



## OWL 2 RL in RIF

W3C Editor's Draft 23 April 2010

**This version:**

<http://www.w3.org/2005/rules/wg/draft/ED-rif-owl-rl-20100423/>

**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/ED-rif-owl-rl-20091001/> ([color-coded diff](#))

**Editors:**

Dave Reynolds, Hewlett Packard Laboratories

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

[Copyright](#) © 2010 [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 6) and via a mapping algorithm which converts an OWL 2 RL ontology to a customized RIF Core rule set (Appendix 7).

## 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### May Be Superseded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 10 documents:

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

## XML Schema Datatypes Dependency

RIF is defined to use datatypes defined in the [XML Schema Definition Language \(XSD\)](#). As of this writing, the latest W3C Recommendation for XSD is version 1.0, with [version 1.1](#) progressing toward Recommendation. RIF has been designed to take advantage of the new datatypes and clearer explanations available in XSD 1.1, but for now those advantages are being partially put on hold. Specifically, until XSD 1.1 becomes a W3C Recommendation, the elements of RIF which are based on it should be considered *optional*, as detailed in [Datatypes and Builtins, section 2.3](#). Upon the publication of XSD 1.1 as a W3C Recommendation, those elements will cease to be optional and are to be considered required as otherwise specified.

We suggest that for now developers and users follow the [XSD 1.1 Last Call Working Draft](#). Based on discussions between the Schema, RIF and OWL Working Groups, we do not expect any implementation changes will be necessary as XSD 1.1 advances to Recommendation.

## 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 Comment By 21 May 2010

The [Rule Interchange Format \(RIF\) Working Group](#) seeks public feedback on this Editor's Draft. Please send your comments to [public-rif-comments@w3.org](mailto:public-rif-comments@w3.org) ([public archive](#)). If possible, please offer specific changes to the text that would address your concern. You may also wish to check the [Wiki Version](#) of this document and see if the relevant text has already been updated.

## No Endorsement

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

## Patents

*This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). 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](#)
    - [4.3.5 Simple member rules](#)
  - [4.4 Datatype rules](#)
    - [4.4.1 Datatypes supported](#)
    - [4.4.2 Datatype rules and safety](#)
    - [4.4.3 Rules for supported datatypes](#)
- [5 References](#)
- [6 Appendix: OWL 2 RL ruleset - presentation syntax](#)
- [7 Appendix: OWL 2 RL to RIF translation](#)
  - [7.1 FixedRules](#)
  - [7.2 templateRules algorithm](#)
- [8 Appendix: OWL 2 RL ruleset - XML syntax](#)
- [9 Appendix: Change Log \(Informative\)](#)

## 1 Introduction

OWL 2 RL is a syntactic subset (*profile*) of OWL 2 that is amenable to implementation using rule-based technologies. The [\[OWL2-Profiles\]](#) document provides a partial axiomatization of the OWL 2 RDF-Based Semantics in the form of first-order implications that can be used as the basis for such an implementation.

This note describes how that axiomatization can be expressed as a set of RIF rules within the RIF Core dialect.

There are two approaches to this, both of which exploit the correspondence between RIF and RDF given in [\[RIF RDF and OWL Compatibility\]](#). The first approach is to directly translate the OWL 2 RL rules to RIF Core, those simply interested in this ruleset can skip directly to [\[Appendix: OWL 2 RL Rules\]](#). The second approach, which is expected to be more scalable in practice, is to translate the source OWL 2 RL ontology to a specialized RIF Core rule set, see [\[Appendix: OWL 2 RL to RIF translation\]](#).

## 2 Summary of issues

The rules for the OWL 2 RL semantics can be divided into four (non-disjoint) categories of rules.

**Triple pattern rules** derive one or more RDF triples from a conjunction of RDF triple patterns. Translation of these to RIF by means of Frame formulae is trivial.

**List rules** involve processing of RDF lists in the input graph and are expressed in OWL 2 RL using an informal elipsis notation. In implementing OWL 2 RL in RIF there are two ways of handling such rules. They can either be expressed as a set of recursive rules that traverse the RDF list structures *at run-time* or they can be regarded as templates that, for a given RDF input document, can be translated to a set of triple pattern rules as part of a preprocessing step. We define both options in this note.

**Inconsistency rules** indicate inconsistencies in the original RDF graph, they are expressed as first order rules which derive a propositional symbol *false*. We suggest a predicate symbol within the RIF namespace be used for this purpose.

**Datatype rules** these provide type checking and value equality/inequality checking for typed literals across a set of supported datatypes. The set of datatypes supported by RIF and OWL align and translation of the rules to RIF is largely straightforward.

Putting these together we conclude that implementation of OWL 2 RL rule-based inference via RIF rules is possible. We exhibit both a fixed rule set and a translation algorithm for generating an ontology-specific ruleset.

### 3 Background OWL 2 RL

The OWL 2 RL profile [[OWL2-Profiles](#)] is aimed at applications that require scalable reasoning in return for some restriction on expressive power. It defines a syntactic subset of OWL 2 that is amenable to implementation using rule-based technologies together with a partial axiomatization of the OWL 2 RDF-Based Semantics in the form of first-order implications that can be used as the basis for such an implementation.

For ontologies satisfying the syntactic constraints of the OWL 2 RL profile then a suitable rule-based implementation can provide sound and complete entailment checking and conjunctive query answering (so long as the query also meets the OWL 2 RL syntactic restrictions).

The purpose of this note is to provide a translation of the OWL 2 RL rules into RIF notation. The aim is that this translation be faithful to the OWL 2 RL semantics so that the follow theorem should hold:

**Theorem 1.** Let  $O_1$  and  $O_2$  be OWL 2 RL ontologies satisfying the following properties:

- neither  $O_1$  nor  $O_2$  contains a IRI that is used for more than one type of entity (i.e., no IRIs is used both as, say, a class and an individual);
- $O_1$  does not contain `SubAnnotationPropertyOf`, `AnnotationPropertyDomain`, and `AnnotationPropertyRange` axioms; and
- each axiom in  $O_2$  is an assertion of the form as specified below, for  $a, a_1, \dots, a_n$  named individuals:
  - `ClassAssertion( C a )` where  $C$  is a class,
  - `ObjectPropertyAssertion( OP a1 a2 )` where  $OP$  is an object property,
  - `DataPropertyAssertion( DP a v )` where  $DP$  is a data property, or
  - `SameIndividual( a1 ... an )`.

Let  $R(O_1)$  be either the fixed RIF rule set described in [[Appendix: OWL 2 RL ruleset - presentation syntax](#)] or the RIF rule set derived from  $O_1$  by the algorithm described in [[Appendix: OWL 2 RL to RIF translation](#)].

Furthermore, let  $RDF(O_1)$  and  $RDF(O_2)$  be translations of  $O_1$  and  $O_2$ , respectively, into RDF graphs as specified in the OWL 2 Mapping to RDF Graphs [[OWL2-RDF-Mapping](#)].

Then,  $O_1$  entails  $O_2$  under the OWL 2 RDF-Based semantics [[OWL2-RDF-Semantics](#)] if and only if the RIF-RDF combination of  $R(O_1)$  and  $RDF(O_1)$  entails the generalized RDF graph  $RDF(O_2)$  according to the notion of *simple entailment* for RIF-RDF combinations as defined in [[RIF RDF and OWL Compatibility](#)].

## 4 Analysis of OWL 2 RL rules

As noted above the OWL 2 RL rules fall into four categories which pose different issues for translation to RIF. We discuss each category in turn.

Note that throughout this section all rules given in RIF Core presentation syntax will assume the following prefix definitions:

```
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/>)
  ...
)
```

### 4.1 Triple pattern rules

The triple pattern rules take the form:

if	then
T( <i>s</i> <sub>1</sub> , <i>p</i> <sub>1</sub> , <i>o</i> <sub>1</sub> )	T( <i>sr</i> <sub>1</sub> , <i>pr</i> <sub>1</sub> , <i>or</i> <sub>1</sub> )
...	...
T( <i>s</i> <sub><i>n</i></sub> , <i>p</i> <sub><i>n</i></sub> , <i>o</i> <sub><i>n</i></sub> )	T( <i>sr</i> <sub><i>m</i></sub> , <i>pr</i> <sub><i>m</i></sub> , <i>or</i> <sub><i>m</i></sub> )

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 ?v11 ... ?v1o (
    sr1[pr1->or1] :- And( s1[p1->o1] ... sn[pn->on] ) )
    ...
  Forall ?vm1 ... ?vmo (
    srm[prm->orm] :- And( s1[p1->o1] ... sn[pn->on] ) )
)
```

where ?v<sub>11</sub> ... ?v<sub>1<sub>o</sub></sub> 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
<code>T(?p<sub>1</sub>, owl:propertyDisjointWith, ?p<sub>2</sub>)</code> <code>T(?x, ?p<sub>1</sub>, ?y)</code> <code>T(?x, ?p<sub>2</sub>, ?y)</code>	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 <sub>1</sub> , ..., ?z <sub>n</sub> ] T(?z <sub>i</sub> , owl:sameAs, ?z <sub>j</sub> )	false	for each 1 ≤ i < j ≤ n
prp-adp	T(?x, rdf:type, owl:AllDisjointProperties) T(?x, owl:members, ?y) LIST[?y, ?p <sub>1</sub> , ..., ?p <sub>n</sub> ] T(?u, ?p <sub>i</sub> , ?v) T(?u, ?p <sub>j</sub> , ?v)	false	for each 1 ≤ i < j ≤ n
cax-adc	T(?x, rdf:type, owl:AllDisjointClasses) T(?x, owl:members, ?y) LIST[?y, ?c <sub>1</sub> , ..., ?c <sub>n</sub> ] T(?z, rdf:type, ?c <sub>i</sub> ) T(?z, rdf:type, ?c <sub>j</sub> )	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-o
    External(pred:list-contains(?y ?z2))    ?iz2 = External(func:index-o
    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(
  External(pred:list-contains(?y ?z2))    ?iz2 = External(func:index-of(
  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(
    External(pred:list-contains(?l ?y))    ?iy = External(func:index-of(
    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(
    External(pred:list-contains(?l ?y))    ?iy = External(func:index-of(
    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	<pre> T(?p,  owl:propertyChainAxiom, ?x)  LIST[?x, ?p1, ..., ?pn]  T(?u1, ?p1, ?u2)  T(?u2, ?p2 ?u3)  ...  T(?un, ?pn, ?un+1) </pre>	T(?u1, ?p, ?un+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 ?t1 ?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->?t1]
    ?x[?key->?v] ?y[?key->?v]

```

```

    _sameKey(?t1 ?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>	<pre> T(?y,  rdf:type, ?c) </pre>
----------	---	-----------------------------------

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 ?t1 (
  _allTypes(?l ?y) :- And (
    ?l[rdf:first->?ty rdf:rest->?t1]
    ?y[rdf:type->?ty]
    _allTypes(?t1 ?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	T(?c, owl:unionOf, ?x) LIST[?x, ?c <sub>1</sub> , ..., ?c <sub>n</sub> ] T(?y, rdf:type, ?c <sub>i</sub> )	T(?y, rdf:type, ?c)	for each 1 ≤ i ≤ n
cls-oo	T(?c, owl:oneOf, ?x) LIST[?x, ?y <sub>1</sub> , ..., ?y <sub>n</sub> ]	T(?y <sub>i</sub> , rdf:type, ?c)	for each 1 ≤ i ≤ n
cls-int2	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c <sub>1</sub> , ..., ?c <sub>n</sub> ] T(?y, rdf:type, ?c)	T(?y, rdf:type, ?c <sub>1</sub> ) T(?y, rdf:type, ?c <sub>2</sub> ) ... T(?y, rdf:type, ?c <sub>n</sub> )	
scm-int	T(?c, owl:intersectionOf, ?x) LIST[?x, ?c <sub>1</sub> , ..., ?c <sub>n</sub> ]	T(?c, rdfs:subClassOf, ?c <sub>1</sub> ) T(?c, rdfs:subClassOf, ?c <sub>2</sub> ) ... T(?c, rdfs:subClassOf, ?c <sub>n</sub> )	
scm-uni	T(?c, owl:unionOf, ?x) LIST[?x, ?c <sub>1</sub> , ..., ?c <sub>n</sub> ]	T(?c <sub>1</sub> , rdfs:subClassOf, ?c) T(?c <sub>2</sub> , rdfs:subClassOf, ?c) ... T(?c <sub>n</sub> , 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(lt <sub>1</sub> , owl:sameAs, lt <sub>2</sub> )	for all literals lt <sub>1</sub> and lt <sub>2</sub> with the same data value
dt-diff	true	T(lt <sub>1</sub> , owl:differentFrom, lt <sub>2</sub> )	for all literals lt <sub>1</sub> and lt <sub>2</sub> with different data values

dt-not-type	T( <i>lt</i> , rdf:type, <i>dt</i> )	false	for each literal <i>lt</i> and each datatype <i>dt</i> supported in OWL 2 RL such that the data value of <i>lt</i> is not contained in the value space of <i>dt</i>
-------------	--	-------	--

#### 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-PlainLit
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral
  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-no
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:nonPositiveInteger] External(pred:is-literal-not-no
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:positiveInteger] External(pred:is-literal-not-posit
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:negativeInteger] External(pred:is-literal-not-negat
  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-unsigned
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:unsignedInt] External(pred:is-literal-not-unsignedI
  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[rdف:type->xsd:dateTimeStamp ] External(pred:is-literal-not-dateTi
)

```

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[rdف:type->rdf:PlainLiteral] External(pred:is-literal-not-PlainLit
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:decimal] External(pred:is-literal-not-decimal( ?lt
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:integer] External(pred:is-literal-not-integer( ?lt
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:nonNegativeInteger] External(pred:is-literal-not-no
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:nonPositiveInteger] External(pred:is-literal-not-no
    Forall ?lt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:positiveInteger] External(pred:is-literal-not-posit
    Forall ?lt ?dt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:negativeInteger] External(pred:is-literal-not-negat
    Forall ?lt ?dt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:long] External(pred:is-literal-not-long( ?lt )) ))
    Forall ?lt ?dt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:int] External(pred:is-literal-not-int( ?lt )) ))
    Forall ?lt ?dt (
      rif:error() :- And (
        ?lt[rdف:type->xsd:short] External(pred:is-literal-not-short( ?lt )) ))

```

```

Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:byte] External(pred:is-literal-not-byte( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedLong] External(pred:is-literal-not-unsigned
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedInt] External(pred:is-literal-not-unsignedI
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedShort] External(pred:is-literal-not-unsigned
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedByte] External(pred:is-literal-not-unsigned
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:float] External(pred:is-literal-not-float( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:double] External(pred:is-literal-not-double( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:string] External(pred:is-literal-not-string( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:normalizedString] External(pred:is-literal-not-norm
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:token] External(pred:is-literal-not-token( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:language] External(pred:is-literal-not-language( ?l
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:Name] External(pred:is-literal-not-Name( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:NCName] External(pred:is-literal-not-NCName( ?lt )) )
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:NMTOKEN] External(pred:is-literal-not-NMTOKEN( ?lt
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:boolean] External(pred:is-literal-not-boolean( ?lt
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:hexBinary] External(pred:is-literal-not-hexBinary(

```

```

Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:base64Binary] External(pred:is-literal-not-base64Bi
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:anyURI] External(pred:is-literal-not-anyURI( ?lt ))
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:dateTime] External(pred:is-literal-not-dateTime( ?l
Forall ?lt ?dt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:dateTimeStamp ] External(pred:is-literal-not-dateTi
)

```

**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]

```

## 5 References

### [OWL2-Profiles]

[OWL 2 Web Ontology Language: Profiles](#), 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/CR-owl2-profiles-20090611/>. Latest version available at <http://www.w3.org/TR/owl2-profiles/>.

#### [OWL2-RDF-Mapping]

*OWL 2 Web Ontology Language Mapping to RDF Graphs*, 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*, 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* Jos de Bruijn, editor. W3C Editor's Draft, 23 April 2010, <http://www.w3.org/2005/rules/wg/draft/ED-rif-rdf-owl-20100423/>. Latest version available at <http://www.w3.org/2005/rules/wg/draft/rif-rdf-owl/>.

## 6 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[rdf:type->?x]
    ?u[?p->?y]
    ?y[rdf:type->?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[rdf:type->?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[rdf:type->?x]
        ?u[?p->?y1]
        ?y1[rdf:type->?c]
        ?u[?p->?y2]
        ?y2[rdf:type->?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[rdf:type->?x]
    ?u[?p->?y1]
    ?u[?p->?y2]  ))

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

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

(* <#cax-ecq2> *)
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[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-ecq2> *)
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
    External(pred:list-contains(?y ?z2))    ?iz2 = External(func:index-of
    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
    External(pred:list-contains(?y ?z2))    ?iz2 = External(func:index-of
    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(?
    External(pred:list-contains(?l ?y))    ?iy = External(func:index-of(?
    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(?
    External(pred:list-contains(?l ?y))    ?iy = External(func:index-of(?
    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[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] ))

Forall ?x ?y ?c ?u ?c (
  ?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] ))

(* <#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] ))

(* <#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] ))

(* <#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)) ))

```

### 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-PlainLit
      Forall ?s ?p ?lt ( ?lt[rdf:type->rdf:XMLLiteral rdf:type->rdfs:Literal]
        :- And( ?s[?p->?lt] External( pred:is-literal-XMLLiter
      Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:decimal rdf:type->rdfs:Literal]
        :- And( ?s[?p->?lt] External( pred:is-literal-decimal(
      Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:integer rdf:type->rdfs:Literal]
        :- And( ?s[?p->?lt] External( pred:is-literal-integer(
      Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonNegativeInteger rdf:type->rdfs:
        :- And( ?s[?p->?lt] External( pred:is-literal-nonNegat
      Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonPositiveInteger rdf:type->rdfs:

```

W3C Editor's Draft

```

:- And( ?s[?p->?lt] External( pred:is-literal-nonPosit
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:positiveInteger rdf:type->rdfs:Lit
:- And( ?s[?p->?lt] External( pred:is-literal-positive
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:negativeInteger rdf:type->rdfs:Lit
:- And( ?s[?p->?lt] External( pred:is-literal-negative
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:long rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-long( ?l
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( ?
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:byte rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-byte( ?l
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedLong rdf:type->rdfs:Litera
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedInt rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedShort rdf:type->rdfs:Liter
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedByte rdf:type->rdfs:Litera
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:float rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-float( ?
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:double rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-double(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:string rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-string(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:normalizedString rdf:type->rdfs:Li
:- And( ?s[?p->?lt] External( pred:is-literal-normaliz
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:token rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-token( ?
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:language rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-language
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:Name rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-Name( ?l
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NCName rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NCName(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:NMTOKEN rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NMTOKEN(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:boolean rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-boolean(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:hexBinary rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-hexBinar
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:base64Binary rdf:type->rdfs:Litera
:- And( ?s[?p->?lt] External( pred:is-literal-base64Bi
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:anyURI rdf:type->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-anyURI(
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTime rdf:type->rdfs:Literal]

```

```

                                :- And( ?s[?p->?lt] External( pred:is-literal-dateTime
Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:dateTimeStamp rdf:type->rdfs:Lite
                                :- And( ?s[?p->?lt] External( pred:is-literal-dateTime
)
(* <#dt-not-type> *)
Group (
  Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->rdf:PlainLiteral] External(pred:is-literal-not-PlainLite
Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral(
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-non
Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:nonPositiveInteger] External(pred:is-literal-not-non
Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:positiveInteger] External(pred:is-literal-not-positi
Forall ?lt (
    rif:error() :- And (
      ?lt[rdf:type->xsd:negativeInteger] External(pred:is-literal-not-negati
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-unsignedL
Forall ?lt (

```

```

rif:error() :- And (
  ?lt[rdf:type->xsd:unsignedInt] External(pred:is-literal-not-unsignedInt)
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedShort] External(pred:is-literal-not-unsignedShort)
Forall ?lt (
  rif:error() :- And (
    ?lt[rdf:type->xsd:unsignedByte] External(pred:is-literal-not-unsignedByte)
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-dateTim
)

(* <#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:literalNotIdentical(?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-typel-PlainLiteral> *) rdf:PlainLiteral[rdf:type -> rdfs:Datatype]
(* <#dt-typel-decimal> *) xsd:decimal[rdf:type -> rdfs:Datatype]
(* <#dt-typel-integer> *) xsd:integer[rdf:type -> rdfs:Datatype]
(* <#dt-typel-double> *) xsd:double[rdf:type -> rdfs:Datatype]
(* <#dt-typel-string> *) xsd:string[rdf:type -> rdfs:Datatype]
(* <#dt-typel-dateTime> *) xsd:dateTime[rdf:type -> rdfs:Datatype]
(* <#dt-typel-XMLLiteral> *) rdf:XMLLiteral[rdf:type -> rdfs:Datatype]
(* <#dt-typel-Literal> *) rdfs:Literal[rdf:type -> rdfs:Datatype]

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

))

```

## 7 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.

### 7.1 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-ecq1> *)
Forall ?x ?c1 ?c2 (
  ?x[rdf:type->?c2] :- And(
    ?c1[owl:equivalentClass->?c2]
    ?x[rdf:type->?c1]  ))

(* <#cax-ecq2> *)
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-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]  ))

(* <#dt-type2> *)
Group (
  Forall ?s ?p ?lt ( ?lt[rdf:type->rdf:PlainLiteral rdf:type->rdfs:Literal
    :- And( ?s[?p->?lt] External( pred:is-literal-PlainLit
  Forall ?s ?p ?lt ( ?lt[rdf:type->rdf:XMLLiteral rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-XMLLiter
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:decimal rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-decimal(
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:integer rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-integer(
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonNegativeInteger rdf:type->rdfs:
    :- And( ?s[?p->?lt] External( pred:is-literal-nonNegat
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:nonPositiveInteger rdf:type->rdfs:
    :- And( ?s[?p->?lt] External( pred:is-literal-nonPosit
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:positiveInteger rdf:type->rdfs:Lit
    :- And( ?s[?p->?lt] External( pred:is-literal-positive
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:negativeInteger rdf:type->rdfs:Lit
    :- And( ?s[?p->?lt] External( pred:is-literal-negative
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:long rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-long( ?l
  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( ?
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:byte rdf:type->rdfs:Literal]
    :- And( ?s[?p->?lt] External( pred:is-literal-byte( ?l
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedLong rdf:type->rdfs:Literal
    :- And( ?s[?p->?lt] External( pred:is-literal-unsigned
  Forall ?s ?p ?lt ( ?lt[rdf:type->xsd:unsignedInt rdf:type->rdfs:Literal
    :- And( ?s[?p->?lt] External( pred:is-literal-unsigned

```

W3C Editor's Draft

```

Forall ?s ?p ?lt ( ?lt[rdftype->xsd:unsignedShort rdftype->rdfs:Literal
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:unsignedByte rdftype->rdfs:Literal
:- And( ?s[?p->?lt] External( pred:is-literal-unsigned
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:float rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-float( ?
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:double rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-double(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:string rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-string(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:normalizedString rdftype->rdfs:Li
:- And( ?s[?p->?lt] External( pred:is-literal-normaliz
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:token rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-token( ?
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:language rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-language
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:Name rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-Name( ?l
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:NCName rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NCName(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:NMTOKEN rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-NMTOKEN(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:boolean rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-boolean(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:hexBinary rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-hexBinar
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:base64Binary rdftype->rdfs:Literal
:- And( ?s[?p->?lt] External( pred:is-literal-base64Bi
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:anyURI rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-anyURI(
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:dateTime rdftype->rdfs:Literal]
:- And( ?s[?p->?lt] External( pred:is-literal-dateTime
Forall ?s ?p ?lt ( ?lt[rdftype->xsd:dateTimeStamp rdftype->rdfs:Lite
:- And( ?s[?p->?lt] External( pred:is-literal-dateTime
)

(* <#dt-not-type> *)
Group (

Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->rdf:PlainLiteral] External(pred:is-literal-not-PlainLite
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->rdf:XMLLiteral] External(pred:is-literal-not-XMLLiteral(
Forall ?lt (
  rif:error() :- And (
    ?lt[rdftype->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-dateTime(?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:literalNotIdentical(?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-typel-PlainLiteral> *) rdf:PlainLiteral[rdf:type -> rdfs:Datatype]
(* <#dt-typel-decimal> *) xsd:decimal[rdf:type -> rdfs:Datatype]
(* <#dt-typel-integer> *) xsd:integer[rdf:type -> rdfs:Datatype]
(* <#dt-typel-double> *) xsd:double[rdf:type -> rdfs:Datatype]
(* <#dt-typel-string> *) xsd:string[rdf:type -> rdfs:Datatype]
(* <#dt-typel-dateTime> *) xsd:dateTime[rdf:type -> rdfs:Datatype]
(* <#dt-typel-XMLLiteral> *) rdf:XMLLiteral[rdf:type -> rdfs:Datatype]
(* <#dt-typel-Literal> *) rdfs:Literal[rdf:type -> rdfs:Datatype]

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

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

```

## 7.2 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)	<pre>Forall ?x ?y (   ?y[?p-&gt;?x] :- And(     ?x[?p-&gt;?y]  ) )</pre>

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
{ <i>\$ rule text \$</i> }	Emit the <i>rule text</i> substituting any occurrence of variables from the binding map. The outer { <i>\$ \$</i> } can be omitted in cases where there is no ambiguity.

<pre>for(?elt in ?list) {   template }</pre>	<p>?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 <i>?list</i> 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.</p>
<pre>for(?elt in ?list, != ?other) {   template }</pre>	<p>?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 <i>?list</i> in turn, skipping any element identical to <i>?other</i>, 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.</p>
<pre>\$length(?list)\$</pre>	<p>Used within a template this will be replaced by the length the RDFList bound to <i>?list</i>.</p>
<pre>\$i\$</pre>	<p>Used within a <i>for(?elt in ?list){ template }</i> this will be replaced by the index of the current ?elt.</p>
<pre>\$i+1\$</pre>	<p>Used within a <i>for(?elt in ?list){ template }</i> this will be replaced by the index of the current ?elt, plus 1.</p>

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

Pattern	Template
---------	----------

<p>(?p rdf:type owl:FunctionalProperty)</p>	<pre>(* &lt;#prp-fp&gt; *) Forall ?y2 ?x ?y1 (   ?y1[owl:sameAs-&gt;?y2] :- And(     ?x[?p-&gt;?y1]     ?x[?p-&gt;?y2] ))</pre>
<p>(?p rdf:type owl:InverseFunctionalProperty))</p>	<pre>(* &lt;#prp-ifp&gt; *) Forall ?x1 ?x2 ?y (   ?x1[owl:sameAs-&gt;?x2] :- And(     ?x1[?p-&gt;?y]     ?x2[?p-&gt;?y] ))</pre>
<p>(?p rdf:type owl:IrreflexiveProperty)</p>	<pre>(* &lt;#prp-irp&gt; *) Forall ?x (   rif:error() :- And(     ?x[?p-&gt;?x] ))</pre>
<p>(?p rdf:type owl:SymmetricProperty)</p>	<pre>(* &lt;#prp-symp&gt; *) Forall ?x ?y (   ?y[?p-&gt;?x] :- And(     ?x[?p-&gt;?y] ))</pre>
<p>(?p rdf:type owl:AsymmetricProperty)</p>	<pre>(* &lt;#prp-asymp&gt; *) Forall ?x ?y (   rif:error() :- And(     ?x[?p-&gt;?y]     ?y[?p-&gt;?x] ))</pre>
<p>(?p rdf:type owl:TransitiveProperty)</p>	<pre>(* &lt;#prp-trp&gt; *) Forall ?x ?z ?y (   ?x[?p-&gt;?z] :- And(     ?x[?p-&gt;?y]     ?y[?p-&gt;?z] ))</pre>
<p>(?p1 rdfs:subPropertyOf ?p2)</p>	<pre>(* &lt;#prp-spo1&gt; *) Forall ?x ?y (</pre>

	<pre>?x[?p2-&gt;?y] :- And(     ?x[?p1-&gt;?y] )</pre>
(?p1 owl:equivalentProperty ?p2)	<pre>(* &lt;#prp-eqp1&gt; *) Forall ?x ?y (     ?x[?p2-&gt;?y] :- And(         ?x[?p1-&gt;?y] ) )  (* &lt;#prp-eqp2&gt; *) Forall ?x ?y (     ?x[?p1-&gt;?y] :- And(         ?x[?p2-&gt;?y] ) )</pre>
(?p1 owl:propertyDisjointWith ?p2)	<pre>(* &lt;#prp-pdw&gt; *) Forall ?x ?y (     rif:error() :- And(         ?x[?p1-&gt;?y]         ?x[?p2-&gt;?y] ) )</pre>
(?p1 owl:inverseOf ?p2)	<pre>(* &lt;#prp-inv1&gt; *) Forall ?x ?y (     ?y[?p2-&gt;?x] :- And(         ?x[?p1-&gt;?y] ) )  (* &lt;#prp-inv2&gt; *) Forall ?x ?y (     ?y[?p1-&gt;?x] :- And(         ?x[?p2-&gt;?y] ) )</pre>
(?x owl:someValuesFrom ?y) (?x owl:onProperty ?p)	<pre>(* &lt;#cls-svf1&gt; *) Forall ?v ?u (     ?u[rdf:type-&gt;?x] :- And(         ?u[?p-&gt;?v]         ?v[rdf:type-&gt;?y] ) )</pre>
(?x owl:someValuesFrom owl:Thing) (?x owl:onProperty ?p)	<pre>(* &lt;#cls-svf2&gt; *) Forall ?v ?u (     ?u[rdf:type-&gt;?x] :- And(         ?u[?p-&gt;?v] ) )</pre>

<p>(?x owl:allValuesFrom ?y) (?x owl:onProperty ?p)</p>	<pre>(* &lt;#cls-avf&gt; *) Forall ?v ?u (   ?v[rdf:type-&gt;?y] :- And(     ?u[rdf:type-&gt;?x]     ?u[?p-&gt;?v] ))</pre>
<p>(?x owl:hasValue ?y) (?x owl:onProperty ?p)</p>	<pre>(* &lt;#cls-hv1&gt; *) Forall ?u (   ?u[?p-&gt;?y] :- And(     ?u[rdf:type-&gt;?x] ))  (* &lt;#cls-hv2&gt; *) Forall ?u (   ?u[rdf:type-&gt;?x] :- And(     ?u[?p-&gt;?y] ))</pre>
<p>(?x owl:maxCardinality 0) (?x owl:onProperty ?p)</p>	<pre>(* &lt;#cls-maxc1&gt; *) Forall ?u ?y (   rif:error() :- And(     ?u[?p-&gt;?y]     ?u[rdf:type-&gt;?x] ))</pre>
<p>(?x owl:maxCardinality 1) (?x owl:onProperty ?p)</p>	<pre>(* &lt;#cls-maxc2&gt; *) Forall ?y2 ?u ?y1 (   ?y1[owl:sameAs-&gt;?y2] :- And(     ?u[?p-&gt;?y1]     ?u[?p-&gt;?y2]     ?u[rdf:type-&gt;?x] ))</pre>
<p>(?x owl:maxQualifiedCardinality 0) (?x owl:onProperty ?p) (?x owl:onClass ?c)</p>	<pre>(* &lt;#cls-maxqc1&gt; *) Forall ?u ?y (   rif:error() :- And(     ?u[rdf:type-&gt;?x]     ?u[?p-&gt;?y]     ?y[rdf:type-&gt;?c] ))</pre>
<p>(?x owl:maxQualifiedCardinality 0) (?x owl:onProperty ?p) (?x owl:onClass owl:Thing)</p>	<pre>(* &lt;#cls-maxqc2&gt; *) Forall ?y (   rif:error() :- And(</pre>

	<pre>?u[rdf:type-&gt;?x] ?u[?p-&gt;?y]  ) )</pre>
<pre>(?x owl:maxQualifiedCardinality 1) (?x owl:onProperty ?p) (?x owl:onClass ?c)</pre>	<pre>(* &lt;#cls-maxqc3&gt; *) Forall ?y2 ?u ?y1 (   ?y1[owl:sameAs-&gt;?y2] :- And(     ?u[rdf:type-&gt;?x]     ?u[?p-&gt;?y1]     ?y1[rdf:type-&gt;?c]     ?u[?p-&gt;?y2]     ?y2[rdf:type-&gt;?c]  ) )</pre>
<pre>(?x owl:maxQualifiedCardinality 1) (?x owl:onProperty ?p) (?x owl:onClass owl:Thing)</pre>	<pre>(* &lt;#cls-maxqc4&gt; *) Forall ?y2 ?u ?y1 (   ?y1[owl:sameAs-&gt;?y2] :- And(     ?u[rdf:type-&gt;?x]     ?u[?p-&gt;?y1]     ?u[?p-&gt;?y2]  ) )</pre>
<pre>(?c1 owl:hasValue ?i) (?c1 owl:onProperty ?p1) (?c2 owl:hasValue ?i) (?c2 owl:onProperty ?p2)</pre>	<pre>(* &lt;#scm-hv&gt; *) ?c1[rdfs:subClassOf-&gt;?c2] :- And(   ?p1[rdfs:subPropertyOf-&gt;?p2]  )</pre>
<pre>(?c1 owl:someValuesFrom ?y1) (?c1 owl:onProperty ?p) (?c2 owl:someValuesFrom ?y2) (?c2 owl:onProperty ?p)</pre>	<pre>(* &lt;#scm-svf1&gt; *) ?c1[rdfs:subClassOf-&gt;?c2] :- And(   ?y1[rdfs:subClassOf-&gt;?y2]  )</pre>
<pre>(?c1 owl:someValuesFrom ?y) (?c1 owl:onProperty ?p1) (?c2 owl:someValuesFrom ?y) (?c2 owl:onProperty ?p2)</pre>	<pre>(* &lt;#scm-svf2&gt; *) ?c1[rdfs:subClassOf-&gt;?c2] :- And(   ?p1[rdfs:subPropertyOf-&gt;?p2]  )</pre>
<pre>(?c1 owl:allValuesFrom ?y1) (?c1 owl:onProperty ?p) (?c2 owl:allValuesFrom ?y2) (?c2 owl:onProperty ?p)</pre>	<pre>(* &lt;#scm-avf1&gt; *) ?c1[rdfs:subClassOf-&gt;?c2] :- And(   ?y1[rdfs:subClassOf-&gt;?y2]  )</pre>

<p>(?c1 owl:allValuesFrom ?y)                  (?c1 owl:onProperty ?p1)                  (?c2 owl:allValuesFrom ?y)                  (?c2 owl:onProperty ?p2)</p>	<pre>(* &lt;#scm-avf2&gt; *)   ?c2[rdfs:subClassOf-&gt;?c1] :- And(     ?p1[rdfs:subPropertyOf-&gt;?p2] )</pre>
<p>(?a rdf:type owl:AllDifferent)                  (?a owl:members ?l)</p>	<pre>for(?x in ?l) {   for(?y in ?l, != x) {     {\$       Forall (         rif:error() :- And (           ?x[owl:sameAs-&gt;?y] ) )       \$}     }   }</pre>
<p>(?a rdf:type owl:AllDifferent)                  (?a owl:distinctMembers ?l)</p>	<pre>for(?x in ?l) {   for(?y in ?l, != ?x) {     {\$       Forall (         rif:error() :- And (           ?x[owl:sameAs-&gt;?y] ) )       \$}     }   }</pre>
<p>(?l rdf:type                  owl:AllDisjointProperties)</p>	<pre>for(?x in ?l) {   for(?y in ?l, != ?x) {     {\$       Forall ?p1 ?p2 (         rif:error() :- And (           ?x[?p1-&gt;?y ?p2-&gt;?y] ) )       \$}     }   }</pre>
<p>(?l rdf:type                  owl:AllDisjointClasses)</p>	<pre>for(?x in ?l) {   for(?y in ?l, != ?x) {     {\$       Forall ?c1 ?c2 (         rif:error() :- And (           ?x[rdf:type-&gt;?c1 rdf:type-&gt;?c2] ) )       \$}     }   }</pre>

	<pre> } } </pre>
(?p owl:propertyChainAxiom ?pc)	<pre> {\$   Forall ?u0 ?u\$length(?pc)\$ (     ?start[?p-&gt;?last] :- And (   \$)   for(?next in ?sc) {     {\$       ?u\$i\$[?next-&gt;?u\$i+1\$]     \$}   }   {\$ ) ) \$} </pre>
(?c owl:hasKey ?u)	<pre> {\$   Forall ?x ?y (     ?x[owl:sameAs-&gt;?y] :- And (       ?x[rdf:type-&gt;?c] ?y[rdf:type-&gt;?c]     \$)   for(?key in ?u) {     {\$       ?x[?key-&gt;?v] ?y[?key-&gt;?v]     \$}   }   {\$ ) ) \$} </pre>
(?c owl:intersectionOf ?l)	<pre> {\$   Forall ?y (     ?y[rdf:type-&gt;?c] :- And (   \$)   for(?ty in ?l) {     {\$       ?y[rdf:type-&gt;?ty]     \$}   }   {\$ ) ) \$} </pre>
(?c owl:unionOf ?l)	<pre> for(?ci in ?l) {   {\$     Forall ?y (       ?y[rdf:type-&gt;?c] :- And ( </pre>



	<pre>                 ?y[ rdf:type-&gt;?ci ] )             \$}         }     </pre>
(?c owl:oneOf ?l)	<pre>         for(?yi in ?l) {             {\$                 ?yi[ rdf:type-&gt;?c ]             \$}         }     </pre>
(?c owl:intersectionOf ?l)	<pre>         for(?ci in ?l) {             {\$                 Forall ?y (                     ?y[ rdf:type-&gt;?ci ] :- And (                         ?y[ rdf:type-&gt;?c ] )                 )                  Forall( ?c[ rdfs:subClassOf-&gt;?ci ] )             \$}         }     </pre>
(?c owl:unionOf ?l)	<pre>         for(?ci in ?l) {             {\$                 ?ci[ rdfs:subClassOf-&gt;?c ]             \$}         }     </pre>
(?x owl:sourceIndividual ?i1) (?x owl:assertionProperty ?p) (?x owl:targetIndividual ?i2)	<pre>         (* &lt;#prp-npa1&gt; *)         rif:error() :- And(             ?i1[?p-&gt;?i2] )     </pre>
(?x owl:sourceIndividual ?i1) (?x owl:assertionProperty ?p) (?x owl:targetValue ?i2)	<pre>         (* &lt;#prp-npa2&gt; *)         rif:error() :- And(             ?i1[?p-&gt;?i2] )     </pre>
(?c1 owl:disjointWith ?c2)	<pre>         (* &lt;#cax-dw&gt; *)         Forall ?x (             rif:error() :- And(     </pre>

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

## 8 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#sa
          <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>
    <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="&rif;iri">http://www.w3.org/2002/07/owl#sa
        <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="&rif;iri">http://www.w3.org/2002/07/owl#sa
              <Var>o</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

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

```

```

        </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
                  <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>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

    <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>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl
                  <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
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        <Frame>
          <object>
            <Var>x</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl
            <Var>y</Var>
          </slot>
        </Frame>
      </formula>
    </And>
  </if>
  <then>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif#error
      </op>
    </Atom>
  </then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#lab
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#com
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>

```

```

        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#see
    </object>
    <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
    </slot>
    </Frame>
</sentence>
<sentence>
    <Frame>
        <object>
            <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#isD
        </object>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
        </slot>
        </Frame>
    </sentence>
<sentence>
    <Frame>
        <object>
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#deprecated
        </object>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
        </slot>
        </Frame>
    </sentence>
<sentence>
    <Frame>
        <object>
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#priorVersi
        </object>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
        </slot>
        </Frame>
    </sentence>
<sentence>
    <Frame>
        <object>
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#backwardCo
        </object>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-

```

```

        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
    </slot>
</Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#incompatib
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Annotation
    </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/rd
                  <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>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf
      <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="&rif;iri">http://www.w3.org/2000/01/rdf
        <Var>c</Var>
      </slot>
    </Frame>
  </formula>
</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="&rif;iri">http://www.w3.org/1999/02/22-rdf
      <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>
  </Forall>
</sentence>

```

```

<Implies>
  <if>
    <And>
      <formula>
        <Frame>
          <object>
            <Var>p</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22
            <Const type="&rif;iri">http://www.w3.org/2002/07/owl
          </slot>
        </Frame>
      </formula>
      <formula>
        <Frame>
          <object>
            <Var>x</Var>
          </object>
          <slot ordered="yes">
            <Var>p</Var>
            <Var>y1</Var>
          </slot>
        </Frame>
      </formula>
      <formula>
        <Frame>
          <object>
            <Var>x</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#sa
        <Var>y2</Var>
      </slot>
    </Frame>
  </then>

```



```

        </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
                  <Const type="&rif;iri">http://www.w3.org/2002/07/ow
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x1</Var>
                </object>
                <slot ordered="yes">
                  <Var>p</Var>
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
  <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="&rif;iri">http://www.w3.org/2002/07/owl#sa
      <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="&rif;iri">http://www.w3.org/1999/02/22
                  <Const type="&rif;iri">http://www.w3.org/2002/07/owl
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

    <formula>
      <Frame>
        <object>
          <Var>x</Var>
        </object>
        <slot ordered="yes">
          <Var>p</Var>
          <Var>x</Var>
        </slot>
      </Frame>
    </formula>
  </And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </op>
  </Atom>
</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="&rif;iri">http://www.w3.org/1999/02/22
                  <Const type="&rif;iri">http://www.w3.org/2002/07/ow
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        </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="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</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="&rif;iri">http://www.w3.org/2007/rif#error
    </op>
  </Atom>
</then>
</Implies>
</formula>
</Forall>
</sentence>
<sentence>
  <Forall>
    <declare>
      <Var>p</Var>
    </declare>

```

```

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

```

```

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

```

```

        <Atom>
          <op>
            <Const type="&rif;iri">http://www.w3.org/2007/rif#error
          </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/ow
                  <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>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </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
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
          </if>
        </And>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        <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>y</Var>
        </object>
        <slot ordered="yes">
          <Var>p1</Var>
          <Var>x</Var>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Thing</Con
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Class</Con
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Nothing</C
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Class</Con
    </slot>

```

```

</Frame>
</sentence>
<sentence>
  <forall>
    <declare>
      <Var>x</Var>
    </declare>
    <formula>
      <Implies>
        <if>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#No
            </slot>
          </Frame>
        </if>
        <then>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/rif#error
            </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>
  </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
              <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
              <Var>p</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>u</Var>
            </object>
            <slot ordered="yes">
              <Var>p</Var>
              <Var>v</Var>
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>v</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22
              <Var>y</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
  </Implies>
</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
        <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>
    <formula>
      <Implies>
        <if>
          <And>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/ow
                  <Const type="&rif;iri">http://www.w3.org/2002/07/ow
                </slot>
              </Frame>
            </formula>
          <formula>
            <Frame>

```

```

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

```

```

        <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
      <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

```

```

        <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#
      <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#
      <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
              <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
              <Var>p</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
  </Implies>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Var>p</Var>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</if>

```

```

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

```

```

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

```

```

</Frame>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <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/ow
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
    </slot>
  </Frame>
</formula>
<formula>
  <Frame>
    <object>
      <Var>u</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <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#sa
      <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/rd

```

```

        <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-
    <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
      <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
        <Var>c2</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <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
      <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
              <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
              <Var>c2</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
          <Var>c1</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
              <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
              <Var>c1</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
    <then>
      <Atom>

```

```

        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif#error
        </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="&rif;iri">http://www.w3.org/1999/02/22-rdf
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Cl
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>c</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
              <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="&rif;iri">http://www.w3.org/1999/02/22-rdf
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Cl
      </slot>
    </Frame>
  </if>
  <then>
    <Frame>
      <object>
        <Var>c</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#eq
        <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="&rif;iri">http://www.w3.org/1999/02/22-rdf
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Cl
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>

```

```

        <Var>c</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Th
      </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="&rif;iri">http://www.w3.org/1999/02/22-rdf
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Cl
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#No
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
              <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-sc
                <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-sc
                <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-sc
            <Var>c3</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>

```

```

    </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#eq
              <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-sc
              <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#eq
        <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-sc
        <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-sc
                  <Var>c2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        </Frame>
    </formula>
</formula>
<formula>
    <Frame>
        <object>
            <Var>c2</Var>
        </object>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/2000/01/rdf
            <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#eq
            <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>
                        <slot ordered="yes">
                            <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf
                            <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Ob
                        </slot>
                    </Frame>
                </if>
            </Implies>
        </formula>
    </Forall>
</sentence>

```

```

        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
              <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
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Ob
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#eq
              <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
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Da
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-s
              <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
              <Const type="&rif;iri">http://www.w3.org/2002/07/owl#Da
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-s
              <Var>p</Var>
            </slot>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

        </slot>
      </Frame>
    </if>
    <then>
      <Frame>
        <object>
          <Var>p</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2002/07/owl#eq
          <Var>p</Var>
        </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="&rif;iri">http://www.w3.org/2000/01/rdf
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p2</Var>

```

```

        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-
          <Var>p3</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-sc
      <Var>p3</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="&rif;iri">http://www.w3.org/2002/07/owl#eq
              <Var>p2</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>

```

```

        <Var>p1</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
        <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="&rif;iri">http://www.w3.org/2002/07/owl#eq
              <Var>p2</Var>
            </slot>
          </Frame>
        </if>
        <then>
          <Frame>
            <object>
              <Var>p2</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
              <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="&rif;iri">http://www.w3.org/2000/01/rdf-syntax-ns#</Const>
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p2</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-syntax-ns#</Const>
                  <Var>p1</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
        <then>
          <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>
        </then>
      </Implies>
    </formula>
  </forall>

```

```

    </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-syntax-ns#</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-syntax-ns#</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-sc
        <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-sc
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>p1</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rf;iri">http://www.w3.org/2000/01/rdf-sc
                  <Var>p2</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        <then>
          <Frame>
            <object>
              <Var>p1</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
              <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="&rif;iri">http://www.w3.org/2000/01/rdf-sc
                  <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-sc

```

```

        <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-sc
      <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/rd
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```



```

        <Frame>
          <object>
            <Var>p1</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
            <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-sc
      <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/ow
      <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/ow
      <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/ow
      <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/ow
      <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-sc
          <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-sc
      <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/ow
        <Var>y1</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>y2</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd

```

```

        <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-sc
      <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/ow
        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p1</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p2</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>p1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      <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-sc
        <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/ow
                <Var>y1</Var>
              </slot>

```

```

    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>y2</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      <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-so
              <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/ow
                  <Var>y</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    <Frame>
      <object>
        <Var>c1</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2002/07/ow
        <Var>p1</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>c2</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/ow
      <Var>p2</Var>
    </slot>
  </Frame>
</formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>p1</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      <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-sc
        <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="&rif;iri">http://www.w3.org/2002/07/ow
                  <Var>i1</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/2002/07/ow

```

```

        <Var>p</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
      <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="&rif;iri">http://www.w3.org/2007/rif#error
    </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="&rif;iri">http://www.w3.org/2002/07/owl#</Const>
                <Var>i</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#</Const>
                <Var>p</Var>
              </slot>
            </Frame>
          </formula>
        </And>
      </if>
    </Implies>
  </formula>
  <formula>
    <Frame>
      <object>
        <Var>x</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl#</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="&rif;iri">http://www.w3.org/2007/rif#error
    </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="&rif;iri">http://www.w3.org/2002/07/ow
                  <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
          <Var>c1</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
      <Var>c2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </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="&rif;iri">http://www.w3.org/2002/07/ow
          <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
      <Var>c1</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
      <Var>c2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </op>
  </Atom>
</then>
</Implies>
</formula>
</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="&rif;">
  <payload>
    <Group>
      <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/20
                              </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
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf
    </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/20
        </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
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                </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
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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/20
                    </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="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
  <External>
    <content>

```

```

        <Atom>
          <op>
            <Const type="&rif;iri">http://www.w3.org/20
          </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                  </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
            <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
          </slot>

```

```

        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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/20
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </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
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      </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/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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>
</sentence>
</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/20
              </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
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      </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/20
                    </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>

```

```

        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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/20
                  </op>
                </atom>
              </content>
            </External>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
</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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                </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
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>

```

```

        </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/20
                    </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="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      </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/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                </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
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">

```

```

        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                  </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
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      </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/20
            </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
            </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                    </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
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                    </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                  </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
            <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
          </slot>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22
            <Const type="&rif;iri">http://www.w3.org/2000/01/rd
          </slot>
        </Frame>
      </then>

```



```

        </Implies>
      </formula>
    </forall>
  </sentence>
</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/20
                    </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
        <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/2000/01/rd
      </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/20
    </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
            <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/1999/02/22
            <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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/20
      </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/20
                </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
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22

```

```

        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf
      </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/20
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </And>
      </if>
    <then>
      <Frame>
        <object>
          <Var>lt</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
        </slot>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22
          <Const type="&rif;iri">http://www.w3.org/2000/01/rd
        </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>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        <External>
          <content>
            <Atom>
              <op>
                <Const type="&rif;iri">http://www.w3.org/20
              </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
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSc
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Const type="&rif;iri">http://www.w3.org/2000/01/rd
    </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/1999/0
        <Const type="&rif;iri">http://www.w3.org/1999/0
      </slot>
    </Frame>
  </formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/20
      </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#e
    </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/1999/0
          <Const type="&rif;iri">http://www.w3.org/1999/0
        </slot>
      </Frame>
    </formula>
  </formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
        <Const type="&rif;iri">http://www.w3.org/2001/X
      </slot>
    </Frame>
  </formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/20
      </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#e
    </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/1999/0
          <Const type="&rif;iri">http://www.w3.org/2001/X
        </slot>
      </Frame>
    </formula>
  </formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
          <Const type="&rif;iri">http://www.w3.org/2001/X
        </slot>
      </Frame>
    </formula>
  </formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
        <Const type="&rif;iri">http://www.w3.org/2001/X
    </slot>
  </Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0

```

```

        <Const type="&rif;iri">http://www.w3.org/2001/X
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X

```

```

        </slot>
      </Frame>
    </formula>
  </formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>

```



```

    </Frame>
  </formula>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

    <formula>
      <External>
        <content>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/20
            </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```

```

        <External>
        <content>
        <Atom>
        <op>
        <Const type="&rif;iri">http://www.w3.org/20
        </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#e
    </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/1999/0
    <Const type="&rif;iri">http://www.w3.org/2001/X
    </slot>
    </Frame>
    </formula>
    </formula>
    </forall>
    <External>

```

```

        <content>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/20
            </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>
</content>

```

```

    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/20
      </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>
</content>
<Atom>

```

```

        <op>
        <Const type="&rif;iri">http://www.w3.org/20
    </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#e
    </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/1999/0
                                    <Const type="&rif;iri">http://www.w3.org/2001/X
                                </slot>
                            </Frame>
                        </formula>
                    </And>
                </if>
            </Implies>
        </formula>
    </forall>
    </sentence>
    </External>
    </content>
    <Atom>
        <op>

```

```

        <Const type="&rif;iri">http://www.w3.org/20
    </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#e
        </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/1999/0
                                    <Const type="&rif;iri">http://www.w3.org/2001/X
                                </slot>
                            </Frame>
                        </formula>
                    </And>
                </if>
            </Implies>
        </formula>
    </forall>
</sentence>
    <Atom>
        <op>
            <Const type="&rif;iri">http://www.w3.org/20

```



```

        </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </op>

```



```

        <Var>lt</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </op>
                </Atom>
                <args ordered="yes">
                  <Var>lt</Var>

```



```

        </Atom>
      </content>
    </External>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        </External>
      </formula>
    </And>
  </if>
  <then>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif#e
      </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </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#e
      </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/20
                  </op>
                  <args ordered="yes">
                    <Var>lt</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </if>
      </Implies>
    </formula>
  </Forall>
</sentence>

```





```

    </if>
    <then>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif#e
        </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/20
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </forall>
  </sentence>
</document>

```

```

    <then>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif#e
        </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/20
                    </op>
                    <args ordered="yes">
                      <Var>lt</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </forall>
  </sentence>
</text>

```

```

        <Atom>
          <op>
            <Const type="&rif;iri">http://www.w3.org/2007/rif#e
          </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/20
                    </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#e
        </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/20
                    </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#e
    </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/1999/0
                  <Const type="&rif;iri">http://www.w3.org/2001/X
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/20
                    </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#e

```

```

        </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</Const>
        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<External>
  <content>
    <Atom>
      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/rif#error</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>
      </And>
    </if>
  </Implies>
  <Equal>
    <left>
      <Var>x</Var>
    </left>
  </Equal>
</formula>

```

```

        </left>
        <right>
          <Var>y</Var>
        </right>
      </Equal>
    </formula>
  <formula>
    <Frame>
      <object>
        <Var>x</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/2002/07/owl
        <Var>y</Var>
      </slot>
    </Frame>
  </formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </op>
  </Atom>
</then>
</Implies>
</formula>
</forall>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#decimal
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-

```

```

        <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#integer
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#double<
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#string<
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTim
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>

```

```

<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Lit
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonNegat
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#nonPosit
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#positiv

```

```

    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#negativ
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#long</C
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#int</Co
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#short</
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat

```



```

<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#unsigned
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
  </slot>
</Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#normali
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#token</
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#languag
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
      <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
    </slot>
  </Frame>
</sentence>
<sentence>
  <Frame>
    <object>
      <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#Name</C
    </object>

```

```

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

```



```

    </Frame>
  </sentence>
</sentence>
<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#hexBina
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
  </slot>
</Frame>
</sentence>
</sentence>
<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#base64B
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
  </slot>
</Frame>
</sentence>
</sentence>
<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#anyURI<
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
  </slot>
</Frame>
</sentence>
</sentence>
<Frame>
  <object>
    <Const type="&rif;iri">http://www.w3.org/2001/XMLSchema#dateTim
  </object>
  <slot ordered="yes">
    <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf-syntax-
    <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-schema#Dat
  </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="&rif;">
  <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>
                    </Frame>
                  </formula>
                  <slot ordered="yes">
                    <Const type="&rif;iri">http://www.w3.org/1999/02/22
                    <Const type="&rif;iri">http://www.w3.org/2002/07/ow
```

```

    </slot>
  </Frame>
</formula>
<formula>
  <Frame>
    <object>
      <Var>x</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/2002/07/owl#</Const>
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif#</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#</Const>
            </op>
            <args ordered="yes">
              <Var>y</Var>
              <Var>z1</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>

```

```

    </right>
  </Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/r
        </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/20
            </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/r
        </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
      <Var>z2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </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
              <Const type="&rif;iri">http://www.w3.org/2002/07/ow
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>x</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/ow
              <Var>y</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
  </Implies>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/r
        </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/20
        </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/r
        </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/20
            </op>
            <args ordered="yes">
              <Var>y</Var>
              <Var>z2</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>
  </Equal>
</formula>

```

```

        </args>
      </Expr>
    </content>
  </External>
</right>
</Equal>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/r
        </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/ow
      <Var>z2</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </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
              <Const type="&rif;iri">http://www.w3.org/2002/07/ow
            </slot>
          </Frame>
        </formula>
        <formula>
          <Frame>
            <object>
              <Var>r</Var>
            </object>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/2002/07/ow
              <Var>l</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
  </Implies>
</formula>

```

```

</Frame>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/r
        </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/20
            </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/r
        </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/20
            </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/r
        </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
    </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>
  <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
          <Const type="&rif;iri">http://www.w3.org/2002/07/ow
        </slot>
      </Frame>
    </formula>
    <formula>
      <Frame>
        <object>
          <Var>r</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2002/07/ow
          <Var>l</Var>
        </slot>
      </Frame>
    </formula>
    <formula>
      <External>
        <content>
          <Atom>
            <op>
              <Const type="&rif;iri">http://www.w3.org/2007/r
            </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/20
        </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/r
        </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/20
            </op>
            <args ordered="yes">
              <Var>l</Var>
              <Var>y</Var>
            </args>
          </Expr>
        </content>
      </External>
    </right>

```

```

    </Equal>
  </formula>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/rif#error
        </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/rif#
      <Var>x</Var>
    </slot>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22/rif#
      <Var>y</Var>
    </slot>
  </Frame>
</formula>
</And>
</if>
<then>
  <Atom>
    <op>
      <Const type="&rif;iri">http://www.w3.org/2007/rif#error
    </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</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>
      <then>
        <Frame>
          <object>
            <Var>start</Var>
          </object>
          <slot ordered="yes">
            <Var>p</Var>
            <Var>last</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>

```



```

        </slot>
      </Frame>
    </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>
    <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
                  <Var>p</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22
                  <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
                <Var>p</Var>
              </slot>
              <slot ordered="yes">
                <Const type="&rif;iri">http://www.w3.org/1999/02/22
                <Const type="&rif;iri">http://www.w3.org/1999/02/22
              </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>
    <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
                  <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
                  <Var>c</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>

```

```

        <Var>y</Var>
      </object>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <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#sa
      <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
              <Var>key</Var>
            </slot>
            <slot ordered="yes">
              <Const type="&rif;iri">http://www.w3.org/1999/02/22
              <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>
      </And>
    </if>
  </Implies>
</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>
    </args>
  </Atom>
</formula>

```

```

        <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/
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">

```

```

        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
    </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>
    <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
                <Var>l</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <Atom>
              <op>
                <Const type="&rif;local">allTypes</Const>
              </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
            <Var>c</Var>
          </slot>
        </Frame>
      </then>
    </Implies>
  </formula>

```

```

    </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
                  <Var>ty</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22
                  <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
                  <Var>ty</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

        <Atom>
          <op>
            <Const type="&rif;local">allTypes</Const>
          </op>
          <args ordered="yes">
            <Var>t1</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
        <Var>ty</Var>
      </slot>
      <slot ordered="yes">
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
        <Const type="&rif;iri">http://www.w3.org/1999/02/22
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Var>ty</Var>
    </slot>
  </Frame>
</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>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</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</Const>
              <Var>c</Var>
            </slot>
          </Frame>
        </formula>
      </And>
    </if>
  </Implies>
</formula>
<formula>
  <Atom>
    <op>
      <Const type="&rif;local">sameKey</Const>
    </op>
    <args ordered="yes">
      <Var>u</Var>
      <Var>x</Var>
    </args>
  </Atom>
</formula>

```

```

        <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#sa
      <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
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22
                  <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>
<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>

```

```

        </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
                  <Var>key</Var>
                </slot>
                <slot ordered="yes">
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22
                  <Const type="&rif;iri">http://www.w3.org/1999/02/22
                </slot>
              </Frame>
            </formula>
            <formula>
              <Frame>
                <object>
                  <Var>x</Var>
                </object>
                <slot ordered="yes">
                  <Var>key</Var>
                  <Var>v</Var>
                </slot>
              </Frame>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```



```

        <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>
      <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
        <Var>l</Var>
      </slot>
    </Frame>
  </formula>
</formula>
<formula>
  <External>
    <content>
      <Atom>
        <op>
          <Const type="&rif;iri">http://www.w3.org/2007/r
        </op>
        <args ordered="yes">
          <Var>l</Var>
          <Var>ci</Var>
        </args>
      </Atom>
    </content>
  </External>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Var>ci</Var>
    </slot>
  </Frame>
</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
      <Var>c</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
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/r
                    </op>
                    <args ordered="yes">
                      <Var>l</Var>
                      <Var>yi</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```

    <then>
      <Frame>
        <object>
          <Var>yi</Var>
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/1999/02/22-rdf
          <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="&rif;iri">http://www.w3.org/2002/07/ow
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>

```

```

      <op>
        <Const type="&rif;iri">http://www.w3.org/2007/r
      </op>
      <args ordered="yes">
        <Var>l</Var>
        <Var>ci</Var>
      </args>
    </Atom>
  </content>
</External>
</formula>
<formula>
  <Frame>
    <object>
      <Var>y</Var>
    </object>
    <slot ordered="yes">
      <Const type="&rif;iri">http://www.w3.org/1999/02/22
      <Var>c</Var>
    </slot>
  </Frame>
</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
      <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
                <Var>l</Var>
              </slot>
            </Frame>
          </formula>
          <formula>
            <External>
              <content>
                <Atom>
                  <op>
                    <Const type="&rif;iri">http://www.w3.org/2007/r
                  </op>
                  <args ordered="yes">
                    <Var>l</Var>
                    <Var>ci</Var>
                  </args>
                </Atom>
              </content>
            </External>
          </formula>
        </And>
      </if>
      <then>
        <Frame>
          <object>
            <Var>c</Var>
          </object>
          <slot ordered="yes">
            <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
            <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#
                  <Var>l</Var>
                </slot>
              </Frame>
            </formula>
            <formula>
              <External>
                <content>
                  <Atom>
                    <op>
                      <Const type="&rif;iri">http://www.w3.org/2007/rif#
                    </op>
                    <args ordered="yes">
                      <Var>l</Var>
                      <Var>ci</Var>
                    </args>
                  </Atom>
                </content>
              </External>
            </formula>
          </And>
        </if>
        <then>
          <Frame>
            <object>
              <Var>ci</Var>
            </object>
          </Frame>
        </then>
      </Implies>
    </formula>
  </forall>
</sentence>

```

```
        </object>
        <slot ordered="yes">
          <Const type="&rif;iri">http://www.w3.org/2000/01/rdf-sc
          <Var>c</Var>
        </slot>
      </Frame>
    </then>
  </Implies>
</formula>
</forall>
</sentence>
</Group>
</payload>
</Document>
```

## 9 Appendix: Change Log (Informative)

This section summarizes changes with 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