Content Models for RuleML

This document is a collection of content models for all RuleML tags as of version 0.88 (2005-03-01), 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/0.88/xsd/modules/ for the actual XML schemas of the modules.

atom_module.xsd

-------------

*** Atom *** attributes: @closure

content model:
{
  oid?,
  { (opr | Rel), (slot)*, (arg | Ind | Var)*, (slot)* } |
  ( (slot)*, (arg | Ind | Var)*, (slot)*, opr )
}

however, this is non-deterministic, so it is (equivalently) restructured as follows:
{
  oid?,
  { (opr | Rel),
    (slot)*,
    { (arg | Ind | Var)*, (slot)*? |
    |
    {
      { (slot)*,
        { (arg | Ind | Var)*, (slot)*? |
        |
        { (arg | Ind | Var)*, (slot)* |
        },
    opr |
  }
}
}
in hornlog:
{
  oid?,
  { (opr | Rel), (slot)*, (arg | Ind | Var | Cterm | Plex)*, (slot)* } |
  ( (slot)*, (arg | Ind | Var | Cterm | Plex)+, (slot)*, opr )
}
in bindatalog and urcbindatalog:
{
  oid?,
  { (opr | Rel), (slot)*, (arg | Ind | Var), (arg | Ind | Var), (slot)* } |
  ( (slot)*, (arg | Ind | Var), (arg | Ind | Var), (slot)*, opr )
}
in urcbindatagroundlog:
{
  oid?,
  { (opr | Rel), (slot)*, (arg | Ind), (arg | Ind), (slot)* } |
  ( (slot)*, (arg | Ind), (arg | Ind), (slot)*, opr )
}

*** opr *** content model: (Rel)

*** Rel *** attributes: @wref (in ur sublanguages)

content model: (#PCDATA)
**connective_module.xsd**

---

*** Implies ***
attributes: @closure

content model: ( oid?, ( head, body) | ( body, head) | ( Atom | And | Or), Atom )

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

  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 ) )

*** body ***
content model: (Atom | And | Or)

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

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

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

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

*** head ***
content model: (Atom)

  in equalalog: (Atom | Equal)

  in negdatalog: (Atom | Neg)

  in nafnegdatalog: (Atom | Neg)

*** Equivalent ***
attributes: @closure

content model: ( oid?, ( torso, torso) | ( Atom, Atom ) )

*** torso ***
content model: (Atom)

*** And ***
attributes: @direction and @innerclose (below Assert), @closure (below Query)

content model (below Assert): ( oid?, (formula | Atom | Implies | Equivalent | Forall)* )

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

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

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

content model (below Query, Implies, body, And/Or): ( (formula | Atom | And | Or)* )

  in equalalog: ( (formula | Atom | And | Or | Equal)* )

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

  in nafnegdatalog: ( (formula | Atom | And | Or | Naf)* )

*** Or ***
attributes: @closure (below Query)

content model: ( (formula | Atom | And | Or)* )

  in equalalog: ( (formula | Atom | And | Or | Equal)* )

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

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

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

content model (below top level And): ( Atom | Implies | Equivalent | Forall )

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

  in urcbindatagroundlog: ( Atom | Implies | Equivalent )

  in urcbindatagroundfact: ( Atom )

content model (below inner And/Or): (Atom | And | Or)

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

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

  in nafnegdatalog: (Atom And Or Naf)

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

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

*** @closure *** [optional] (universal | existential)
cterm_module.xsd

-------------

*** Cterm ***
attributes: @type

content model:
{
  ( (opc | Ctor), (slot)*, (arg | Ind | Var | Cterm | Plex)*, (slot)* ) |
  ( (slot)*, (arg | Ind | Var | Cterm | Plex)+, (slot)*, opc )
}

however, this is non-deterministic, so it is (equivalently) restructured as follows:
{
  ( (opc | Ctor),
    (slot)*, 
    ( (arg | Ind | Var | Cterm | Plex)+, (slot)* )?
  ) |
  |
  ( (slot)+, 
    ( (arg | Ind | Var | Cterm | Plex)+, (slot)* )?
  ) |
  ( (arg | Ind | Var | Cterm | Plex)+, (slot)* 
  ),
  opc
}

*** opc ***
content model: (Ctor)

*** Ctor ***
attributes: @wref (in ur sublanguages)
content model: (#PCDATA)

*** Plex ***
content model: ( (slot)*, (arg | Ind | Var | Cterm | Plex)*, (slot)* )

however, this is non-deterministic, so it is (equivalently) restructured as follows:
( (slot)*, ( (arg | Ind | Var | Cterm | Plex)*, (slot)* )? )

desc_module.xsd

-------------

*** oid ***
content model: (Ind)
in hornlog: (Ind | Cterm)

equality_module.xsd

-------------

*** Equal ***
content model: ( (side | Ind | Var | Cterm | Nano), (side | Ind | Var | Cterm | Nano) )

*** side ***
content model: ( Ind | Var | Cterm | Nano )

*** Nano ***
content model: ( ( (opf | Fun), (arg | Ind | Var | Cterm)* ) | ((arg | Ind | Var | Cterm)+, opf )

*** opf ***
content model: (Fun)

*** Fun ***
attributes: @wref (in ur sublanguages)
content model: (#PCDATA)
negation_module.xsd

*** Neg ***
content model: (strong | Atom)

*** strong ***
content model: (Atom)

*** Naf ***
content model: (weak | Atom)
in nafnegdatalog: (weak | Atom | Neg)

*** weak ***
content model: (Atom)
in nafnegdatalog: (Atom | Neg)

performative_module.xsd

*** Assert ***
content model: (content | And)

*** Query ***
content model: (content | Atom | And | Or | Exists)
in equalog: (content | Atom | And | Or | Exists | Equal)
in negdatalog: (content | Atom | And | Or | Exists | Neg)
in nafdatalog: (content | Atom | And | Or | Exists | Naf)
in urcbindinggroundlog: (content | Atom | And | Or)

*** content ***
content model (below Assert): (And)
content model (below Query): (Atom | And | Or | Exists)
in equalog: (Atom | And | Or | Exists | Equal)
in negdatalog: (Atom | And | Or | Exists | Neg)
in nafdatalog: (Atom | And | Or | Exists | Naf | Neg)
in urcbindinggroundlog: (Atom | And | Or)

quantifier_module.xsd

*** Forall ***
content model: (oid?, (declare | Var)*, (formula | Atom | Implies | Equivalent | Forall))

*** Exists ***
content model: (oid?, (declare | Var)*, (formula | Atom | And | Or | Exists))

*** declare ***
content model: (Var)

*** formula *** (see also the connective module)
content model (below Forall): (Atom | Implies | Equivalent | Forall)
content model (below Exists): (Atom | And | Or | Exists)

slot_module.xsd

*** slot ***
attributes: @card and @weight
content model: (Ind | Var), (Ind | Var)
in hornlog: (Ind | Var | Cterm | Plex), (Ind | Var | Cterm | Plex)
in urcbindinggroundlog: (Ind, Ind)

*** @card *** [optional] nonNegativeInt
*** @weight *** [optional] decimal [0,1]
term_module.xsd
--------------

*** arg ***
attributes: @index
content model: ( Ind | Var )
in hornlog: (Ind | Var | Cterm | Plex)
in urcbindatagroundlog: ( Ind, Ind )

*** Ind ***
attributes: @type, @wref and @wlab (in ur sublanguages, @wlab only when oid is its parent)
content model: (#PCDATA)

*** Var ***
attributes: @type
content model: (#PCDATA)

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

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

ur_module.xsd
--------------

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

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