl:hasValue ?i) (?c1 owl:onProperty ?p1) (?c2 owl:hasValue ?i) (?c2 owl:onProperty ?p2)</pre>	<pre>(* <#scm-hv> *) ?c1[rdfs:subClassOf->?c2] :- And(?p1[rdfs:subPropertyOf->?p2])</pre>
<pre>(?c1 owl:someValuesFrom ?y1) (?c1 owl:onProperty ?p) (?c2 owl:someValuesFrom ?y2) (?c2 owl:onProperty ?p)</pre>	<pre>(* <#scm-svf1> *) ?c1[rdfs:subClassOf->?c2] :- And(?y1[rdfs:subClassOf->?y2])</pre>
<pre>(?c1 owl:someValuesFrom ?y) (?c1 owl:onProperty ?p1) (?c2 owl:someValuesFrom ?y) (?c2 owl:onProperty ?p2)</pre>	<pre>(* <#scm-svf2> *) ?c1[rdfs:subClassOf->?c2] :- And(?p1[rdfs:subPropertyOf->?p2])</pre>
<pre>(?c1 owl:allValuesFrom ?y1) (?c1 owl:onProperty ?p) (?c2 owl:allValuesFrom ?y2) (?c2 owl:onProperty ?p)</pre>	<pre>(* <#scm-avf1> *) ?c1[rdfs:subClassOf->?c2] :- And(?y1[rdfs:subClassOf->?y2])</pre>
<pre>(?c1 owl:allValuesFrom ?y) (?c1 owl:onProperty ?p1) (?c2 owl:allValuesFrom ?y) (?c2 owl:onProperty ?p2)</pre>	<pre>(* <#scm-avf2> *) ?c2[rdfs:subClassOf->?c1] :- And(?p1[rdfs:subPropertyOf->?p2])</pre>
<pre>(?a rdf:type owl:AllDifferent) (?a owl:members ?l)</pre>	<pre>(* <#eq-diff2> *) for(?x in ?l) { for(?y in ?l, != x) { {\$ rif:error() :- And (?x[owl:sameAs->?y]) } } }</pre>
<pre>(?a rdf:type owl:AllDifferent) (?a owl:distinctMembers ?l)</pre>	<pre>(* <#eq-diff3> *) for(?x in ?l) { for(?y in ?l, != ?x) { {\$ rif:error() :- And (?x[owl:sameAs->?y]) } } }</pre>
<pre>(?l rdf:type owl:AllDisjointProperties)</pre>	<pre>(* <#prp-adp> *) for(?x in ?l) { for(?y in ?l, != ?x) { {\$ Forall ?o ?v (rif:error() :- And (?o[?x->?v ?y->?v])) } } }</pre>

<p>(?l rdf:type owl:AllDisjointClasses)</p>	<pre>(* <#cax-adc> *) for(?x in ?l) { for(?y in ?l, != ?x) { {\$ Forall ?something (rif:error() :- And (?something[rdf:type->?x rdf:type->?y])) } } } }</pre>
<p>(?p owl:propertyChainAxiom ?pc)</p>	<pre>(* <#prp-spo2> *) {\$ Forall {\$ for(?next in ?pc) { {\$?u\$i\$ \$} } } {\$?u\$length(?pc)\$ (?u\$0\$[?p->?u\$length(?pc)\$] :- And ()) } for(?next in ?pc) { {\$?u\$i\$[?next->?u\$i+1\$] } } {\$) \$} }</pre>
<p>(?c owl:hasKey ?u)</p>	<pre>(* <#prp-key> *) {\$ Forall ?x ?y ?v (?x[owl:sameAs->?y] :- And (?x[rdf:type->?c] ?y[rdf:type->?c])) for(?key in ?u) { {\$?x[?key->?v] ?y[?key->?v] } } {\$) \$} }</pre>
<p>(?c owl:intersectionOf ?l)</p>	<pre>(* <#cls-int2> *) {\$ Forall ?y (?y[rdf:type->?c] :- And ()) } for(?ty in ?l) { {\$?y[rdf:type->?ty] } } {\$) \$} }</pre>
<p>(?c owl:unionOf ?l)</p>	<pre>(* <#cls-uni> *) for(?ci in ?l) { {\$ Forall ?y (?y[rdf:type->?c] :- And (?y[rdf:type->?ci])) } } }</pre>
<p>(?c owl:oneOf ?l)</p>	<pre>(* <#cls-oo> *) for(?yi in ?l) { {\$?yi[rdf:type->?c] } }</pre>
<p>(?c owl:intersectionOf ?l)</p>	<pre>(* <#scm-int> *) for(?ci in ?l) { {\$ Forall ?y (?y[rdf:type->?ci] :- And (?y[rdf:type->?c])) } ?c[rdfs:subClassOf->?ci] } }</pre>
<p>(?c owl:unionOf ?l)</p>	<pre>(* <#scm-uni> *) for(?ci in ?l) { {\$?ci[rdfs:subClassOf->?c] } }</pre>

	<pre> \$} } </pre>
<pre> (?x owl:sourceIndividual ?i1) (?x owl:assertionProperty ?p) (?x owl:targetIndividual ?i2) </pre>	<pre> (* <#prp-npa1> *) rif:error() :- And(?i1[?p->?i2]) </pre>
<pre> (?x owl:sourceIndividual ?i1) (?x owl:assertionProperty ?p) (?x owl:targetValue ?i2) </pre>	<pre> (* <#prp-npa2> *) rif:error() :- And(?i1[?p->?i2]) </pre>
<pre> (?c1 owl:disjointWith ?c2) </pre>	<pre> (* <#cax-dw> *) Forall ?x (rif:error() :- And(?x[rdftype->?c1] ?x[rdftype->?c2])) </pre>
<pre> (?c1 owl:complementOf ?c2) </pre>	<pre> (* <#cls-com> *) Forall ?x (rif:error() :- And(?x[rdftype->?c1] ?x[rdftype->?c2])) </pre>

9 Appendix: OWL 2 RL ruleset - XML syntax

We provide the fixed ruleset in XML syntax below, split into three groups.

Simple triple rules

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Document [
  <!ENTITY rif "http://www.w3.org/2007/rif#">
  <!ENTITY xs "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
]>
<Document xmlns="&rif;">
  <payload>
    <Group>
      <sentence>
        <Forall>
          <declare>
            <Var>p</Var>
          </declare>
          <declare>
            <Var>o</Var>
          </declare>
          <declare>
            <Var>s</Var>
          </declare>
          <formula>
            <Implies>
              <if>
                <Frame>
                  <object>
                    <Var>s</Var>
                  </object>
                  <slot ordered="yes">
                    <Var>p</Var>
                    <Var>o</Var>
                  </slot>
                </Frame>
              </if>
              <then>
                <Frame>
                  <object>
                    <Var>s</Var>
                  </object>
                  <slot ordered="yes">
                    <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                    <Var>s</Var>
                  </slot>
                </Frame>
              </then>
            </Implies>
          </formula>
        </Forall>
      </sentence>
      <sentence>
        <Forall>
          <declare>
            <Var>p</Var>
          </declare>
          <declare>
            <Var>o</Var>
          </declare>
          <declare>
            <Var>s</Var>
          </declare>
        </Forall>
      </sentence>
    </Group>
  </payload>
</Document>
        
```

```

</declare>
<formula>
  <Implies>
    <if>
      <Frame>
        <object>
          <Var>s</Var>
        </object>
        <slot ordered="yes">
          <Var>p</Var>
          <Var>o</Var>
        </slot>
      </Frame>
    </if>
    <then>
      <Frame>
        <object>
          <Var>p</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
          <Var>p</Var>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>o</Var>
    </declare>
    <declare>
      <Var>s</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
              <Var>o</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>o</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
              <Var>o</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
              <Var>y</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>y</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
              <Var>x</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>z</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>z</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
              <Var>z</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>o</Var>
    </declare>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>s2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>s2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>o</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </Frame>
      </formula>
    </And>
  </if>
  <then>
    <Frame>
      <object>
        <Var>s2</Var>
      </object>
      <slot ordered="yes">
        <Var>p</Var>
        <Var>o</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>o</Var>
    </declare>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>o</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p2</Var>
              <Var>o</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>o</Var>
    </declare>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>o2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>o</Var>
                </object>

```

```

        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
          <Var>o2</Var>
        </slot>
      </Frame>
    </formula>
  </formula>
  <Frame>
    <object>
      <Var>s</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>o</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>s</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>o2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#differentFrom</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#label</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#comment</Const>
    </object>

```

```

    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#seeAlso</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#isDefinedBy</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#deprecated</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#priorVersion</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#backwardCompatibleWith</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#incompatibleWith</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AnnotationProperty</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#domain</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>

```



```

        </slot>
      </Frame>
    </formula>
  </formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#range</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>y</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Var>c</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>y2</Var>
    </declare>
    <declare>
      <Var>x</Var>

```

```

</declare>
<declare>
  <Var>y1</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#FunctionalProperty</Const>
            </slot>
          </Frame>
        </formula>
      </And>
      <formula>
        <Frame>
          <object>
            <Var>x</Var>
          </object>
          <slot ordered="yes">
            <Var>p</Var>
            <Var>y1</Var>
          </slot>
        </Frame>
      </formula>
    </if>
    <then>
      <Frame>
        <object>
          <Var>y1</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
          <Var>y2</Var>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>x1</Var>
    </declare>
    <declare>
      <Var>x2</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#InverseFunctionalProperty</Const>
                </slot>
              </Frame>
            </formula>
          </And>
          <formula>
            <Frame>
              <object>
                <Var>x1</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    <formula>
      <Frame>
        <object>
          <Var>x2</Var>
        </object>
        <slot ordered="yes">
          <Var>p</Var>
          <Var>y</Var>
        </slot>
      </Frame>
    </formula>
  </And>
</if>
<then>
  <Frame>
    <object>
      <Var>x1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>x2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#IrreflexiveProperty</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#SymmetricProperty</Const>
                </slot>
              </Frame>
            </formula>
            <Atom>
              <op>
                <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
              </op>
            </Atom>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    </slot>
  </Frame>
</formula>
</formula>
<Frame>
  <object>
    <Var>x</Var>
  </object>
  <slot ordered="yes">
    <Var>p</Var>
    <Var>y</Var>
  </slot>
</Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>x</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#AsymmetricProperty</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
</sentence>
<forall>
  <declare>
    <Var>p</Var>
  </declare>
</forall>

```

```

    <Var>x</Var>
  </declare>
  <declare>
    <Var>z</Var>
  </declare>
  <declare>
    <Var>y</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>p</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#TransitiveProperty</Const>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>y</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>z</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>x</Var>
          </object>
          <slot ordered="yes">
            <Var>p</Var>
            <Var>z</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Var>p2</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Var>p2</Var>
              <Var>y</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>

```

```

<object>
  <Var>p1</Var>
</object>
<slot ordered="yes">
  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
  <Var>p2</Var>
</slot>
</Frame>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Var>p2</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Var>p1</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#propertyDisjointWith</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Var>p2</Var>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
  <op>
    <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
  </op>
</Atom>
</then>
</Implies>

```

```

</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#inverse0f</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>y</Var>
            </object>
            <slot ordered="yes">
              <Var>p2</Var>
              <Var>x</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#inverse0f</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>p2</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

        </Frame>
      </formula>
    </And>
  </if>
  <then>
    <Frame>
      <object>
        <Var>y</Var>
      </object>
      <slot ordered="yes">
        <Var>p1</Var>
        <Var>x</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&ref;iri">http://www.w3.org/2002/07/owl#Thing</Const>
    </object>
    <slot ordered="yes">
      <Const type="&ref;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&ref;iri">http://www.w3.org/2002/07/owl#Class</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&ref;iri">http://www.w3.org/2002/07/owl#Nothing</Const>
    </object>
    <slot ordered="yes">
      <Const type="&ref;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&ref;iri">http://www.w3.org/2002/07/owl#Class</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>x</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&ref;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&ref;iri">http://www.w3.org/2002/07/owl#Nothing</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&ref;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>v</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&ref;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>

```

```

        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
      <Var>p</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>v</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>v</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>x</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>v</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
  </forall>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Thing</Const>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>

```

```

        </object>
        <slot ordered="yes">
          <Var>p</Var>
          <Var>v</Var>
        </slot>
      </Frame>
    </formula>
  </And>
</if>
<then>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>x</Var>
    </slot>
  </Frame>
</then>
</Implies>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>v</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#allValuesFrom</Const>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>u</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Var>x</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>u</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>v</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>v</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          </slot>
        </Frame>
      </then>
    </Implies>
  </forall>
</sentence>

```

```

    <Var>y</Var>
  </slot>
</Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
<Forall>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>u</Var>
  </declare>
  <declare>
    <Var>x</Var>
  </declare>
  <declare>
    <Var>y</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasValue</Const>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>u</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Var>x</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>u</Var>
          </object>
          <slot ordered="yes">
            <Var>p</Var>
            <Var>y</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>
</Forall>
</sentence>
</sentence>
<Forall>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>u</Var>
  </declare>
  <declare>
    <Var>x</Var>
  </declare>
  <declare>
    <Var>y</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>

```

```

</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasValue</Const>
  <Var>y</Var>
</slot>
</Frame>
</formula>
</formula>
<Frame>
  <object>
    <Var>x</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
    <Var>p</Var>
  </slot>
</Frame>
</formula>
</formula>
<Frame>
  <object>
    <Var>u</Var>
  </object>
  <slot ordered="yes">
    <Var>p</Var>
    <Var>y</Var>
  </slot>
</Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>x</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxCardinality</Const>
                  <Const type="&xs;integer">0</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>u</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Var>x</Var>
              </slot>
            </Frame>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
</forall>

```

```

    <Frame>
      <object>
        <Var>u</Var>
      </object>
      <slot ordered="yes">
        <Var>p</Var>
        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>y2</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxCardinality</Const>
                  <Const type="&xs;integer">1</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
</sentence>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y1</Var>
    </slot>
  </Frame>
</formula>
</sentence>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y2</Var>
    </slot>
  </Frame>
</sentence>

```

```

    </formula>
  </And>
</if>
</then>
  <Frame>
    <object>
      <Var>y1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>y2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxQualifiedCardinality</Const>
      <Const type="&xs;integer">0</Const>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxQualifiedCardinality</Const>
                  <Const type="&xs;integer">0</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onClass</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <formula>
    <Frame>
      <object>
        <Var>y</Var>
      </object>
    </Frame>
  </formula>

```

```

</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Var>c</Var>
</slot>
</Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxQualifiedCardinality</Const>
                  <Const type="&xs;integer">0</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onClass</Const>
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Thing</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </Atom>
  </op>

```



```

<formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y2</Var>
    </slot>
  </Frame>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>y1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>y2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>y2</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#maxQualifiedCardinality</Const>
                  <Const type="&xs;integer">1</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onClass</Const>
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Thing</Const>
                </slot>
              </Frame>
            </formula>
          </And>
          <formula>
            <Frame>
              <object>
                <Var>u</Var>
              </object>
            </Frame>
          </formula>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Var>x</Var>
        </slot>
      </Frame>
    </formula>
  </formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y1</Var>
    </slot>
  </Frame>
</formula>
</formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>y1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>y2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>c1</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <then>
    <Frame>
      <object>
        <Var>x</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Var>c2</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>

```

```

</sentence>
<sentence>
  <forall>
    <declare>
      <var>x</var>
    </declare>
    <declare>
      <var>c1</var>
    </declare>
    <declare>
      <var>c2</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>c1</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentClass</const>
                  <var>c2</var>
                </slot>
              </frame>
            </formula>
            <formula>
              <frame>
                <object>
                  <var>x</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</const>
                  <var>c1</var>
                </slot>
              </frame>
            </formula>
          </and>
        </if>
        <then>
          <frame>
            <object>
              <var>x</var>
            </object>
            <slot ordered="yes">
              <const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</const>
              <var>c2</var>
            </slot>
          </frame>
        </then>
      </implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>x</var>
    </declare>
    <declare>
      <var>c1</var>
    </declare>
    <declare>
      <var>c2</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>c1</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentClass</const>
                  <var>c2</var>
                </slot>
              </frame>
            </formula>
            <formula>
              <frame>
                <object>
                  <var>x</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</const>
                  <var>c2</var>
                </slot>
              </frame>
            </formula>
          </and>
        </if>
        <then>
          <frame>
            <object>
              <var>x</var>
            </object>

```

```

    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#disjointWith</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
            </slot>
            <slot type="&rf;iri">http://www.w3.org/2002/07/owl#Class</Const>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>

```

```

        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#Class</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentClass</Const>
              <Var>c</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#Class</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#Thing</Const>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#Class</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#Nothing</Const>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>

```

```

        <Var>c</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>c3</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>c3</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
              <Var>c3</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>c1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentClass</Const>
              <Var>c2</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
              <Var>c2</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

<sentence>
  <forall>
    <declare>
      <var>c1</var>
    </declare>
    <declare>
      <var>c2</var>
    </declare>
    <formula>
      <implies>
        <if>
          <frame>
            <object>
              <var>c1</var>
            </object>
            <slot ordered="yes">
              <const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentClass</const>
              <var>c2</var>
            </slot>
          </frame>
        </if>
        <then>
          <frame>
            <object>
              <var>c2</var>
            </object>
            <slot ordered="yes">
              <const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</const>
              <var>c1</var>
            </slot>
          </frame>
        </then>
      </implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>c1</var>
    </declare>
    <declare>
      <var>c2</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>c1</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</const>
                  <var>c2</var>
                </slot>
              </frame>
            </formula>
            <formula>
              <frame>
                <object>
                  <var>c2</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</const>
                  <var>c1</var>
                </slot>
              </frame>
            </formula>
          </and>
        </if>
        <then>
          <frame>
            <object>
              <var>c1</var>
            </object>
            <slot ordered="yes">
              <const type="&rif;iri">http://www.w3.org/2002/07/owl#equivalentClass</const>
              <var>c2</var>
            </slot>
          </frame>
        </then>
      </implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>p</var>
    </declare>
    <formula>
      <implies>
        <if>
          <frame>
            <object>
              <var>p</var>
            </object>
          </frame>
        </if>
      </implies>
    </formula>
  </forall>
</sentence>

```



```

    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#ObjectProperty</Const>
    </slot>
  </Frame>
</if>
<then>
  <Frame>
    <object>
      <Var>p</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
      <Var>p</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#ObjectProperty</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#EquivalentProperty</Const>
              <Var>p</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#DatatypeProperty</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
              <Var>p</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#DatatypeProperty</Const>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
              <Var>p</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>

```

```

    <Var>p</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rf;iri">http://www.w3.org/2002/07/owl#DatatypeProperty</Const>
  </slot>
</Frame>
</if>
<then>
  <Frame>
    <object>
      <Var>p</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p3</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p3</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>p1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
              <Var>p3</Var>
            </slot>
          </Frame>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<forall>
  <declare>
    <Var>p2</Var>
  </declare>
  <declare>
    <Var>p1</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <Frame>
          <object>
            <Var>p1</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
            <Var>p2</Var>
          </slot>
        </Frame>
      </if>
    </Implies>
  </formula>
</forall>
</if>

```

```

<then>
  <Frame>
    <object>
      <Var>p1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
      <Var>p2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>p1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
              <Var>p2</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p2</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
              <Var>p1</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
</sentence>
<Forall>
  <declare>
    <Var>p2</Var>
  </declare>
  <declare>
    <Var>p1</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>p1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                <Var>p2</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>p2</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                <Var>p1</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>p1</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rf;iri">http://www.w3.org/2002/07/owl#equivalentProperty</Const>
            <Var>p2</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>
</Forall>

```

```

    </Implies>
  </formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#domain</Const>
                  <Var>c1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#domain</Const>
              <Var>c2</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#domain</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

    <Frame>
      <object>
        <Var>p1</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#domain</Const>
        <Var>c</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#range</Const>
                  <Var>c1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#range</Const>
              <Var>c2</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#range</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
      <Var>p2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>p1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#range</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>i</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>c1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasValue</Const>
                <Var>i</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p1</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c2</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasValue</Const>
                <Var>i</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c2</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p2</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>p1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                <Var>p2</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
</forall>
</sentence>

```

```

    </Frame>
  </formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>c1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
      <Var>c2</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>y2</Var>
    </declare>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>y1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>
                  <Var>y1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>
                  <Var>y2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                  <Var>y2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
</then>

```

```

    <Frame>
      <object>
        <Var>c1</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
        <Var>c2</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#someValuesFrom</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c1</Var>
            </object>
            <slot ordered="yes">

```



```

        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
        <Var>c2</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
<forall>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>y2</Var>
  </declare>
  <declare>
    <Var>c1</Var>
  </declare>
  <declare>
    <Var>c2</Var>
  </declare>
  <declare>
    <Var>y1</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>c1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#allValuesFrom</Const>
                <Var>y1</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c2</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#allValuesFrom</Const>
                <Var>y2</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>c2</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>y1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
                <Var>y2</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </then>
    <Frame>
      <object>
        <Var>c1</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
        <Var>c2</Var>
      </slot>
    </Frame>
  </then>

```

```

    </Implies>
  </formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#allValuesFrom</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#allValuesFrom</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>c2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#onProperty</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c2</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
              <Var>c1</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
</sentence>

```

```

<forall>
  <declare>
    <Var>x</Var>
  </declare>
  <declare>
    <Var>i1</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>i2</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sourceIndividual</Const>
                <Var>i1</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2002/07/owl#assertionProperty</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2002/07/owl#targetIndividual</Const>
                <Var>i2</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>i1</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>i2</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
      <then>
        <Atom>
          <op>
            <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
          </op>
        </Atom>
      </then>
    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>i</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">

```

```

        <Const type="&rf;iri">http://www.w3.org/2002/07/owl#sourceIndividual</Const>
        <Var>i</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<Frame>
  <object>
    <Var>x</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rf;iri">http://www.w3.org/2002/07/owl#assertionProperty</Const>
    <Var>p</Var>
  </slot>
</Frame>
</formula>
</formula>
<Frame>
  <object>
    <Var>x</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rf;iri">http://www.w3.org/2002/07/owl#targetValue</Const>
    <Var>lt</Var>
  </slot>
</Frame>
</formula>
</formula>
<Frame>
  <object>
    <Var>i</Var>
  </object>
  <slot ordered="yes">
    <Var>p</Var>
    <Var>lt</Var>
  </slot>
</Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<forall>
  <declare>
    <Var>c1</Var>
  </declare>
  <declare>
    <Var>c2</Var>
  </declare>
  <declare>
    <Var>x</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>c1</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/2002/07/owl#disjointWith</Const>
                <Var>c2</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Var>c1</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <formula>
    <Frame>
      <object>
        <Var>x</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Var>c2</Var>
      </slot>
    </Frame>
  </formula>
</forall>

```

```

        </formula>
      </And>
    </if>
  </then>
  <Atom>
    <op>
      <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>c1</Var>
    </declare>
    <declare>
      <Var>c2</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#complementOf</Const>
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>c1</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </then>
    <Atom>
      <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
      </op>
    </Atom>
  </then>
</Implies>
</forall>
</sentence>
</Group>
</payload>
</Document>

```

Datatype rules

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Document [
  <!ENTITY rif "http://www.w3.org/2007/rif#">
  <!ENTITY xs "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
]>
<Document xmlns="&rf;">
  <payload>
    <Group>
      <sentence>
        <Group>
          <sentence>
            <forall>
              <declare>
                <Var>s</Var>
              </declare>

```

```

<declare>
  <Var>p</Var>
</declare>
<declare>
  <Var>lt</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
              <Var>lt</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <External>
            <content>
              <Atom>
                <op>
                  <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-PlainLiteral</Const>
                </op>
                <args ordered="yes">
                  <Var>lt</Var>
                </args>
              </Atom>
            </content>
          </External>
        </formula>
      </And>
    </if>
    <then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-XMLLiteral</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-decimal</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#decimal</Const>
                </slot>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>

```

```

    <Var>s</Var>
  </object>
  <slot ordered="yes">
    <Var>p</Var>
    <Var>lt</Var>
  </slot>
</Frame>
</formula>
<formula>
  <External>
  <content>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-integer</Const>
    </op>
    <args ordered="yes">
      <Var>lt</Var>
    </args>
  </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
  <object>
    <Var>lt</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#integer</Const>
  </slot>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
  </slot>
  </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
  <Implies>
  <if>
    <And>
    <formula>
    <Frame>
    <object>
      <Var>s</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>lt</Var>
    </slot>
    </Frame>
    </formula>
    <formula>
    <External>
    <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-nonNegativeInteger</Const>
      </op>
      <args ordered="yes">
        <Var>lt</Var>
      </args>
    </Atom>
    </content>
    </External>
    </formula>
  </And>
  </if>
  <then>
  <Frame>
  <object>
    <Var>lt</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonNegativeInteger</Const>
  </slot>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
  </slot>

```



```

        </Frame>
      </then>
    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-nonPositiveInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
    <Frame>
      <object>
        <Var>lt</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonPositiveInteger</Const>
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
</sentence>
<forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>s</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>lt</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>

```

```

        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-positiveInteger</Const>
      </op>
      <args ordered="yes">
        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#positiveInteger</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-negativeInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#negativeInteger</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
  </forall>

```

```

</declare>
<declare>
  <Var>lt</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
              <Var>lt</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <External>
            <content>
              <Atom>
                <op>
                  <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-long</Const>
                </op>
                <args ordered="yes">
                  <Var>lt</Var>
                </args>
              </Atom>
            </content>
          </External>
        </formula>
      </And>
    </if>
    <then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#long</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-int</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Frame>

```

```

<object>
  <Var>lt</Var>
</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#int</Const>
</slot>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
</slot>
</Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-short</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#short</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>

```

```

        <slot ordered="yes">
          <Var>p</Var>
          <Var>lt</Var>
        </slot>
      </Frame>
    </formula>
  </formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-byte</Const>
        </op>
        <args ordered="yes">
          <Var>lt</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#byte</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-unsignedLong</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedLong</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>

```

```

    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-unsignedInt</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>lt</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#unsignedInt</Const>
            </slot>
            <slot ordered="yes">
              <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-unsignedShort</Const>
                    </op>

```

```

      <args ordered="yes">
        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedShort</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-unsignedByte</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      <then>
        <Frame>
          <object>
            <Var>lt</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
            <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedByte</Const>
          </slot>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
            <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
  </forall>

```

```

<Var>lt</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <External>
            <content>
              <Atom>
                <op>
                  <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-float</Const>
                </op>
                <args ordered="yes">
                  <Var>lt</Var>
                </args>
              </Atom>
            </content>
          </External>
        </formula>
      </And>
    </if>
    <then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#float</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-double</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>lt</Var>
            </object>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#double</Const>
</slot>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
</slot>
</Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-string</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#string</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>

```

```

        <Var>lt</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-normalizedString</Const>
      </op>
      <args ordered="yes">
        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#normalizedString</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>s</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>lt</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-token</Const>
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
    </Implies>
  </then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#token</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>

```

```

</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>s</var>
    </declare>
    <declare>
      <var>p</var>
    </declare>
    <declare>
      <var>lt</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>s</var>
                </object>
                <slot ordered="yes">
                  <var>p</var>
                  <var>lt</var>
                </slot>
              </frame>
            </formula>
            <formula>
              <external>
                <content>
                  <atom>
                    <op>
                      <const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-language</const>
                    </op>
                    <args ordered="yes">
                      <var>lt</var>
                    </args>
                  </atom>
                </content>
              </external>
            </formula>
          </and>
        </if>
        <then>
          <frame>
            <object>
              <var>lt</var>
            </object>
            <slot ordered="yes">
              <const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</const>
              <const type="&rf;iri">http://www.w3.org/2001/XMLSchema#language</const>
            </slot>
            <slot ordered="yes">
              <const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</const>
              <const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Literal</const>
            </slot>
          </frame>
        </then>
      </implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>s</var>
    </declare>
    <declare>
      <var>p</var>
    </declare>
    <declare>
      <var>lt</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>s</var>
                </object>
                <slot ordered="yes">
                  <var>p</var>
                  <var>lt</var>
                </slot>
              </frame>
            </formula>
            <formula>
              <external>
                <content>
                  <atom>
                    <op>
                      <const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-Name</const>
                    </op>
                    <args ordered="yes">
                      <var>lt</var>
                    </args>
                  </atom>
                </content>
              </external>
            </formula>
          </and>
        </if>
      </implies>
    </formula>
  </forall>
</sentence>

```

```

        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Name</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-NCName</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
    </Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#NCName</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
  </Forall>

```

```

<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>s</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
              <Var>lt</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <External>
            <content>
              <Atom>
                <op>
                  <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-NMTOKEN</Const>
                </op>
                <args ordered="yes">
                  <Var>lt</Var>
                </args>
              </Atom>
            </content>
          </External>
        </formula>
      </And>
    </if>
    <then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#NMTOKEN</Const>
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-boolean</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </forall>
  </sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <slot ordered="yes">

```

```

    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#boolean</Const>
  </slot>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
  </slot>
</Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
</sentence>
<forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>s</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>lt</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-hexBinary</Const>
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <then>
    <Frame>
      <object>
        <Var>lt</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#hexBinary</Const>
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
</sentence>
<forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>s</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>lt</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-hexBinary</Const>
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <then>
    <Frame>
      <object>
        <Var>lt</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#hexBinary</Const>
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
</sentence>
<forall>
  <declare>
    <Var>s</Var>
  </declare>
  <declare>
    <Var>p</Var>
  </declare>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>s</Var>
              </object>
              <slot ordered="yes">
                <Var>p</Var>
                <Var>lt</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-hexBinary</Const>
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <then>
    <Frame>
      <object>
        <Var>lt</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#hexBinary</Const>
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
</sentence>

```

```

    </Frame>
  </formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-base64Binary</Const>
      </op>
      <args ordered="yes">
        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#base64Binary</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-anyURI</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-dateTime</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>lt</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTime</Const>
            </slot>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>s</Var>
    </declare>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>lt</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-dateTimeStamp</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

        </content>
      </External>
    </formula>
  </And>
</if>
<then>
  <Frame>
    <object>
      <Var>lt</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTimeStamp</Const>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Literal</Const>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
</Group>
</sentence>
<sentence>
  <Group>
    <sentence>
      <Forall>
        <declare>
          <Var>lt</Var>
        </declare>
        <formula>
          <Implies>
            <if>
              <And>
                <formula>
                  <Frame>
                    <object>
                      <Var>lt</Var>
                    </object>
                    <slot ordered="yes">
                      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral</Const>
                    </slot>
                  </Frame>
                </formula>
              <formula>
                <External>
                  <content>
                    <Atom>
                      <op>
                        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-PlainLiteral</Const>
                      </op>
                      <args ordered="yes">
                        <Var>lt</Var>
                      </args>
                    </Atom>
                  </content>
                </External>
              </formula>
            </And>
          </if>
        </formula>
      </Forall>
    </sentence>
    <sentence>
      <Forall>
        <declare>
          <Var>lt</Var>
        </declare>
        <formula>
          <Implies>
            <if>
              <And>
                <formula>
                  <Frame>
                    <object>
                      <Var>lt</Var>
                    </object>
                    <slot ordered="yes">
                      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral</Const>
                    </slot>
                  </Frame>
                </formula>
              <formula>
                <External>
                  <content>

```



```

        <args ordered="yes">
          <Var>lt</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonNegativeInteger</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-nonNegativeInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonPositiveInteger</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-nonPositiveInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </content>
      </External>
    </formula>
  </And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#positiveInteger</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-positiveInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#negativeInteger</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-negativeInteger</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
</And>

```

```

</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Long</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-long</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
      </op>
    </Atom>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#int</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-int</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </then>
    <Atom>
      <op>

```

```

      <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#short</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-short</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
</sentence>
</Forall>
  <declare>
    <Var>lt</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>lt</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#byte</Const>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-byte</Const>
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
      <then>
        <Atom>
          <op>
            <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
          </op>
        </Atom>
      </then>
    </Implies>
  </formula>
</then>
</then>

```

```

    </Implies>
  </formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedLong</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-unsignedLong</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedInt</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-unsignedInt</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedShort</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-unsignedShort</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedByte</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-unsignedByte</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>

```



```

</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>lt</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#float</Const>
            </slot>
          </Frame>
        </formula>
      </And>
      <External>
        <content>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-float</Const>
            </op>
            <args ordered="yes">
              <Var>lt</Var>
            </args>
          </Atom>
        </content>
      </External>
    </formula>
  </if>
  <then>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
      </op>
    </Atom>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#double</Const>
                </slot>
              </Frame>
            </formula>
          </And>
          <External>
            <content>
              <Atom>
                <op>
                  <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-double</Const>
                </op>
                <args ordered="yes">
                  <Var>lt</Var>
                </args>
              </Atom>
            </content>
          </External>
        </formula>
      </if>
      <then>
        <Atom>
          <op>
            <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
          </op>
        </Atom>
      </then>
    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>

```

```

<And>
  <formula>
    <Frame>
      <object>
        <Var>lt</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#string</Const>
      </slot>
    </Frame>
  </formula>
  <formula>
    <External>
      <content>
        <Atom>
          <op>
            <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-string</Const>
          </op>
          <args ordered="yes">
            <Var>lt</Var>
          </args>
        </Atom>
      </content>
    </External>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#normalizedString</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-normalizedString</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>

```

```

    <Var>lt</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#token</Const>
  </slot>
</Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-token</Const>
        </op>
        <args ordered="yes">
          <Var>lt</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Language</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-language</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-NMTOKEN</Const>
        </op>
        <args ordered="yes">
          <Var>lt</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#boolean</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-boolean</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#hexBinary</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>

```

```

        <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-hexBinary</Const>
        </op>
        <args ordered="yes">
        <Var>lt</Var>
        </args>
        </Atom>
        </content>
        </External>
        </formula>
        </And>
        </if>
        <then>
        <Atom>
        <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
        </op>
        </Atom>
        </then>
        </Implies>
        </formula>
        </Forall>
        </sentence>
        <sentence>
        <Forall>
        <declare>
        <Var>lt</Var>
        </declare>
        <formula>
        <Implies>
        <if>
        <And>
        <formula>
        <Frame>
        <object>
        <Var>lt</Var>
        </object>
        <slot ordered="yes">
        <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#base64Binary</Const>
        </slot>
        </Frame>
        </formula>
        <formula>
        <External>
        <content>
        <Atom>
        <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-base64Binary</Const>
        </op>
        <args ordered="yes">
        <Var>lt</Var>
        </args>
        </Atom>
        </content>
        </External>
        </formula>
        </And>
        </if>
        <then>
        <Atom>
        <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
        </op>
        </Atom>
        </then>
        </Implies>
        </formula>
        </Forall>
        </sentence>
        <sentence>
        <Forall>
        <declare>
        <Var>lt</Var>
        </declare>
        <formula>
        <Implies>
        <if>
        <And>
        <formula>
        <Frame>
        <object>
        <Var>lt</Var>
        </object>
        <slot ordered="yes">
        <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#anyURI</Const>
        </slot>
        </Frame>
        </formula>
        <formula>
        <External>
        <content>
        <Atom>
        <op>
        <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-anyURI</Const>
        </op>
        <args ordered="yes">
    
```

```

        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTime</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-dateTime</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>lt</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>lt</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTimeStamp</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#is-literal-not-dateTimeStamp</Const>
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
            </op>
          </Atom>
        </then>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

    </External>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
</Group>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>s1</Var>
    </declare>
    <declare>
      <Var>s2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s1</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                  <Var>x</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>s2</Var>
                </object>
                <slot ordered="yes">
                  <Var>p2</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#literal-not-identical</Const>
        </op>
        <args ordered="yes">
          <Var>x</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>

```



```

    </Implies>
  </formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>s1</Var>
    </declare>
    <declare>
      <Var>s2</Var>
    </declare>
    <declare>
      <Var>p1</Var>
    </declare>
    <declare>
      <Var>p2</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>s1</Var>
                </object>
                <slot ordered="yes">
                  <Var>p1</Var>
                <Var>x</Var>
              </slot>
            </Frame>
          </formula>
            <formula>
              <Frame>
                <object>
                  <Var>s2</Var>
                </object>
                <slot ordered="yes">
                  <Var>p2</Var>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
        </if>
        <formula>
          <Equal>
            <left>
              <Var>x</Var>
            </left>
            <right>
              <Var>y</Var>
            </right>
          </Equal>
        </formula>
      </Implies>
    </formula>
  </And>
</if>
</then>
  <Atom>
    <op>
      <Const type="&rf;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>

```



```

<object>
  <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#positiveInteger</Const>
</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
</slot>
</Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#negativeInteger</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Long</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#int</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#short</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#byte</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedLong</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedInt</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedShort</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
    </Frame>
  </sentence>
<sentence>

```

```

<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsignedByte</Const>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
  </slot>
</Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#normalizedString</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#token</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#language</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Name</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#NCName</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#NMTOKEN</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#float</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#boolean</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</sentence>

```

```

<sentence>
  <Frame>
    <object>
      <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#hexBinary</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#base64Binary</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#anyURI</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rf;iri">http://www.w3.org/2001/XMLSchema#dateTimeStamp</Const>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-schema#Datatype</Const>
    </slot>
  </Frame>
</sentence>
</Group>
</payload>
</Document>

```

List rules

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Document [
  <!ENTITY rif "http://www.w3.org/2007/rif#">
  <!ENTITY xs "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
]>
<Document xmlns="&rf;">
  <payload>
    <Group>
      <sentence>
        <Forall>
          <declare>
            <Var>x</Var>
          </declare>
          <declare>
            <Var>y</Var>
          </declare>
          <declare>
            <Var>z1</Var>
          </declare>
          <declare>
            <Var>z2</Var>
          </declare>
          <declare>
            <Var>iz1</Var>
          </declare>
          <declare>
            <Var>iz2</Var>
          </declare>
          <formula>
            <Implies>
              <if>
                <And>
                  <formula>
                    <Frame>
                      <object>
                        <Var>x</Var>
                      </object>
                      <slot ordered="yes">
                        <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                        <Const type="&rf;iri">http://www.w3.org/2002/07/owl#AllDifferent</Const>
                      </slot>
                    </Frame>
                  </formula>
                </And>
              </if>
            </Implies>
          </formula>
        </formula>
      </formula>
    </formula>
  </formula>
</formula>
</Frame>

```

```

<object>
  <Var>x</Var>
</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#members</Const>
  <Var>y</Var>
</slot>
</Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>y</Var>
          <Var>z1</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>iz1</Var>
    </left>
    <right>
      <External>
        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>y</Var>
              <Var>z1</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>y</Var>
          <Var>z2</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>iz2</Var>
    </left>
    <right>
      <External>
        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>y</Var>
              <Var>z2</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#numeric-not-equal</Const>
        </op>
        <args ordered="yes">
          <Var>iz1</Var>
          <Var>iz2</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>

```

```

<formula>
  <Frame>
    <object>
      <Var>x1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>z2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>z1</Var>
    </declare>
    <declare>
      <Var>z2</Var>
    </declare>
    <declare>
      <Var>iz1</Var>
    </declare>
    <declare>
      <Var>iz2</Var>
    </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AllDifferent</Const>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Frame>
              <object>
                <Var>x</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#distinctMembers</Const>
                <Var>y</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
                  </op>
                  <args ordered="yes">
                    <Var>y</Var>
                    <Var>z1</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
          <formula>
            <Equal>
              <left>
                <Var>iz1</Var>
              </left>
              <right>
                <External>
                  <content>
                    <Expr>
                      <op>
                        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
                      </op>
                      <args ordered="yes">

```

```

        <Var>y</Var>
        <Var>z1</Var>
      </args>
    </Expr>
  </content>
</External>
</right>
</Equal>
</formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
      </op>
      <args ordered="yes">
        <Var>y</Var>
        <Var>z2</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</formula>
<Equal>
  <left>
    <Var>iz2</Var>
  </left>
  <right>
    <External>
      <content>
        <Expr>
          <op>
            <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
          </op>
          <args ordered="yes">
            <Var>y</Var>
            <Var>z2</Var>
          </args>
        </Expr>
      </content>
    </External>
  </right>
</Equal>
</formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#numeric-not-equal</Const>
      </op>
      <args ordered="yes">
        <Var>iz1</Var>
        <Var>iz2</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</formula>
<Frame>
  <object>
    <Var>z1</Var>
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
    <Var>z2</Var>
  </slot>
</Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
</sentence>
<Forall>
  <declare>
    <Var>x</Var>
  </declare>
  <declare>
    <Var>y</Var>
  </declare>
  <declare>
    <Var>o</Var>
  </declare>
  <declare>
    <Var>v</Var>

```



```

</declare>
<declare>
  <Var>l</Var>
</declare>
<declare>
  <Var>ix</Var>
</declare>
<declare>
  <Var>iy</Var>
</declare>
<declare>
  <Var>r</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>r</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AllDisjointProperties</Const>
            </slot>
          </Frame>
        </formula>
      </And>
      <formula>
        <Frame>
          <object>
            <Var>r</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#members</Const>
            <Var>l</Var>
          </slot>
        </Frame>
      </formula>
    </if>
  </Implies>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>x</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>ix</Var>
    </left>
    <right>
      <External>
        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>l</Var>
              <Var>x</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>iy</Var>
    </left>
    <right>
      <External>

```

```

        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>l</Var>
              <Var>y</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#numeric-not-equal</Const>
        </op>
        <args ordered="yes">
          <Var>ix</Var>
          <Var>iy</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Frame>
    <object>
      <Var>o</Var>
    </object>
    <slot ordered="yes">
      <Var>x</Var>
      <Var>v</Var>
    </slot>
    <slot ordered="yes">
      <Var>y</Var>
      <Var>v</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>o</Var>
    </declare>
    <declare>
      <Var>l</Var>
    </declare>
    <declare>
      <Var>ix</Var>
    </declare>
    <declare>
      <Var>iy</Var>
    </declare>
    <declare>
      <Var>r</Var>
    </declare>
  </forall>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>r</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#AllDisjointClasses</Const>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
</sentence>
</forall>

```

```

<object>
  <Var>r</Var>
</object>
<slot ordered="yes">
  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#members</Const>
  <Var>l</Var>
</slot>
</Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>x</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>ix</Var>
    </left>
    <right>
      <External>
        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>l</Var>
              <Var>x</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Equal>
    <left>
      <Var>iy</Var>
    </left>
    <right>
      <External>
        <content>
          <Expr>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-function#index-of</Const>
            </op>
            <args ordered="yes">
              <Var>l</Var>
              <Var>y</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#numeric-not-equal</Const>
        </op>
        <args ordered="yes">
          <Var>ix</Var>
          <Var>iy</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>

```

```

<formula>
  <Frame>
    <object>
      <Var>o</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>x</Var>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error</Const>
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>
    <declare>
      <Var>last</Var>
    </declare>
    <declare>
      <Var>pc</Var>
    </declare>
    <declare>
      <Var>start</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>p</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#propertyChainAxiom</Const>
                  <Var>pc</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Atom>
                <op>
                  <Const type="&rif;local">checkChain</Const>
                </op>
                <args ordered="yes">
                  <Var>start</Var>
                  <Var>pc</Var>
                  <Var>last</Var>
                </args>
              </Atom>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>start</Var>
    </declare>
    <declare>
      <Var>pc</Var>
    </declare>
    <declare>
      <Var>last</Var>
    </declare>
  </Forall>

```

```

<Var>p</Var>
</declare>
<declare>
<Var>tl</Var>
</declare>
<formula>
<Implies>
<if>
<And>
<formula>
<Frame>
<object>
<Var>pc</Var>
</object>
<slot ordered="yes">
<Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
<Var>p</Var>
</slot>
<slot ordered="yes">
<Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
<Var>tl</Var>
</slot>
</Frame>
</formula>
<formula>
<Frame>
<object>
<Var>start</Var>
</object>
<slot ordered="yes">
<Var>p</Var>
<Var>next</Var>
</slot>
</Frame>
</formula>
<formula>
<Atom>
<op>
<Const type="&rif;local">checkChain</Const>
</op>
<args ordered="yes">
<Var>next</Var>
<Var>tl</Var>
<Var>last</Var>
</args>
</Atom>
</formula>
</And>
</if>
<then>
<Atom>
<op>
<Const type="&rif;local">checkChain</Const>
</op>
<args ordered="yes">
<Var>start</Var>
<Var>pc</Var>
<Var>last</Var>
</args>
</Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
<forall>
<declare>
<Var>start</Var>
</declare>
<declare>
<Var>pc</Var>
</declare>
<declare>
<Var>last</Var>
</declare>
<declare>
<Var>p</Var>
</declare>
<formula>
<Implies>
<if>
<And>
<formula>
<Frame>
<object>
<Var>pc</Var>
</object>
<slot ordered="yes">
<Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
<Var>p</Var>
</slot>
<slot ordered="yes">
<Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
<Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#nil</Const>
</slot>
</Frame>
</formula>

```

```

<formula>
  <Frame>
    <object>
      <Var>start</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>last</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;local">checkChain</Const>
    </op>
    <args ordered="yes">
      <Var>start</Var>
      <Var>pc</Var>
      <Var>last</Var>
    </args>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
  </forall>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasKey</Const>
              <Var>u</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Var>c</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>y</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
              <Var>c</Var>
            </slot>
          </Frame>
        </formula>
      </And>
      <Atom>
        <op>
          <Const type="&rif;local">sameKey</Const>
        </op>
        <args ordered="yes">
          <Var>u</Var>
          <Var>x</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </if>
  </Implies>
</sentence>
</then>
</if>
</then>

```

```

    <Frame>
      <object>
        <Var>x</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
        <Var>y</Var>
      </slot>
    </Frame>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
                  <Var>tl</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Atom>
                <op>
                  <Const type="&rif;local">sameKey</Const>
                </op>
                <args ordered="yes">
                  <Var>tl</Var>
                  <Var>x</Var>
                  <Var>y</Var>
                </args>
              </Atom>
            </formula>
          </And>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;local">sameKey</Const>
            </op>
            <args ordered="yes">
              <Var>u</Var>
              <Var>x</Var>
              <Var>y</Var>
            </args>
          </Atom>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>

```

```

<declare>
  <Var>u</Var>
</declare>
<declare>
  <Var>x</Var>
</declare>
<declare>
  <Var>y</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>u</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
              <Var>key</Var>
            </slot>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#nil</Const>
            </slot>
          </Frame>
        </formula>
      </And>
      <formula>
        <Frame>
          <object>
            <Var>x</Var>
          </object>
          <slot ordered="yes">
            <Var>key</Var>
            <Var>v</Var>
          </slot>
        </Frame>
      </formula>
      <formula>
        <Frame>
          <object>
            <Var>y</Var>
          </object>
          <slot ordered="yes">
            <Var>key</Var>
            <Var>v</Var>
          </slot>
        </Frame>
      </formula>
    </if>
    <then>
      <Atom>
        <op>
          <Const type="&rif;local">sameKey</Const>
        </op>
        <args ordered="yes">
          <Var>u</Var>
          <Var>x</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>l</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#intersectionOf</Const>
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
          </And>
          <Atom>
            <op>
              <Const type="&rif;local">allTypes</Const>
            </op>
          </Atom>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

      </op>
      <args ordered="yes">
        <Var>l</Var>
        <Var>y</Var>
      </args>
    </Atom>
  </formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>l</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>ty</Var>
    </declare>
    <declare>
      <Var>tl</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>l</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
                  <Var>ty</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
                  <Var>tl</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>ty</Var>
                </slot>
              </Frame>
            </formula>
          </formula>
        </if>
      </Implies>
    </formula>
    <Atom>
      <op>
        <Const type="&rif;local">allTypes</Const>
      </op>
      <args ordered="yes">
        <Var>tl</Var>
        <Var>y</Var>
      </args>
    </Atom>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;local">allTypes</Const>
    </op>
    <args ordered="yes">
      <Var>l</Var>
      <Var>y</Var>
    </args>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>

```

```

<declare>
  <Var>l</Var>
</declare>
<declare>
  <Var>y</Var>
</declare>
<declare>
  <Var>ty</Var>
</declare>
<formula>
  <Implies>
    <if>
      <And>
        <formula>
          <Frame>
            <object>
              <Var>l</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
              <Var>ty</Var>
            </slot>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#nil</Const>
            </slot>
          </Frame>
        </formula>
      </And>
      <formula>
        <Frame>
          <object>
            <Var>y</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
            <Var>ty</Var>
          </slot>
        </Frame>
      </formula>
    </if>
    <then>
      <Atom>
        <op>
          <Const type="&rif;local">allTypes</Const>
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>y</Var>
        </args>
      </Atom>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>u</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#hasKey</Const>
                  <Var>u</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

      <object>
        <Var>y</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
        <Var>c</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<Atom>
  <op>
    <Const type="&rif;local">sameKey</Const>
  </op>
  <args ordered="yes">
    <Var>u</Var>
    <Var>x</Var>
    <Var>y</Var>
  </args>
</Atom>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#sameAs</Const>
      <Var>y</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
                  <Var>tl</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
<Atom>
  <op>
    <Const type="&rif;local">sameKey</Const>
  </op>
  <args ordered="yes">
    <Var>tl</Var>
    <Var>x</Var>
  </args>
</Atom>

```

```

        <Var>y</Var>
      </args>
    </Atom>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;local">sameKey</Const>
    </op>
    <args ordered="yes">
      <Var>u</Var>
      <Var>x</Var>
      <Var>y</Var>
    </args>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>u</Var>
    </declare>
    <declare>
      <Var>x</Var>
    </declare>
    <declare>
      <Var>y</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>u</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#first</Const>
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#rest</Const>
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#nil</Const>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>y</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </then>
    <Atom>
      <op>
        <Const type="&rif;local">sameKey</Const>
      </op>
      <args ordered="yes">
        <Var>u</Var>
        <Var>x</Var>
        <Var>y</Var>
      </args>
    </Atom>
  </then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>y</Var>
    </declare>
  </declare>
</declare>

```

```

    <Var>c</Var>
  </declare>
  <declare>
    <Var>l</Var>
  </declare>
  <declare>
    <Var>ci</Var>
  </declare>
  <formula>
    <Implies>
      <if>
        <And>
          <formula>
            <Frame>
              <object>
                <Var>c</Var>
              </object>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/2002/07/owl#union0f</Const>
                <Var>l</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
                  </op>
                  <args ordered="yes">
                    <Var>l</Var>
                    <Var>ci</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>y</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
            <Var>ci</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>yi</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>l</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl#one0f</Const>
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
                    </op>

```

```

      </op>
      <args ordered="yes">
        <Var>l</Var>
        <Var>yi</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Frame>
    <object>
      <Var>yi</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>y</Var>
    </declare>
    <declare>
      <Var>c</Var>
    </declare>
    <declare>
      <Var>ci</Var>
    </declare>
    <declare>
      <Var>l</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>c</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2002/07/owl#intersectionOf</Const>
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rf;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</Const>
                    </op>
                    <args ordered="yes">
                      <Var>l</Var>
                      <Var>ci</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rf;iri">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</forall>
</sentence>

```

```

<sentence>
  <forall>
    <declare>
      <var>c</var>
    </declare>
    <declare>
      <var>ci</var>
    </declare>
    <declare>
      <var>l</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>c</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2002/07/owl#intersectionOf</const>
                  <var>l</var>
                </slot>
              </frame>
            </formula>
          </and>
          <formula>
            <external>
              <content>
                <atom>
                  <op>
                    <const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</const>
                  </op>
                  <args ordered="yes">
                    <var>l</var>
                    <var>ci</var>
                  </args>
                </atom>
              </content>
            </external>
          </formula>
        </if>
        <then>
          <frame>
            <object>
              <var>c</var>
            </object>
            <slot ordered="yes">
              <const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</const>
              <var>ci</var>
            </slot>
          </frame>
        </then>
      </implies>
    </formula>
  </forall>
</sentence>
<sentence>
  <forall>
    <declare>
      <var>c</var>
    </declare>
    <declare>
      <var>ci</var>
    </declare>
    <declare>
      <var>l</var>
    </declare>
    <formula>
      <implies>
        <if>
          <and>
            <formula>
              <frame>
                <object>
                  <var>c</var>
                </object>
                <slot ordered="yes">
                  <const type="&rif;iri">http://www.w3.org/2002/07/owl#unionOf</const>
                  <var>l</var>
                </slot>
              </frame>
            </formula>
          </and>
          <formula>
            <external>
              <content>
                <atom>
                  <op>
                    <const type="&rif;iri">http://www.w3.org/2007/rif-builtin-predicate#list-contains</const>
                  </op>
                  <args ordered="yes">
                    <var>l</var>
                    <var>ci</var>
                  </args>
                </atom>
              </content>
            </external>
          </formula>
        </if>
      </implies>
    </formula>
  </forall>
</sentence>

```

```

    </formula>
  </And>
</if>
<then>
  <Frame>
    <object>
      <Var>ci</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#subClassOf</Const>
      <Var>c</Var>
    </slot>
  </Frame>
</then>
</Implies>
</formula>
</Forall>
</sentence>
</Group>
</payload>
</Document>

```

10 Appendix: Change Log (Informative)

Changes from first Note publication <http://www.w3.org/TR/2010/NOTE-rif-owl-rl-20100622/>

- Corrected reference to `pred:literal-not-identical`
- Corrected template rules in appendix 8: `prp-adp`, `cax-adc`, `prp-spo2`, `prp-key`, `scm-int`, `cls-maxqc2`, `eq-diff2` and `eq-diff3`. Many thanks to Hendrick Marx for identifying the problems.

Changes from the first [Working Draft of 1st October 2009](#).

- Fixed errors in rules `eq-diff2`, `prp-adp`, `cax-adc`; added missing rule `eq-diff3`. Other small syntactic fixes to `prp-key`.
- Updated and corrected list of datatypes.
- Updated to use RIF List operators for handling of RDF lists.
- Fixed syntax errors in rule names (missing angle brackets in relative URIs)
- Fixed use of `Forall` with empty variable list