Content Models for RuleML

David Hirtle

2005-06-08, version 0.89

Introduction

This document is a collection of content models for all RuleML tags as of version 0.89, organized alphabetically by module name. Each module is a grouping of related elements and/or attributes (prefixed with "@"). The content models, i.e. the content permitted within a given element, are given in BNF-like DTD syntax. See http://www.ruleml.org/0.89/xsd/modules/ for the actual XML schemas of the modules.

Note that the content model of a given element often varies depending on the sublanguage. In such cases, all variations of the content model are provided along with the corresponding sublanguage(s). The official model for the modularization of RuleML, including all sublanguages, is at http://www.ruleml.org/modularization.

Content models may also vary depending on context, e.g. surrounding elements, especially parent elements. In these cases, the content models are listed under a heading such as "within x" where x indicates the context.

For clarification on any RuleML-related topic, including this document, the RuleML-all mailing list may be quite helpful. Another available resource is the RuleML tutorial.

Index of Modules

Atom - atom_module.xsd

```
*** Atom ***
        attributes: @closure

        in datalog, nafdatalog, nafnegdatalog, negdatalog:
        (  oid?,
             ( (opr | Rel), (slot)*, (arg | Ind | Data | Skolem | Var | Reify)*, (slot)* ) |
             ( (slot)*, (arg | Ind | Data | Skolem | Var | Reify)+, (slot)*, opr )
        )

        in bindatalog:
        (
           oid?,
            ((opr|Rel),(slot)*,(arg|Ind|Data|Skolem|Var|Reify),(arg|Ind|Data|Skolem|Var|Reify),(slot)*)|
            ( (slot)*, (arg|Ind|Data|Skolem|Var|Reify), (arg|Ind|Data|Skolem|Var|Reify), (slot)*, opr )
        )

        in bindatagroundlog and bindatagroundfact:
        (
           oid?,
            ((opr|Rel),(slot)*,(arg|Ind|Data|Skolem|Reify),(arg|Ind|Data|Skolem|Reify),(slot)*) |
            ( (slot)*, (arg|Ind|Data|Skolem|Reify), (arg|Ind|Data|Skolem|Reify), (slot)*, opr )
        )

        in hornlog & up (except framehohornlogeq):
        (
           oid?,
             ((opr|Rel),(slot)*,(arg|Ind|Data|Skolem|Var|Reify|Cterm|Plex)*,(repo)?,(slot)*,(resl)?) |
             ((slot)*, (arg|Ind|Data|Skolem|Var|Reify|Cterm|Plex)+, (repo)?, (slot)*, (resl)?, opr )
        )

        in framehohornlogeq & up: ( oid, (op | Con | Skolem | Var | Reify | Hterm)?, slot* )


*** opr ***

        in all sublanguages: (Rel)


*** Rel ***
        attributes: @wref

        in all sublanguages: (#PCDATA)
```

Connective - connective_module.xsd

```
*** Implies ***
        attributes: @closure, @direction ( + @mapDirection and @mapClosure in folog & up)

        in datalog & down and hornlog:
        ( oid?, ( head, body) | ( body, head) | ( (Atom | And | Or), Atom ) )

        in negdatalog:
        ( oid?, ( head, body) | ( body, head) | ( (Atom | And | Or | Neg), (Atom | Neg) ))

        in nafdatalog:
        ( oid?, ( head, body) | ( body, head) | ( (Atom | And | Or | Naf), Atom ) )

        in nafnegdatalog:
        (oid?, ( head, body) | ( body, head) | ( (Atom | And | Or | Neg | Naf), (Atom | Neg) ))

        in hornlogeq:
        ( oid?, ( head, body) | ( body, head) | ( (Atom | And | Or | Equal), (Atom | Equal) ))

        in hohornlog & up: ( oid?, ( head, body) | ( body, head) | ( (And | Or | Hterm), Hterm) )

        in framehohornlogeq:
        (
           oid?, ( head, body ) | ( body, head ) |
                          (
                           (Atom | InstanceOf | SubclassOf | And | Or | Hterm),
                           (Atom | InstanceOf | SubclassOf | Hterm)
                          )
        )

        in dishornlog: ( oid?, ( head, body) | ( body, head) | ( (Atom | And | Or), (Atom | Or)))

        in folog:
        (
           oid?, (head, body) | (body, head) |
                         (
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists ),
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )
                         )
        )

        in naffolog:
        (
           oid?, (head, body) | (body, head) |
                         (
                           (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists ),
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )
                         )
        )

        in fologeq:
        (
           oid?, (head, body) | (body, head) |
                         (
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal ),
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )
                         )
        )

        in naffologeq:
        (
           oid?, (head, body) | (body, head) |
                         (
                           (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists | Equal ),
                           (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )
                         )
        )
```

```
*** body ***

        in datalog & down and hornlog and dishornlog: (Atom | And | Or)

        in negdatalog: (Atom | And | Or | Neg)

        in nafdatalog: (Atom | And | Or | Naf)

        in nafnegdatalog: (Atom | And | Or | Neg | Naf)

        in hornlogeq: (Atom | And | Or | Equal)

        in hohornlog & up: (Hterm | And | Or)

        in framehohornlogeq: (Atom | InstanceOf | SubclassOf | And | Or | Hterm)

        in folog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

        in naffolog: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists )

        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )

        in naffologeq: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists | Equal )


*** head ***

        in datalog & down, nafdatalog and hornlog: (Atom)

        in negdatalog: (Atom | Neg)

        in nafnegdatalog: (Atom | Neg)

        in hornlogeq: (Atom | Equal)

        in hohornlog & up: (Hterm)

        in framehohornlogeq: (Atom | InstanceOf | SubclassOf | Hterm)

        in dishornlog: (Atom | Or)

        in folog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )

        in naffologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )


*** Equivalent ***
        attributes: @closure ( + @mapDirection and @mapClosure in folog & up)

        in datalog & down and up to dishornlog: ( oid?, ( ( torso, torso) | ( Atom, Atom) ) )

        in hornlogeq: ( oid?, ( (torso, torso) | ( (Atom | Equal), (Atom | Equal) ) ) )

        in hohornlog & up: ( oid?, ( ( torso, torso) | ( Hterm, Hterm ) ) )

        in framehohornlogeq:
        (
           oid?, (
                  ( torso, torso) |
                  ((Atom | InstanceOf | SubclassOf | Hterm), (Atom | InstanceOf | SubclassOf | Hterm))
                )
        )

        in folog and naffolog:
        (
           oid?, (torso, torso) |
                      (
                         (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists ),
                         (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )
                       )
        )
        in fologeq:
        (
           oid?, (torso, torso) |
                       (
                          (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal),
                          (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)
                        )
        )

        in naffologeq:
        (
           oid?, (torso, torso) |
                       (
                          (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal),
                          (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)
                        )
        )


 *** torso ***

        in datalog & down and up to dishornlog: (Atom)

        in hornlogeq: ( Atom | Equal )

        in hohornlog & up: ( Hterm )

        in framehohornlogeq: (Atom | InstanceOf | SubclassOf | Hterm)

        in folog and naffolog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)

        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)

        in naffologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)


*** And *** (context sensitive)

within Assert...
        attributes: @mapDirection and @mapClosure

        in bindatalog, datalog up to hornlog and dishornlog:
        ( oid?, (formula | Atom | Implies | Equivalent | Forall)* )

        in bindatagroundlog: ( oid?, (formula | Atom | Implies | Equivalent)* )

        in bindatagroundfact: ( oid?, (formula | Atom)* )

        in hornlogeq: ( oid?, (formula | Atom | Implies | Equivalent | Forall | Equal)* )

        in hohornlog & up: ( oid?, (formula | Hterm | Implies | Equivalent | Forall)* )

        in framehohornlogeq:
        ( oid?, (formula | Atom | InstanceOf | SubclassOf | Hterm | Implies | Equivalent | Forall)* )

        in folog and naffolog:
        ( oid?, (formula | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)* )

        in fologeq:
        (oid?,(formula | Atom| And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)*)

        in naffologeq:
        (oid?,(formula | Atom| And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)*)

within Query, Implies, Exists and And/Or...
        attributes within Query only: @closure ( + @mapDirection and @mapClosure in folog & up)

        in datalog & down, hornlog and dishornlog: ( oid?, (formula | Atom | And | Or)* )

        in negdatalog: ( oid?, (formula | Atom | And | Or | Neg)* )

        in nafdatalog: ( oid?, (formula | Atom | And | Or | Naf)* )

        in nafnegdatalog: ( oid?, (formula | Atom | And | Or | Naf | Neg)* )
        in hornlogeq: ( oid?, (formula | Atom | And | Or | Equal)* )

        in hohornlog & up: ( oid?, (formula | Hterm | And | Or)* )

        in framehohornlogeq: ( oid?, (formula | Atom | InstanceOf | SubclassOf | Hterm | And | Or)* )

        in folog:
        ( oid?, (formula | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)* )

        in naffolog:
        (oid?, (formula | Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists)*)

        in fologeq:
        (oid?,(formula | Atom | And|Or | Neg | Implies|Equivalent | Forall|Exists | Equal)*)

        in naffologeq:
        (oid?,(formula | Atom | And|Or | Neg|Naf | Implies|Equivalent | Forall|Exists | Equal)*)

within oppo...

        in all sublanguages (except hohornlog, etc): ( oid?, (formula | Atom), (formula | Atom) )

        hohornlog & up: ( oid?, (formula | Hterm | Neg), (formula | Hterm | Neg) )


*** Or *** (context sensitive)
       attributes within Query only: @closure ( + @mapDirection and @mapClosure in folog & up)

        in datalog & down, hornlog and dishornlog: ( oid?, (formula | Atom | And | Or)* )

        in negdatalog: ( oid?, (formula | Atom | And | Or | Neg)* )

        in nafdatalog: ( oid?, (formula | Atom | And | Or | Naf)* )

        in nafnegdatalog: ( oid?,  (formula | Atom | And | Or | Naf | Neg)* )

        in hornlogeq: ( oid?, (formula | Atom | And | Or | Equal)* )

        in hohornlog & up: ( oid?, (formula | Hterm | And | Or)* )

        in framehohornlogeq: ( oid?, (formula | Atom | InstanceOf | SubclassOf | Hterm | And | Or)* )

        in folog:
        ( oid?, (formula | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)* )

        in naffolog:
        (oid?, (formula | Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists)*)

        in fologeq:
        (oid?,(formula | Atom| And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)*)

        in naffologeq:
        (oid?,(formula | Atom | And|Or | Neg|Naf | Implies|Equivalent | Forall|Exists | Equal)*)


*** formula *** (context sensitive; see also the quantifier module)

within top level And...

        in bindatalog, datalog up to hornlog and dishornlog: ( Atom | Implies | Equivalent | Forall )

        in bindatagroundlog: ( Atom | Implies | Equivalent )

        in bindatagroundfact: ( Atom )

        in hornlogeq: ( Atom | Implies | Equivalent | Forall | Equal )

        in hohornlog & up: (Hterm | Implies | Equivalent | Forall)

        in framehohornlogeq: (Atom | InstanceOf | SubclassOf | Hterm | Implies | Equivalent | Forall)

        in folog and naffolog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)

        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)

        in naffologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)
```

```
within inner And/Or...
        in datalog & down, hornlog and dishornlog: (Atom | And | Or)
        in negdatalog: (Atom | And | Or | Neg)
        in nafdatalog: (Atom | And | Or | Naf)
        in nafnegdatalog: (Atom | And | Or | Naf | Neg)
        in hornlogeq: (Atom | And | Or | Equal)
        in hohornlog & up: (Hterm | And | Or)
        in framehohornlogeq:  (Atom | InstanceOf | SubclassOf | Hterm | And | Or)
        in folog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists)
        in naffolog: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists)
        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal)
        in naffologeq: (Atom | And|Or | Neg|Naf | Implies|Equivalent | Forall|Exists | Equal)
within oppo...
        in all sublanguages (except hohornlog, etc): ( Atom )
        in hohornlog & up: ( Hterm | Neg )

*** @mapDirection *** [optional] (forward | backward | default:bidirectional)

*** @direction *** [optional] (forward | backward | default:bidirectional)

*** @mapClosure *** [optional] (universal | existential)

*** @closure *** [optional] (universal | existential)
```

Cterm - cterm_module.xsd

```
*** Cterm ***
        attributes: @type

        in hornlog & up (except hohornlog, etc):
        (
          ( (opc | Ctor),(slot)*, (arg|Ind|Data|Skolem|Var|Reify|Cterm|Plex)*,(repo)?,(slot)*,(resl)? ) |
          ( (slot)*, (arg|Ind|Data|Skolem|Var|Reify|Cterm|Plex)*, (repo)?, (slot)*, (resl)?, opc )
        )


*** opc ***

        in hornlog & up (except hohornlog, etc): (Ctor)


*** Ctor ***
        attributes: @wref

        in hornlog & up (except hohornlog, etc): (#PCDATA)


*** Plex *** (context sensitive)

within Atom, Plex, slot...

        in hornlog & up (except hohornlog, etc):
        ( (slot)*, (arg | Ind | Data | Skolem | Var | Reify | Cterm | Plex)*, (repo)?, (slot)*, (resl)? )

        in hohornlog & up:
        ( (slot)*, (arg | Con | Skolem | Var | Reify | Hterm)*, (repo)?, (slot)*, (resl)? )

within repo...

        in hornlog & up (except hohornlog, etc):
        ( (arg | Ind | Data | Skolem | Var | Reify | Cterm | Plex | repo)* )

        in hohornlog & up: ( (arg | Con | Skolem | Var | Reify | Hterm | repo)* )

within resl...

        in hornlog & up: ( (slot | resl)* )
```

Desc - desc_module.xsd

```
*** oid ***

        in datalog & down, negdatalog, nafdatalog and nafnegdatalog: (Ind | Data | Var | Skolem | Reify)

        in hornlog & up (except hohornlog, etc): (Ind | Data | Var | Skolem | Reify | Cterm | Plex)

        in hohornlog & up: (Con | Data | Skolem | Var | Reify | Hterm)
```

Equality - equality_module.xsd

```
*** Equal ***

       in hornlogeq, fologeq and naffologeq:
       (
         (side | Ind | Data | Skolem | Var | Reify | Cterm | Plex | Nano),
         (side | Ind | Data | Skolem | Var | Reify | Cterm | Plex | Nano)
       )

       in hohornlogeq:
       (
         (side | Con | Skolem | Var | Reify | Hterm | Nano),
         (side | Con | Skolem | Var | Reify | Hterm | Nano)
       )


*** side ***

       in hornlogeq, fologeq and naffologeq: ( Ind | Data | Skolem | Var | Reify | Cterm | Plex | Nano )

       in hohornlogeq: (Con | Skolem | Var | Reify | Hterm | Nano)


*** Nano ***

       in hornlogeq, fologeq and naffologeq:
       (
         ( (opf | Fun), (arg | Ind | Data | Skolem | Var | Reify | Cterm | Plex)* ) |
         ( (arg | Ind | Data | Skolem | Var | Reify | Cterm | Plex)+, opf )
       )

       in hohornlogeq:
       (
         ( (opf | Fun), (arg | Con | Skolem | Var | Reify | Hterm)* ) |
         ( (arg | Con | Skolem | Var | Reify | Hterm)+, opf )
       )


*** opf ***

       in hornlogeq, hohornlogeq, fologeq and naffologeq: (Fun)

*** Fun ***
       attributes: @wref

       in hornlogeq, hohornlogeq, fologeq and naffologeq: (#PCDATA)
```

Frame - frame_module.xsd

```
*** Set ***

       in framehohornlogeq: ( (Con | Skolem | Var | Reify | Hterm | Get)* )


*** InstanceOf ***

       in framehohornlogeq:
       ( (Con|Skolem|Var|Reify|Hterm|Get),(Con|Skolem|Var|Reify|Hterm|Get) )


*** SubclassOf ***

       in framehohornlogeq:
       ( (Con|Skolem|Var|Reify|Hterm|Get),(Con|Skolem|Var|Reify|Hterm|Get) )


*** Signature ***

       in framehohornlogeq: ( oid, (op | Con | Skolem | Var | Reify | Hterm)?, slot* )


*** Get ***

       in framehohornlogeq: ( oid, SlotProd )


*** SlotProd ***

       in framehohornlogeq: (Con | Skolem | Var | Reify | Hterm | Get)+
```

Holog - holog_module.xsd

```
*** Hterm ***

       in hohornlog & up:
       (
          oid?, (op | Con | Skolem | Var | Reify | Hterm), (slot)*,
          (arg | Con | Skolem | Var | Reify | Hterm)*, (repo)?, (slot)*, (resl)?
       )


*** op ***

       in hohornlog & up: (Con | Skolem | Var | Reify | Hterm)


*** Con *** (context sensitive)

       in hohornlog & up: (#PCDATA)

within oid...
       attributes: @wlab, @type

not within oid...
       attributes: @wref, @type
```

Mutex - mutex_module.xsd

```
*** mutex ***

       in all sublanguages: ( (oppo, mgiv?) | (mgiv, oppo) )

*** oppo ***

       in all sublanguages: ( And )

*** mgiv ***

       in datalog & down and hornlog and dishornlog: (Atom | And | Or)

       in negdatalog: (Atom | And | Or | Neg)

       in nafdatalog: (Atom | And | Or | Naf)

       in nafnegdatalog: (Atom | And | Or | Neg | Naf)

       in hornlogeq: (Atom | And | Or | Equal)

       in hohornlog & up: (Hterm | And | Or)

       in folog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

       in naffolog: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists )

       in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )

       in naffologeq: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists | Equal )
```

Naf - naf_module.xsd

```
*** Naf ***
       attributes: none ( + @mapDirection and @mapClosure in folog & up)

       in nafdatalog: ( oid?, (weak | Atom) )

       in nafnegdatalog: ( oid?, (weak | Atom | Neg) )

       in naffolog: ( oid?, (weak | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists) )

       in naffologeq:( oid?, (weak | Atom | And|Or | Neg | Implies|Equivalent | Forall|Exists | Equal) )


*** weak ***

       in nafdatalog: ( Atom )

       in nafnegdatalog: ( Atom | Neg)

       in naffolog: ( Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

       in naffologeq:( Atom | And|Or | Neg | Implies|Equivalent | Forall|Exists | Equal )
```

Neg - neg_module.xsd

```
*** Neg ***
        attributes: none ( + @mapDirection and @mapClosure in folog & up)

       in negdatalog and nafnegdatalog: ( oid?, (strong | Atom) )

       in hohornlog & up: ( oid?, (strong | Hterm) )

       in folog and naffolog: (oid?, (strong | Atom|And|Or|Neg | Implies|Equivalent | Forall | Exists) )

       in fologeq and naffologeq:
       (oid?, (strong | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal) )


*** strong ***

       in negdatalog and nafnegdatalog: ( Atom )

       in hohornlog & up: ( Hterm )

       in folog and naffolog: ( Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

       in fologeq and naffologeq: (Atom | And|Or | Neg | Implies|Equivalent | Forall|Exists | Equal)
```

Performative - performative_module.xsd

```
*** Assert ***
        attributes: none ( + @mapDirection and @mapClosure in folog & up)

       in datalog & down and up to folog: (content | And)

       in folog and naffolog: (content | Atom | And|Or | Neg | Implies | Equivalent | Forall | Exists)

       in fologeq and naffologeq: (content | Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists|Equal)


*** Query ***
        attributes: none ( + @mapDirection and @mapClosure in folog & up)

       in datalog, bindatalog, hornlog and dishornlog: (content | Atom | And | Or | Exists)

       in bindatagroundlog and bindatagroundfact: (content | Atom | And | Or)

       in negdatalog: (content | Atom | And | Or | Exists | Neg)

       in nafdatalog: (content | Atom | And | Or | Exists | Naf)

       in nafnegdatalog: (content | Atom | And | Or | Exists | Naf | Neg)

       in hornlogeq: (content | Atom | And | Or | Exists | Equal)

       in hohornlog & up: (content | Hterm | And | Or | Exists)

       in framehohornlogeq: (content | Atom | InstanceOf | SubclassOf | Hterm | And | Or | Exists)

       in folog: (content | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

       in naffolog:
       (content | Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists )

       in fologeq:
       (content | Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )

       in naffologeq:
       (content | Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists | Equal )


*** Protect ***

       in datalog & down and up to folog: ( (warden | Mutex), (content | And) )

       in folog and naffolog:
       ( (warden | Mutex), (content | (Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists)) )

       in fologeq and naffologeq:
       ( (warden | Mutex), (content | (Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists|Equal)) )


*** content *** (context sensitive)

within Assert...

       in datalog & down and up to folog: ( And )

       in folog and naffolog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

       in fologeq and naffologeq: (Atom | And|Or | Neg | Implies|Equivalent | Forall|Exists | Equal)

within Query...

       in datalog, bindatalog, hornlog and dishornlog: (Atom | And | Or | Exists)

       in bindatagroundlog and bindatagroundfact: (Atom | And | Or)

       in negdatalog: (Atom | And | Or | Exists | Neg)

       in nafdatalog: (Atom | And | Or | Exists | Naf)

       in nafnegdatalog: (Atom | And | Or | Exists | Neg | Naf)
```

```
        in hornlogeq: (Atom | And | Or | Exists | Equal)

        in hohornlog & up: (Hterm | And | Or | Exists)

        in framehohornlogeq: (Atom | InstanceOf | SubclassOf | Hterm | And | Or | Exists)

        in folog: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

        in naffolog: (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists )

        in fologeq: (Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists | Equal )

        in naffologeq:
        (Atom | And | Or | Neg | Naf | Implies | Equivalent | Forall | Exists | Equal )


*** warden ***

        in all sublanguages: ( Mutex )
```

Quantifier - quantifier_module.xsd

```
        *** Forall ***
                attributes: none ( + @mapDirection and @mapClosure in folog & up)

                in bindatalog, datalog & up to (including) hornlog and dishornlog:
                ( oid?, (declare | Var)+, (formula | Atom | Implies | Equivalent | Forall) )

                in hornlogeq:
                ( oid?, (declare | Var)+, (formula | Atom | Implies | Equivalent | Forall | Equal) )

                in hohornlog & up: ( oid?, (declare | Var)+, (formula | Hterm | Implies | Equivalent | Forall) )

                in framehohornlogeq:
                ( oid?,(declare | Var)+,(formula | Atom|InstanceOf|SubclassOf|Hterm|Implies|Equivalent|Forall) )

                in folog and naffolog:
                ( oid?, (declare | Var)+, (formula|Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists) )

                in fologeq and naffologeq:
                ( oid?, (declare | Var)+, (formula|Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists|Equal) )


       *** Exists ***
               attributes: none ( + @mapDirection and @mapClosure in folog & up)

                in bindatalog, datalog & up to (including) hornlog and dishornlog:
                ( oid?, (declare | Var)+, (formula | Atom | And | Or | Exists) )

                in hornlogeq: ( oid?, (declare | Var)+, (formula | Atom | And | Or | Exists | Equal) )

                in hohornlog & up: ( oid?, (declare | Var)+, (formula | Hterm | And | Or | Exists)

                in framehohornlogeq:
                ( oid?, (declare | Var)+, (formula | Atom|InstanceOf|SubclassOf|Hterm|And|Or|Exists) )

                in folog and naffolog:
                ( oid?, (declare | Var)+, (formula|Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists) )

                in fologeq and naffologeq:
                ( oid?, (declare | Var)+, (formula|Atom|And|Or|Neg|Implies|Equivalent|Forall|Exists|Equal) )

        *** declare ***

                in all sublanguages: ( Var )


        *** formula *** (see also the connective module)

        within Forall...

                in bindatalog, datalog & up to (including) hornlog and dishornlog:
                (Atom | Implies | Equivalent | Forall)

                in hornlogeq: ( Atom | Implies | Equivalent | Forall | Equal )

                in hohornlog & up: (Hterm | Implies | Equivalent | Forall)

                in framehohornlogeq: (Atom|InstanceOf|SubclassOf|Hterm|Implies|Equivalent|Forall) )

                in folog and naffolog: ( Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

                in fologeq and naffologeq: (Atom | And|Or | Neg | Implies | Equivalent | Forall|Exists | Equal)

        within Exists...

                in bindatalog, datalog & up to (including) hornlog and dishornlog: (Atom | And | Or | Exists)

                in hornlogeq: ( Atom | And | Or | Exists | Equal )

                in hohornlog & up: (Hterm | And | Or | Exists)

                in framehohornlogeq: (Atom | InstanceOf | SubclassOf | Hterm | And | Or | Exists)

                in folog and naffolog: ( Atom | And | Or | Neg | Implies | Equivalent | Forall | Exists )

                in fologeq and naffologeq: ( Atom | And|Or | Neg | Implies | Equivalent | Forall|Exists | Equal )
```

Rest - rest_module.xsd

```
*** repo ***

       in hornlog & up: (Var | Plex)

*** resl ***

       in hornlog & up: (Var | Plex)
```

Slot - slot_module.xsd

```
*** slot *** (context sensitive)
        attributes: @card, @weight ( + @minCard and @maxCard in framehohornlogeq)

within Atom, etc...

        in bindatalog, datalog & up to hornlog: ((Ind|Data|Skolem|Var|Reify),(Ind|Data|Skolem|Var|Reify))

        in bindatagroundlog and bindatagroundfact: ( (Ind|Data|Skolem|Reify),(Ind|Data|Skolem|Reify) )

        in hornlog & up (except hohornlog, etc):
        ((Ind|Data|Skolem|Var|Reify|Cterm|Plex), (Ind|Data|Skolem|Var|Reify|Cterm|Plex))

        in hohornlog & up: ( (Con|Skolem|Var|Reify|Hterm), (Con|Skolem|Var|Reify|Hterm) )

        in framehohornlogeq:
        ( (Con|Skolem|Var|Reify|Hterm|Get), (Con|Skolem|Var|Reify|Hterm|Get) )

within Atom-frame...

        in framehohornlogeq:
        ( (Con|Skolem|Var|Reify|Hterm|Get),(Con|Skolem|Var|Reify|Hterm|Get|Set)? )


*** @card *** [optional] nonNegativeInt


*** @minCard *** [optional] nonNegativeInt


*** @maxCard *** [optional] nonNegativeInt


*** @weight *** [optional] decimal [0,1]
```

Term - term_module.xsd

```
*** arg ***
       attributes: @index

       in bindatalog, datalog & up to hornlog: ( Ind | Data | Skolem | Var | Reify)

       in bindatagroundlog and bindatagroundfact: (Ind | Data | Skolem | Reify)

       in hornlog & up (except hohornlog, etc): (Ind | Data | Skolem | Var | Reify | Cterm | Plex)

       in hohornlog & up: (Con | Skolem | Var | Reify | Hterm)


*** Ind *** (context sensitive)

       in all sublanguages: (#PCDATA)


within oid...
       attributes: @wlab, @type

not within oid...
       attributes: @wref, @type


*** Data ***

       in all sublanguages: (#PCDATA) [optionally datatyped with XSD built-ins]


*** Var ***
       attributes: @type

       in all sublanguages: (#PCDATA)


*** Skolem ***
       attributes: @type

       in all sublanguages: (#PCDATA)


*** Reify ***

       in all sublanguages: ( <xs:any>? )


*** @type *** [optional] string


*** @index *** [required] positiveInt
```

Ur - ur_module.xsd

```
*** @wref *** [optional] anyURI

*** @wlab *** [optional] anyURI
```