Schema documentation for morph.xsd

Publication date 13 mai 2011

Table of Contents

Namespace: ""

Schemas

Main schema `morph.xsd`

Namespace	No namespace
Annotations	Schema for morph file. A morph file lists all the known word forms (morph items) together with their stems, parts of speech, semantic classes, associated macros, and excluded lexical categories. The part of speech is used to help determine the mapping between morph items and lexical categories. A morph item is automatically associated with all open families with matching parts of speech. With closed families, the morph item's stem must be listed as a family member, and the parts of speech must match. It is also possible to exclude certain lexical category entries or families, by including the entry name, qualified entry name or family name in the morph item's list of excluded lexical categories. NB: A limitation of the current implementation is that the mapping between morph items and open families can only be done with the stem as the predicate; to use different predicates, closed families must be used. Semantic classes may also be given, for n-gram ranking purposes, and for restricting the unification of nominals to compatible types, specified in the types file. When a category is instantiated, the semantic class is assigned to the nominal var(s) for the proposition with the reserved name '[*DEFAULT*]'. The types of all nominal vars are then propagated to all other nominal vars with the same name, throughout the category. The macros are used to add features or semantic predications to a lexical category that depend on the particular morph item (e.g. tense, number, case, etc.).
Properties	attribute form default: unqualified element form default: unqualified
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Included schema `categories.xsd`

Namespace	No namespace
Annotations	Elements and types for categories and feature structures. By and large, there are no constraints on the names used in defining categories, with just a few exceptions: - '[*DEFAULT*]' proposition or feature value: this reserved value is replaced with the predicate associated with a lexical item (defaulting to the stem) when instantiating the lexical categories associated with a word - 'index' feature: this feature receives special treatment in the realizer; see Mike and Jason's paper on the realizer for details, at http://www.iccs.informatics.ed.ac.uk/~mwhite/White-Baldridge-ENLG-2003-to-appear.pdf - lists: lists are defined with the 'First' and 'Last' relations, which is (optionally) converted to just the 'List' relation by simplify-lists.xsl; elements in the list are represented using nodes with the 'elem' predicate, together with 'Item' and 'Next' relations to point to the actual list item and the next element, respectively

	- tuples: pairs (for argument clusters and gapping) are represented using nodes with the 'tup' (for tuple) predicate, together with 'Item1' and 'Item2' relations to point to the paired items (in principle, further 'ItemN' relations could be used for tuples of length greater than 2); paired items receive special treatment in the realizer - 'BoundVar' relation: this relation indicates that the subordinate nominal is a bound var; as a result, feature-based instantiation is disabled in the realizer - 'mark' feature: this semantic attribute indicates that the realizer should label the phrase in the XML output headed by the index associated with this feature
Properties	attribute form default: unqualified element form default: unqualified
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Included schema `tokens.xsd`

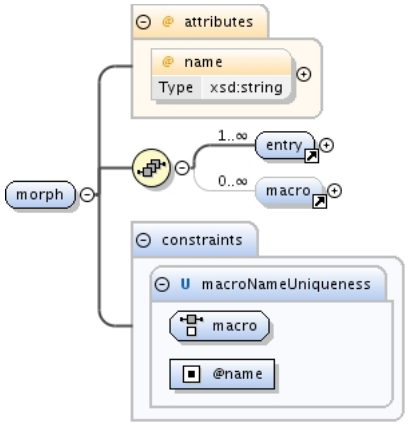
Namespace	No namespace
Annotations	Definitions for tokens.
Properties	attribute form default: unqualified element form default: unqualified
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/tokens.xsd

Included schema `hlds.xsd`

Namespace	No namespace
Annotations	Elements and types for hybrid logic dependency semantics (HLDS) constructs. The subset of hybrid logic terms used in HLDS is defined here. (The Java classes currently support a slightly larger subset.) To support flattening of these terms into elementary predications, the contents of the satisfaction operators and diamond relations (modal ops) are restricted, as described in detail below. Note that where a list of terms is allowed, these are implicitly conjoined; a conjunction operator is inserted upon loading, when necessary. LFs with disjunctive or optionality operators may also be specified. Nominals and variables may be given a type (or sort) listed in the types file, by appending the type to the name after a colon. Propositions whose names appear in the types file are also treated as typed.
Properties	attribute form default: unqualified element form default: unqualified
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Elements

Element `morph`

Namespace	No namespace
Annotations	The root element, containing a list of entries (morph items) and macros. Each macro must have a unique name.
Diagram	

Properties	content: complex
Model	entry+ , macro*
Children	entry, macro
Instance	<morph name=""> <entry class="" coart="" excluded="" macros="" pos="" stem="" word="">{1,unbounded}</entry> <macro name="">{0,unbounded}</macro> </morph>

Attributes

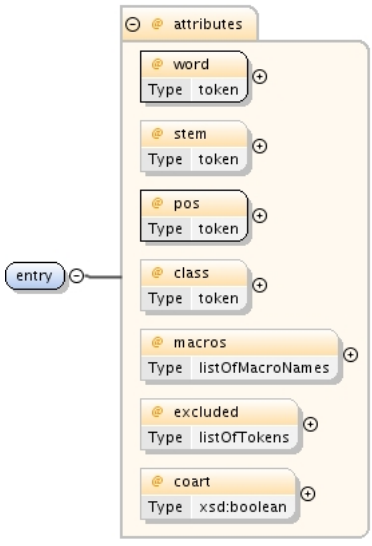
QName	Type	Fixed	Default	Use
name	xsd:string			optional

Source

```
<xsd:element name="morph">
  <xsd:annotation>
    <xsd:documentation>The root element, containing a list of entries (morph items) and
 macros. Each macro must have a unique name.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="entry" maxOccurs="unbounded"/>
      <xsd:element ref="macro" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string"/>
  </xsd:complexType>
  <xsd:unique name="macroNameUniqueness">
    <xsd:selector xpath="macro"/>
    <xsd:field xpath="@name"/>
  </xsd:unique>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Element entry

Namespace	No namespace
Annotations	A morph item, with its word form, stem, part of speech, and optional semantic class, associated macros, and excluded lexical categories. The stem defaults to word form. At run time, when the grammar is loaded, a cross-reference check is performed on the names in the lists of associated macros and excluded lexical categories. The coart flag indicates that this entry is a coarticulation, eg a pitch accent, gesture, or other word-associated element. With coarticulations, the word form should be one or more attribute-value pairs, with attributes and values separated by hyphens, and multiple pairs separated by colons. When multiple attribute-value pairs are given, only the first one is used for indexing purposes. An identifying value for the stem should also be given, eg *accent* for a pitch accent. NB: Lexical attributes supplied by coarticulations must be done so uniformly; that is, they cannot appear already on some lexical items. Also, when there are multipled, independent coarticulations, they must supply disjoint sets of lexical attributes, and their categories must be capable of applying in any order.
Diagram	

Properties	content:	complex
Used by	Element	morph

Attributes

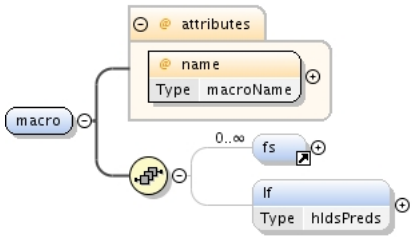
QName	Type	Fixed	Default	Use
class	token			optional
coart	xsd:boolean			optional
excluded	listOfTokens			optional
macros	listOfMacroNames			optional
pos	token			required
stem	token			optional
word	token			required

Source

```
<xsd:element name="entry">
  <xsd:annotation>
    <xsd:documentation>A morph item, with its word form, stem, part of speech, and
 optional semantic class, associated macros, and excluded lexical categories. The stem
 defaults to word form. At run time, when the grammar is loaded, a cross-reference
 check is performed on the names in the lists of associated macros and excluded lexical
 categories. The coart flag indicates that this entry is a coarticulation, eg a pitch
 accent, gesture, or other word-associated element. With coarticulations, the word
 form should be one or more attribute-value pairs, with attributes and values separated
 by hyphens, and multiple pairs separated by colons. When multiple attribute-value
 pairs are given, only the first one is used for indexing purposes. An identifying
 value for the stem should also be given, eg *accent* for a pitch accent. NB: Lexical
 attributes supplied by coarticulations must be done so uniformly; that is, they cannot
 appear already on some lexical items. Also, when there are multipled, independent
 coarticulations, they must supply disjoint sets of lexical attributes, and their
 categories must be capable of applying in any order.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="word" type="token" use="required"/>
    <xsd:attribute name="stem" type="token"/>
    <xsd:attribute name="pos" type="token" use="required"/>
    <xsd:attribute name="class" type="token"/>
    <xsd:attribute name="macros" type="listOfMacroNames"/>
    <xsd:attribute name="excluded" type="listOfTokens"/>
    <xsd:attribute name="coart" type="xsd:boolean"/>
  </xsd:complexType>
</xsd:element>
```

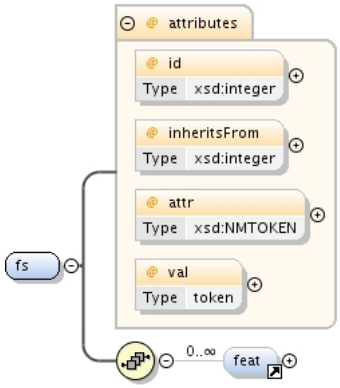
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Element macro

Namespace	No namespace
Annotations	A named macro, with a set of features and/or semantic predications to add to a lexical category. The features are given by feature structure declarations, where the id is used to indicate where the features are to be added (i.e., to the feature structure of which atomic category). The semantic predications are given in an 'lf' element; at run time, these HLDS predications are flattened and added to any predications already present in the lexical category.
Diagram	
Properties	content: complex
Used by	Element morph
Model	fs* , lf{0,1}
Children	fs, lf

Instance	<macro name=""> <fs attr="" id="" inheritsFrom="" val="">{0,unbounded}</fs> <lf>{0,1}</lf> </macro>
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **name** | macroName | | | required
Source	<xsd:element name="macro"> <xsd:annotation> <xsd:documentation>A named macro, with a set of features and/or semantic predications to add to a lexical category. The features are given by feature structure declarations, where the id is used to indicate where the features are to be added (i.e., to the feature structure of which atomic category). The semantic predications are given in an 'lf' element; at run time, these HLDS predications are flattened and added to any predications already present in the lexical category.</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element ref="fs" minOccurs="0" maxOccurs="unbounded"/> <xsd:element name="lf" type="hldsPreds" minOccurs="0"/> </xsd:sequence> <xsd:attribute name="name" type="macroName" use="required"/> </xsd:complexType> </xsd:element>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Element fs

Namespace	No namespace
Annotations	A simple, non-recursive feature structure, consisting of a set of features, i.e. attribute-value pairs. If there is only a single, string-valued feature, it can be specified using the "attr" and "val" attributes on this element.
Diagram	
Properties	content: complex
Used by	Elements: atomcat, macro
Model	feat*
Children	feat
Instance	<fs attr="" id="" inheritsFrom="" val=""> <feat attr="" val="">{0,unbounded}</feat> </fs>
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **attr** | xsd:NMTOKEN | | | optional | The name of the single, string-valued feature (when appropriate). **id** | xsd:integer | | | optional | An integer id for the feature structure. The id is used for coindexation, "inheritsFrom" feature propagation, and macro access. **inheritsFrom** | xsd:integer | | | optional | This attribute is used to specify feature propagation with exceptions (i.e., default unification). At run time, feature equations are added in order to propagate feature values.

QName	Type	Fixed	Default	Use
	In particular, a feature variable is added to this feature structure and a corresponding one is added to the referenced feature structure, for all appropriate attributes except those with explicit values already on this feature structure. The appropriate attributes are determined by the type of the atomic category, and consist of all the attributes seen with this category type in the lexicon file. NB: Attributes appearing in macros in the morph file are not included, since their association with category types is not available statically.			
val	token			optional
	The value of the feature.			

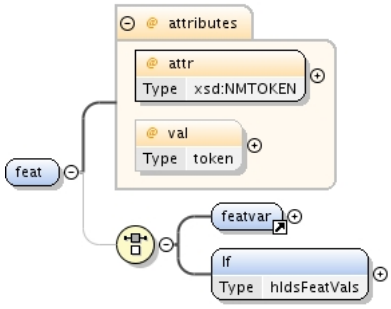
Source

```
<xsd:element name="fs">
  <xsd:annotation>
    <xsd:documentation>A simple, non-recursive feature structure, consisting of a set
 of features, i.e. attribute-value pairs. If there is only a single, string-valued
 feature, it can be specified using the "attr" and "val" attributes on this element.</
xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="feat" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:integer">
      <xsd:annotation>
        <xsd:documentation>An integer id for the feature structure. The id is used for
 coindexation, "inheritsFrom" feature propagation, and macro access.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="inheritsFrom" type="xsd:integer">
      <xsd:annotation>
        <xsd:documentation>This attribute is used to specify feature propagation with
 exceptions (i.e., default unification). At run time, feature equations are added in
 order to propagate feature values. In particular, a feature variable is added to this
 feature structure and a corresponding one is added to the referenced feature structure,
 for all appropriate attributes except those with explicit values already on this feature
 structure. The appropriate attributes are determined by the type of the atomic category,
 and consist of all the attributes seen with this category type in the lexicon file.
 NB: Attributes appearing in macros in the morph file are not included, since their
 association with category types is not available statically.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="attr" type="xsd:NMTOKEN">
      <xsd:annotation>
        <xsd:documentation>The name of the single, string-valued feature (when
 appropriate).</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="val" type="token">
      <xsd:annotation>
        <xsd:documentation>The value of the feature.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

Schema location: file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element feat

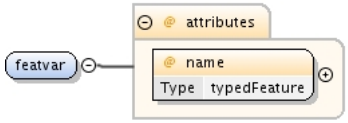
Namespace	No namespace
Annotations	A feature, i.e. an attribute-value pair. Features can be either syntactic or semantic (LF-valued). Syntactic features can be either ground or variable. If ground, they must be string-valued, and specified via the "val" attribute; if variable, the value is a variable named by a "featvar" element. Semantic features can also be either ground or variable; their possible values are given by the "hldsFeatVals" type in hlds.xsd.

Diagram	
Properties	content: complex
Used by	Element fs
Model	featvar | lf
Children	featvar, lf

Instance	`<feat attr="" val="">`
	`  <featvar name="">{1,1}</featvar>`
	`  <lf>{1,1}</lf>`
	`</feat>`

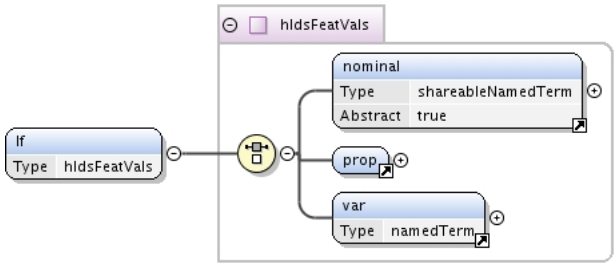
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **attr** | xsd:NMTOKEN | | | required The name of the feature. **val** | token | | | optional The value of the feature, when string-valued.
Source	`<xsd:element name="feat">` `  <xsd:annotation>` ` <xsd:documentation>A feature, i.e. an attribute-value pair. Features can be either syntactic or semantic (LF-valued). Syntactic features can be either ground or variable. If ground, they must be string-valued, and specified via the "val" attribute; if variable, the value is a variable named by a "featvar" element. Semantic features can also be either ground or variable; their possible values are given by the "hldsFeatVals" type in hlds.xsd.</xsd:documentation>` ` </xsd:annotation>` ` <xsd:complexType>` ` <xsd:choice minOccurs="0">` ` <xsd:element ref="featvar"/>` ` <xsd:element name="lf" type="hldsFeatVals"/>` ` </xsd:choice>` ` <xsd:attribute name="attr" type="xsd:NMTOKEN" use="required">` ` <xsd:annotation>` ` <xsd:documentation>The name of the feature.</xsd:documentation>` ` </xsd:annotation>` ` </xsd:attribute>` ` <xsd:attribute name="val" type="token">` ` <xsd:annotation>` ` <xsd:documentation>The value of the feature, when string-valued.</xsd:documentation>` ` </xsd:annotation>` ` </xsd:attribute>` ` </xsd:complexType>` `</xsd:element>`
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element `featvar`

Namespace	No namespace
Annotations	A variable over syntactic feature values.
Diagram	
Properties	content: complex
Used by	Element feat

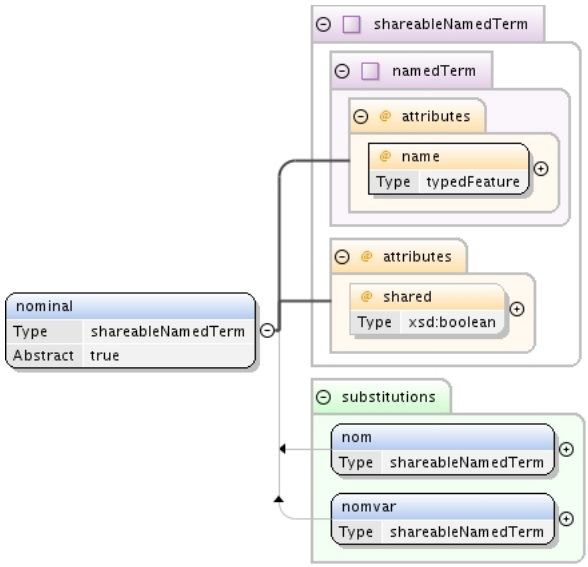
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	name	typedFeature			required
		The name of the feature variable.			
Source	<xsd:element name="featvar"> <xsd:annotation> <xsd:documentation>A variable over syntactic feature values.</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:attribute name="name" type="typedFeature" use="required"> <xsd:annotation> <xsd:documentation>The name of the feature variable.</xsd:documentation> </xsd:annotation> </xsd:attribute> </xsd:complexType> </xsd:element>				
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd				

Element feat/lf

Namespace	No namespace
Diagram	
Type	hldsFeatVals
Properties	content: complex
Model	nominal | prop | var
Children	nominal, prop, var
Instance	<lf> <nominal name="" shared="">{1,1}</nominal> <prop name="">{1,1}</prop> <var name="">{1,1}</var> </lf>
Source	<xsd:element name="lf" type="hldsFeatVals"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element nominal

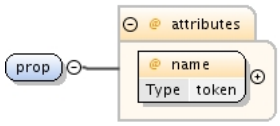
Namespace	No namespace
Annotations	A nominal term, either an atom or a variable. The "shared" attribute indicates that the nominal term provides a reference to a node that is a shared part of multiple alternatives, rather than being a subordinated or coordinated reference.

Diagram	
Type	shareableNamedTerm
Type hierarchy	• namedTerm • shareableNamedTerm
Properties	content: `complex` abstract: `true`
Used by	Elements: diamond, op Complex Type: hldsFeatVals
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **name** | typedFeature | | | required **shared** | xsd:boolean | | | optional

Source	`<xsd:element name="nominal" type="shareableNamedTerm" abstract="true">`
	`  <xsd:annotation>`
	`  <xsd:documentation>A nominal term, either an atom or a variable. The "shared"`
	`attribute indicates that the nominal term provides a reference to a node that is a`
	`shared part of multiple alternatives, rather than being a subordinated or coordinated`
	`reference.</xsd:documentation>`
	`  </xsd:annotation>`
	`</xsd:element>`

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element `prop`

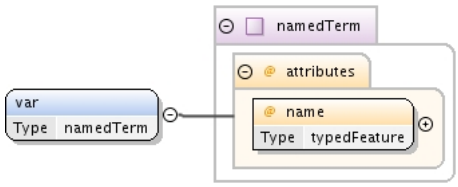
Namespace	No namespace
Annotations	`A named proposition.`
Diagram	
Properties	content: `complex`
Used by	Elements: diamond, satop Complex Type: hldsFeatVals
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **name** | token | | | required

Source	`<xsd:element name="prop">`
	`  <xsd:annotation>`
	`  <xsd:documentation>A named proposition.</xsd:documentation>`
	`  </xsd:annotation>`

```
  <xsd:complexType>
    <xsd:attribute name="name" type="token" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element `var`

Namespace	No namespace
Annotations	`A variable over HLDS terms.`
Diagram	
Type	namedTerm
Properties	content: `complex`
Used by	Element: diamond Complex Type: hldsFeatVals

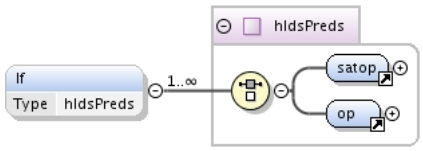
Attributes	QName	Type	Fixed	Default	Use
	name	typedFeature			required

Source

```
<xsd:element name="var" type="namedTerm">
  <xsd:annotation>
    <xsd:documentation>A variable over HLDS terms.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element `macro/lf`

Namespace	No namespace
Diagram	
Type	hldsPreds
Properties	content: `complex` minOccurs: `0`
Model	satop | op
Children	op, satop

Instance

```
<lf>
  <satop nom="" nomvar="" shared="">{1,1}</satop>
  <op name="">{1,1}</op>
</lf>
```

Source

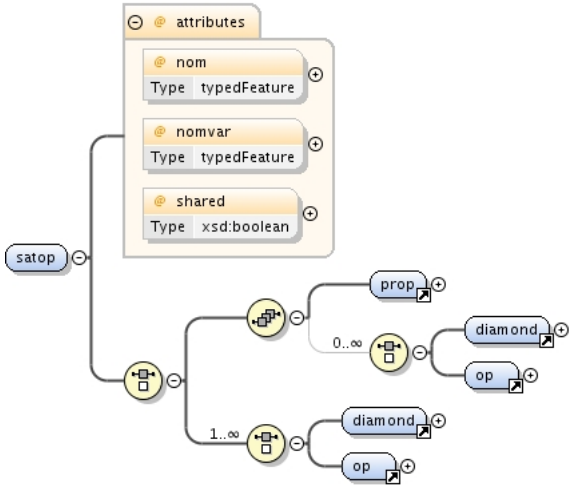
```
<xsd:element name="lf" type="hldsPreds" minOccurs="0"/>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Element `satop`

Namespace	No namespace

Annotations	`A satisfaction operator, with a nominal and an arg.`
	`The nominal can either be a nominal atom, named by the "nom" attribute,`
	`or a nominal variable, named by the "nomvar" attribute.`
	`The nominal may be optionally marked as "shared" (see nominal def below).`
	`The arg optionally begins with a proposition, followed by any number`
	`of diamond relations or disjunctive/optionality operators,`

	but must be non-empty.
Diagram	
Properties	content: complex
Used by	Element: op Complex Type: hldsPreds
Model	(prop , (diamond | op)) | diamond | op
Children	diamond, op, prop

Instance	`<satop nom="" nomvar="" shared="">`
	`  <prop name="">{1,1}</prop>`
	`</satop>`

Attributes

QName	Type	Fixed	Default	Use
nom	typedFeature			optional
nomvar	typedFeature			optional
shared	xsd:boolean			optional

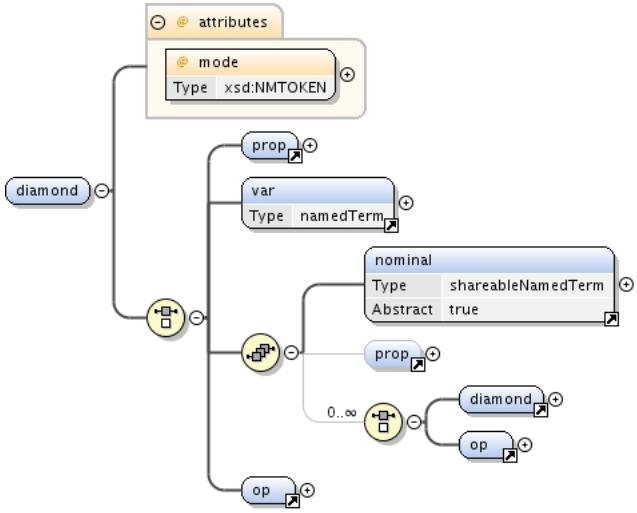
Source

```
<xsd:element name="satop">
  <xsd:annotation>
    <xsd:documentation>A satisfaction operator, with a nominal and an arg. The nominal
 can either be a nominal atom, named by the "nom" attribute, or a nominal variable,
 named by the "nomvar" attribute. The nominal may be optionally marked as "shared" (see
 nominal def below). The arg optionally begins with a proposition, followed by any number
 of diamond relations or disjunctive/optionality operators, but must be non-empty.</
xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:choice>
      <xsd:sequence>
        <xsd:element ref="prop"/>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element ref="diamond"/>
          <xsd:element ref="op"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element ref="diamond"/>
        <xsd:element ref="op"/>
      </xsd:choice>
    </xsd:choice>
    <xsd:attribute name="nom" type="typedFeature"/>
    <xsd:attribute name="nomvar" type="typedFeature"/>
    <xsd:attribute name="shared" type="xsd:boolean"/>
  </xsd:complexType>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element diamond

Namespace	No namespace
Annotations	A diamond relation (modal operator), with a mode and an arg. The "mode" attribute names the relation. The arg is either just a proposition or a variable,

	or a nominal followed optionally by a proposition and any number of nested diamond relations or disjunctive/optionality operators. In the former case, the relation encodes a semantic feature; in the latter case, it encodes a relation to another semantic head. The arg may also be a disjunctive operator containing satops.
Diagram	
Properties	content: complex
Used by	Elements diamond, op, satop
Model	prop | var | (nominal , prop{0,1} , (diamond | op)) | op
Children	diamond, nominal, op, prop, var

Instance	<pre><diamond mode="">
	  <prop name="">{1,1}</prop>
	  <var name="">{1,1}</var>
	  <op name="">{1,1}</op>
	</diamond></pre>

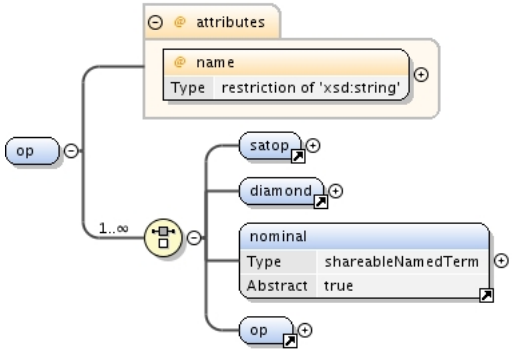
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **mode** | xsd:NMTOKEN | | | required

Source	<pre><xsd:element name="diamond">
	  <xsd:annotation>
	  <xsd:documentation>A diamond relation (modal operator), with a mode and an arg. The
	"mode" attribute names the relation. The arg is either just a proposition or a variable,
	or a nominal followed optionally by a proposition and any number of nested diamond
	relations or disjunctive/optionality operators. In the former case, the relation encodes
	a semantic feature; in the latter case, it encodes a relation to another semantic head.
	The arg may also be a disjunctive operator containing satops.</xsd:documentation>
	  </xsd:annotation>
	  <xsd:complexType>
	  <xsd:choice>
	  <xsd:element ref="prop"/>
	  <xsd:element ref="var"/>
	  <xsd:sequence>
	  <xsd:element ref="nominal"/>
	  <xsd:element ref="prop" minOccurs="0"/>
	  <xsd:choice minOccurs="0" maxOccurs="unbounded">
	  <xsd:element ref="diamond"/>
	  <xsd:element ref="op"/>
	  </xsd:choice>
	  </xsd:sequence>
	  <xsd:element ref="op"/>
	  </xsd:choice>
	  <xsd:attribute name="mode" type="xsd:NMTOKEN" use="required"/>
	  </xsd:complexType>
	</xsd:element></pre>

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element op

Namespace	No namespace
Annotations	A conjunction (conj), exclusive disjunction (xor) or optionality (opt) operator.

	Note that conjunctions are usually left implicit.
Diagram	
Properties	content: complex
Used by	Elements: diamond, op, satop Complex Type: hldsPreds
Model	satop | diamond | nominal | op
Children	diamond, nominal, op, satop

Instance	`<op name="">`
	`  <satop nom="" nomvar="" shared="">{1,1}</satop>`
	`  <diamond mode="">{1,1}</diamond>`
	`  <nominal name="" shared="">{1,1}</nominal>`
	`  <op name="">{1,1}</op>`
	`</op>`

Attributes

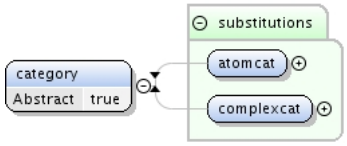
QName	Type	Fixed	Default	Use
name	restriction of xsd:string			required

Source

```
<xsd:element name="op">
  <xsd:annotation>
    <xsd:documentation>A conjunction (conj), exclusive disjunction (xor) or optionality
 (opt) operator. Note that conjunctions are usually left implicit.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="satop"/>
      <xsd:element ref="diamond"/>
      <xsd:element ref="nominal"/>
      <xsd:element ref="op"/>
    </xsd:choice>
    <xsd:attribute name="name" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="conj"/>
          <xsd:enumeration value="xor"/>
          <xsd:enumeration value="opt"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

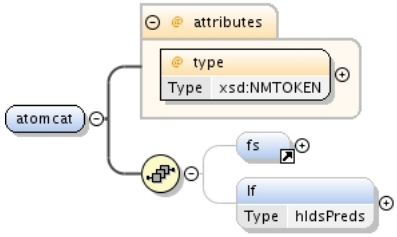
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element category

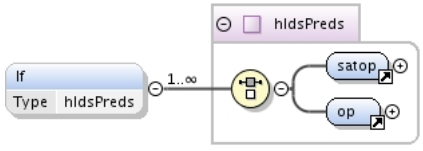
Namespace	No namespace
Annotations	A category, either atomic or complex.
Diagram	
Properties	abstract: true
Used by	Element: complexcat

	Element Group basicArg
Source	<xsd:element name="category" abstract="true"> <xsd:annotation> <xsd:documentation>A category, either atomic or complex.</xsd:documentation> </xsd:annotation> </xsd:element>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element atomcat

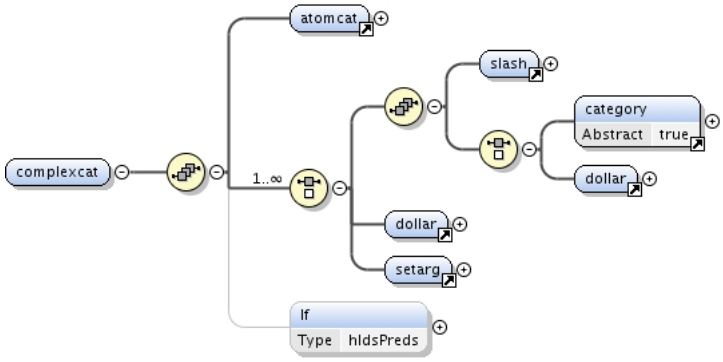
Namespace	No namespace
Annotations	An atomic category. Atomic categories must have a type, and may contain a feature structure and LF. The possible LF predications are given by the "hldsPreds" type in hlds.xsd.
Diagram	
Properties	content: complex
Used by	Element complexcat
Model	fs{0,1} , lf{0,1}
Children	fs, lf
Instance	<atomcat type=""> <fs attr="" id="" inheritsFrom="" val="">{0,1}</fs> <lf>{0,1}</lf> </atomcat>
Attributes	**QName** **Type** **Fixed** **Default** **Use** **type** xsd:NMTOKEN required The type of the category, e.g. "np".
Source	<xsd:element name="atomcat" substitutionGroup="category"> <xsd:annotation> <xsd:documentation>An atomic category. Atomic categories must have a type, and may contain a feature structure and LF. The possible LF predications are given by the "hldsPreds" type in hlds.xsd.</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element ref="fs" minOccurs="0"/> <xsd:element name="lf" type="hldsPreds" minOccurs="0"/> </xsd:sequence> <xsd:attribute name="type" type="xsd:NMTOKEN" use="required"> <xsd:annotation> <xsd:documentation>The type of the category, e.g. "np".</xsd:documentation> </xsd:annotation> </xsd:attribute> </xsd:complexType> </xsd:element>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element atomcat/lf

Namespace	No namespace
Diagram	
Type	hldsPreds

Properties	content: complex minOccurs: 0
Model	satop | op
Children	op, satop
Instance	<lf> <satop nom="" nomvar="" shared="">{1,1}</satop> <op name="">{1,1}</op> </lf>
Source	<xsd:element name="lf" type="hldsPreds" minOccurs="0"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element complexcat

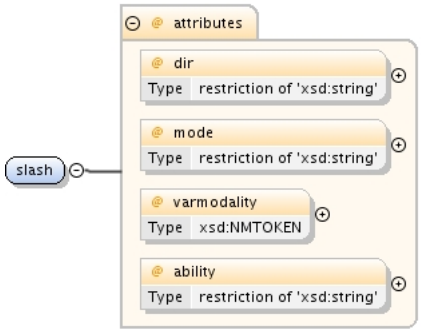
Namespace	No namespace
Annotations	A complex category, consisting of a target category, an argument stack, and an optional LF. Since a complex cat is not curried, the target category will always be atomic. An argument stack is one or more basic args, dollar args or set args.
Diagram	
Properties	content: complex
Model	atomcat , ((slash , (category | dollar)) | dollar | setarg) , lf{0,1}
Children	atomcat, category, dollar, lf, setarg, slash
Instance	<complexcat> <atomcat type="">{1,1}</atomcat> <lf>{0,1}</lf> </complexcat>

Source	<xsd:element name="complexcat" substitutionGroup="category">
	  <xsd:annotation>
	  <xsd:documentation>A complex category, consisting of a target category, an argument
	 stack, and an optional LF. Since a complex cat is not curried, the target category will
	 always be atomic. An argument stack is one or more basic args, dollar args or set args.</
	xsd:documentation>
	  </xsd:annotation>
	  <xsd:complexType>
	  <xsd:sequence>
	  <!-- target cat -->
	  <xsd:element ref="atomcat"/>
	  <!-- arg stack -->
	  <xsd:choice maxOccurs="unbounded">
	  <!--
	  NB: An arg stack should be one or more basic, dollar or set args, as
	  shown in this comment below. Since the slash element is potentially
	  ambiguous between an element of a basic arg or a dollar arg, the slash
	  needs to be factored out, as in the actual definition below.
	
	  <xsd:group ref="basicArg"/>
	  <xsd:group ref="dollarArg"/>
	  <xsd:element ref="setarg"/>
	  -->
	  <xsd:sequence>
	  <xsd:element ref="slash"/>
	  <xsd:choice>
	  <xsd:element ref="category"/>
	  <xsd:element ref="dollar"/>
	  </xsd:choice>
	  </xsd:sequence>

```
        <xsd:element ref="dollar"/>
        <xsd:element ref="setarg"/>
      </xsd:choice>
      <!-- lf -->
      <xsd:element name="lf" type="hldsPreds" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element `slash`

Namespace	No namespace
Annotations	A slash in the definition of a complex category. A slash has a direction, a mode and an ability; a variable can also be given for the mode, using the "varmodality" attribute. For discussion, see Jason's dissertation, esp. Ch. 8 on the implementation; the dissertation is downloadable from http://www.iccs.inf.ed.ac.uk/~jmb/dissertation
Diagram	
Properties	content: complex

Used by

Element	complexcat
Element Groups	basicArg, dollarArg

Attributes

QName	Type	Fixed	Default	Use
ability	restriction of xsd:string			optional
	The ability of the slash, either inert or active. Defaults to either. This is used to implement antecedent government; see Jason's dissertation for details.			
dir	restriction of xsd:string			optional
	The direction of the slash. The direction can be forward (/), backward (\) or both (|). Defaults to both.			
mode	restriction of xsd:string			optional
	The mode of the slash. The possible values are: all (.), application only (*), associative (^), permutative (x), permutative right (x>), permutative left (<x), associative permutative right (>), and associative permutative left (<). Defaults to all. See Jason's dissertation for details.			
varmodality	xsd:NMTOKEN			optional
	A variable over modalities.			

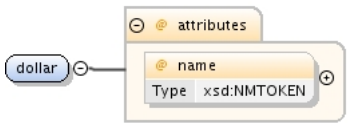
Source

```
<xsd:element name="slash">
  <xsd:annotation>
    <xsd:documentation>A slash in the definition of a complex category. A slash has a
 direction, a mode and an ability; a variable can also be given for the mode, using the
 "varmodality" attribute. For discussion, see Jason's dissertation, esp. Ch. 8 on the
 implementation; the dissertation is downloadable from http://www.iccs.inf.ed.ac.uk/~jmb/
dissertation</xsd:documentation>
```

```
  </xsd:annotation>
  <xsd:complexType>
    <xsd:attribute name="dir">
      <xsd:annotation>
        <xsd:documentation>The direction of the slash. The direction can be forward (/),
 backward (\) or both (|). Defaults to both.</xsd:documentation>
      </xsd:annotation>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="/"/>
          <xsd:enumeration value="\"/>
          <xsd:enumeration value="|"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="mode">
      <xsd:annotation>
        <xsd:documentation>The mode of the slash. The possible values are: all (.),
 application only (*), associative (^), permutative (x), permutative right (x>),
 permutative left (<x), associative permutative right (>), and associative permutative
 left (<). Defaults to all. See Jason's dissertation for details.</xsd:documentation>
      </xsd:annotation>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="."/>
          <xsd:enumeration value="*"/>
          <xsd:enumeration value="^"/>
          <xsd:enumeration value="x"/>
          <xsd:enumeration value="x>"/>
          <xsd:enumeration value="<x"/>
          <xsd:enumeration value=">"/>
          <xsd:enumeration value="<"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="varmodality" type="xsd:NMTOKEN">
      <xsd:annotation>
        <xsd:documentation>A variable over modalities.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="ability">
      <xsd:annotation>
        <xsd:documentation>The ability of the slash, either inert or active. Defaults to
 either. This is used to implement antecedent government; see Jason's dissertation for
 details.</xsd:documentation>
      </xsd:annotation>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="inert"/>
          <xsd:enumeration value="active"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element `dollar`

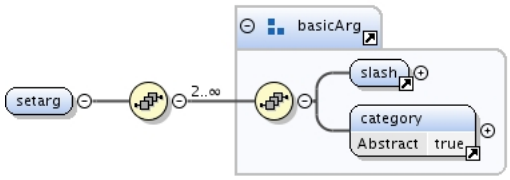
Namespace	No namespace
Annotations	`A dollar variable in the definition of a complex category, i.e. a variable over any sublist of args in an arg stack.`
Diagram	
Properties	content: `complex`
Used by	Element: complexcat Element Group: dollarArg

Attributes

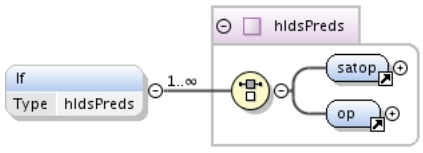
QName	Type	Fixed	Default	Use
name	xsd:NMTOKEN			optional
	`The name of the dollar variable, for coindexation purposes.`			

Source	<xsd:element name="dollar"> <xsd:annotation> <xsd:documentation>A dollar variable in the definition of a complex category, i.e. a variable over any sublist of args in an arg stack.</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:attribute name="name" type="xsd:NMTOKEN"> <xsd:annotation> <xsd:documentation>The name of the dollar variable, for coindexation purposes.</xsd:documentation> </xsd:annotation> </xsd:attribute> </xsd:complexType> </xsd:element>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element setarg

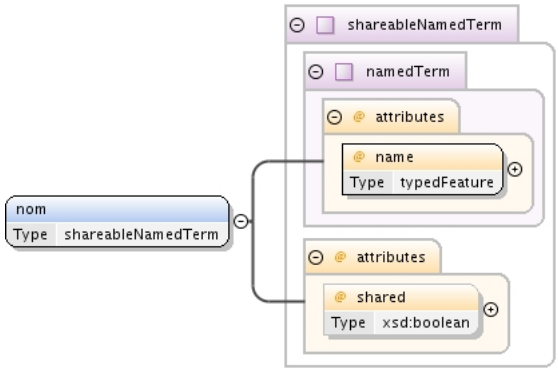
Namespace	No namespace
Annotations	A set arg, i.e. an unordered set of basic args.
Diagram	
Properties	content: complex
Used by	Element: complexcat
Model	slash , category
Children	category, slash
Instance	<setarg> <slash ability="" dir="" mode="" varmodality="">{1,1}</slash> <category>{1,1}</category> </setarg>
Source	<xsd:element name="setarg"> <xsd:annotation> <xsd:documentation>A set arg, i.e. an unordered set of basic args.</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:group ref="basicArg" minOccurs="2" maxOccurs="unbounded"/> </xsd:sequence> </xsd:complexType> </xsd:element>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element complexcat/lf

Namespace	No namespace
Diagram	
Type	hldsPreds
Properties	content: complex minOccurs: 0
Model	satop | op
Children	op, satop
Instance	<lf> <satop nom="" nomvar="" shared="">{1,1}</satop> <op name="">{1,1}</op> </lf>

Source	`<xsd:element name="lf" type="hldsPreds" minOccurs="0"/>`
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element nom

Namespace	No namespace
Annotations	A nominal atom.
Diagram	
Type	shareableNamedTerm
Type hierarchy	• namedTerm • shareableNamedTerm
Properties	content: complex

Attributes

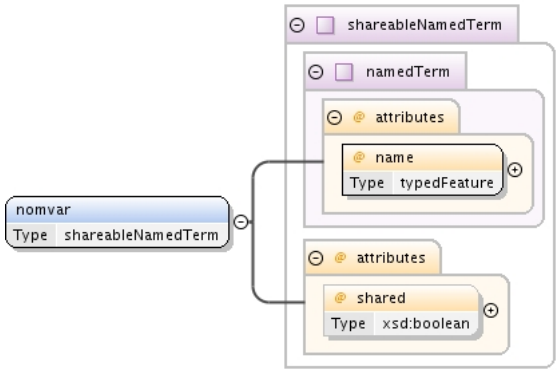
QName	Type	Fixed	Default	Use
name	typedFeature			required
shared	xsd:boolean			optional

Source

```
<xsd:element name="nom" type="shareableNamedTerm" substitutionGroup="nominal">
  <xsd:annotation>
    <xsd:documentation>A nominal atom.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Element nomvar

Namespace	No namespace
Annotations	A nominal variable.
Diagram	
Type	shareableNamedTerm
Type hierarchy	• namedTerm • shareableNamedTerm
Properties	content: complex

Attributes

QName	Type	Fixed	Default	Use
name	typedFeature			required

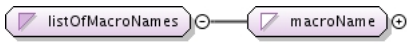
	QName	Type	Fixed	Default	Use
	shared	xsd:boolean			optional
Source	<xsd:element name="nomvar" type="shareableNamedTerm" substitutionGroup="nominal"> <xsd:annotation> <xsd:documentation>A nominal variable.</xsd:documentation> </xsd:annotation> </xsd:element>				
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd				

Simple Types

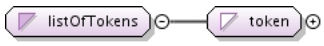
Simple Type `token`

Namespace	No namespace
Annotations	A 'token' as defined here is a string with no white space, for easy parsing, but otherwise more flexible than a NMTOKEN.
Diagram	
Type	restriction of xsd:string
Facets	pattern \S+
Used by	Attributes entry/@class, entry/@pos, entry/@stem, entry/@word, feat/@val, fs/@val, prop/@name
Source	<xsd:simpleType name="token"> <xsd:annotation> <xsd:documentation>A 'token' as defined here is a string with no white space, for easy parsing, but otherwise more flexible than a NMTOKEN.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value="\S+"/> </xsd:restriction> </xsd:simpleType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/tokens.xsd

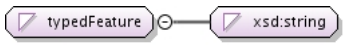
Simple Type `listOfMacroNames`

Namespace	No namespace
Diagram	
Type	list of macroName
Used by	Attribute entry/@macros
Source	<xsd:simpleType name="listOfMacroNames"> <xsd:list itemType="macroName"/> </xsd:simpleType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Simple Type `listOfTokens`

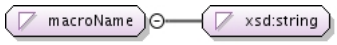
Namespace	No namespace
Diagram	
Type	list of token
Used by	Attribute entry/@excluded
Source	<xsd:simpleType name="listOfTokens"> <xsd:list itemType="token"/> </xsd:simpleType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/tokens.xsd

Simple Type `typedFeature`

Namespace	No namespace
Annotations	Feature variables with optional type.
Diagram	

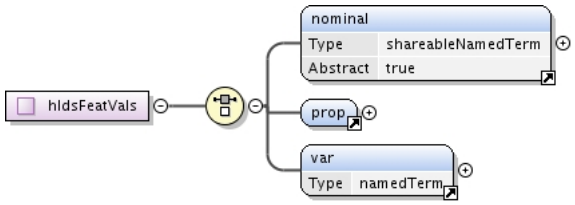
Type	restriction of xsd:string
Facets	pattern \S+(:\S+)?
Used by	Attributes featvar/@name, namedTerm/@name, satop/@nom, satop/@nomvar
Source	<xsd:simpleType name="typedFeature"> <xsd:annotation> <xsd:documentation>Feature variables with optional type.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value="\S+(:\S+)?"/> </xsd:restriction> </xsd:simpleType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/tokens.xsd

Simple Type `macroName`

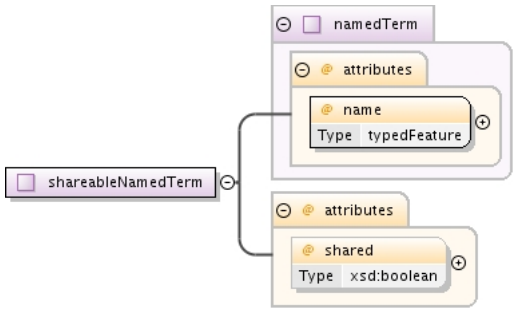
Namespace	No namespace
Annotations	Macro names begin with @ (for historical reasons) and are followed by at least one non-whitespace character.
Diagram	
Type	restriction of xsd:string
Facets	pattern @\S+
Used by	Attribute macro/@name
Source	<xsd:simpleType name="macroName"> <xsd:annotation> <xsd:documentation>Macro names begin with @ (for historical reasons) and are followed by at least one non-whitespace character.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value="@\S+"/> </xsd:restriction> </xsd:simpleType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Complex Types

Complex Type `hldsFeatVals`

Namespace	No namespace
Annotations	The possible values of an LF feature, i.e., a nominal (atom or variable), proposition, or variable (over HLDS terms).
Diagram	
Used by	Element feat/lf
Model	nominal | prop | var
Children	nominal, prop, var
Source	<xsd:complexType name="hldsFeatVals"> <xsd:annotation> <xsd:documentation>The possible values of an LF feature, i.e., a nominal (atom or variable), proposition, or variable (over HLDS terms).</xsd:documentation> </xsd:annotation> <xsd:choice> <xsd:element ref="nominal"/> <xsd:element ref="prop"/> <xsd:element ref="var"/> </xsd:choice> </xsd:complexType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Complex Type `shareableNamedTerm`

Namespace	No namespace
Annotations	A named term with an optional "shared" attribute.
Diagram	
Type	extension of namedTerm
Type hierarchy	• namedTerm • shareableNamedTerm
Used by	Elements nom, nominal, nomvar

Attributes

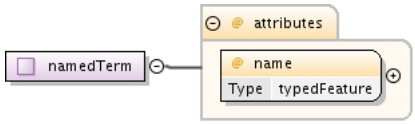
QName	Type	Fixed	Default	Use
name	typedFeature			required
shared	xsd:boolean			optional

Source

```
<xsd:complexType name="shareableNamedTerm">
  <xsd:annotation>
    <xsd:documentation>A named term with an optional "shared" attribute.</
xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="namedTerm">
      <xsd:attribute name="shared" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Complex Type `namedTerm`

Namespace	No namespace
Annotations	A term with a name.
Diagram	

Used by

Complex Type	shareableNamedTerm
Element	var

Attributes

QName	Type	Fixed	Default	Use
name	typedFeature			required

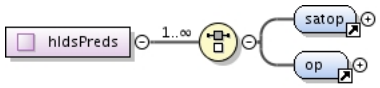
Source

```
<xsd:complexType name="namedTerm">
  <xsd:annotation>
    <xsd:documentation>A term with a name.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="name" type="typedFeature" use="required"/>
</xsd:complexType>
```

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Complex Type `hldsPreds`

Namespace	No namespace
Annotations	Typically one or more satisfaction operators, encoding the semantics associated with a category.

	Exclusive disjunctions (xor) are also allowed.
Diagram	
Used by	Elements atomcat/lf, complexcat/lf, macro/lf
Model	satop | op
Children	op, satop
Source	<xsd:complexType name="hldsPreds"> <xsd:annotation> <xsd:documentation>Typically one or more satisfaction operators, encoding the semantics associated with a category. Exclusive disjunctions (xor) are also allowed.</xsd:documentation> </xsd:annotation> <xsd:choice maxOccurs="unbounded"> <xsd:element ref="satop"/> <xsd:element ref="op"/> </xsd:choice> </xsd:complexType>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attributes

Attribute entry/@word

Namespace	No namespace
Type	token
Properties	use: required
Facets	pattern \S+
Used by	Element entry
Source	<xsd:attribute name="word" type="token" use="required"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@stem

Namespace	No namespace
Type	token
Properties	content: simple
Facets	pattern \S+
Used by	Element entry
Source	<xsd:attribute name="stem" type="token"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@pos

Namespace	No namespace
Type	token
Properties	use: required
Facets	pattern \S+
Used by	Element entry
Source	<xsd:attribute name="pos" type="token" use="required"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@class

Namespace	No namespace

Type	token
Properties	content: simple
Facets	pattern \S+
Used by	Element entry
Source	<xsd:attribute name="class" type="token"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@macros

Namespace	No namespace
Type	listOfMacroNames
Properties	content: simple
Used by	Element entry
Source	<xsd:attribute name="macros" type="listOfMacroNames"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@excluded

Namespace	No namespace
Type	listOfTokens
Properties	content: simple
Used by	Element entry
Source	<xsd:attribute name="excluded" type="listOfTokens"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute entry/@coart

Namespace	No namespace
Type	xsd:boolean
Properties	content: simple
Used by	Element entry
Source	<xsd:attribute name="coart" type="xsd:boolean"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd

Attribute featvar/@name

Namespace	No namespace
Annotations	The name of the feature variable.
Type	typedFeature
Properties	use: required
Facets	pattern \S+(:\S+)?
Used by	Element featvar
Source	<xsd:attribute name="name" type="typedFeature" use="required"> <xsd:annotation> <xsd:documentation>The name of the feature variable.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute namedTerm/@name

Namespace	No namespace

Type	typedFeature
Properties	use: required
Facets	pattern \S+(:\S+)?
Used by	Complex Type namedTerm
Source	<xsd:attribute name="name" type="typedFeature" use="required"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute shareableNamedTerm/@shared

Namespace	No namespace
Type	xsd:boolean
Properties	content: simple
Used by	Complex Type shareableNamedTerm
Source	<xsd:attribute name="shared" type="xsd:boolean"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute prop/@name

Namespace	No namespace
Type	token
Properties	use: required
Facets	pattern \S+
Used by	Element prop
Source	<xsd:attribute name="name" type="token" use="required"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute feat/@attr

Namespace	No namespace
Annotations	The name of the feature.
Type	xsd:NMTOKEN
Properties	use: required
Used by	Element feat
Source	<xsd:attribute name="attr" type="xsd:NMTOKEN" use="required"> <xsd:annotation> <xsd:documentation>The name of the feature.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute feat/@val

Namespace	No namespace
Annotations	The value of the feature, when string-valued.
Type	token
Properties	content: simple
Facets	pattern \S+
Used by	Element feat
Source	<xsd:attribute name="val" type="token"> <xsd:annotation> <xsd:documentation>The value of the feature, when string-valued.</xsd:documentation> </xsd:annotation> </xsd:attribute>

Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `fs/@id`

Namespace	No namespace
Annotations	An integer id for the feature structure. The id is used for coindexation, "inheritsFrom" feature propagation, and macro access.
Type	xsd:integer
Properties	content: simple
Used by	Element fs
Source	<xsd:attribute name="id" type="xsd:integer"> <xsd:annotation> <xsd:documentation>An integer id for the feature structure. The id is used for coindexation, "inheritsFrom" feature propagation, and macro access.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `fs/@inheritsFrom`

Namespace	No namespace
Annotations	This attribute is used to specify feature propagation with exceptions (i.e., default unification). At run time, feature equations are added in order to propagate feature values. In particular, a feature variable is added to this feature structure and a corresponding one is added to the referenced feature structure, for all appropriate attributes except those with explicit values already on this feature structure. The appropriate attributes are determined by the type of the atomic category, and consist of all the attributes seen with this category type in the lexicon file. NB: Attributes appearing in macros in the morph file are not included, since their association with category types is not available statically.
Type	xsd:integer
Properties	content: simple
Used by	Element fs
Source	<xsd:attribute name="inheritsFrom" type="xsd:integer"> <xsd:annotation> <xsd:documentation>This attribute is used to specify feature propagation with exceptions (i.e., default unification). At run time, feature equations are added in order to propagate feature values. In particular, a feature variable is added to this feature structure and a corresponding one is added to the referenced feature structure, for all appropriate attributes except those with explicit values already on this feature structure. The appropriate attributes are determined by the type of the atomic category, and consist of all the attributes seen with this category type in the lexicon file. NB: Attributes appearing in macros in the morph file are not included, since their association with category types is not available statically.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `fs/@attr`

Namespace	No namespace
Annotations	The name of the single, string-valued feature (when appropriate).
Type	xsd:NMTOKEN
Properties	content: simple
Used by	Element fs
Source	<xsd:attribute name="attr" type="xsd:NMTOKEN"> <xsd:annotation> <xsd:documentation>The name of the single, string-valued feature (when appropriate).</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `fs/@val`

Namespace	No namespace
Annotations	`The value of the feature.`
Type	token
Properties	content: `simple`
Facets	pattern `\S+`
Used by	Element fs
Source	<xsd:attribute name="val" type="token"> <xsd:annotation> <xsd:documentation>The value of the feature.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `op/@name`

Namespace	No namespace
Type	restriction of xsd:string
Properties	use: `required`
Facets	enumeration `conj, xor, opt`
Used by	Element op
Source	<xsd:attribute name="name" use="required"> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="conj"/> <xsd:enumeration value="xor"/> <xsd:enumeration value="opt"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute `diamond/@mode`

Namespace	No namespace
Type	xsd:NMTOKEN
Properties	use: `required`
Used by	Element diamond
Source	<xsd:attribute name="mode" type="xsd:NMTOKEN" use="required"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute `satop/@nom`

Namespace	No namespace
Type	typedFeature
Properties	content: `simple`
Facets	pattern `\S+(:\S+)?`
Used by	Element satop
Source	<xsd:attribute name="nom" type="typedFeature"/>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd

Attribute `satop/@nomvar`

Namespace	No namespace
Type	typedFeature

Properties	content:	simple
Facets	pattern	\S+(:\S+)?
Used by	Element	satop
Source	<xsd:attribute name="nomvar" type="typedFeature"/>	
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd	

Attribute satop/@shared

Namespace	No namespace	
Type	xsd:boolean	
Properties	content:	simple
Used by	Element	satop
Source	<xsd:attribute name="shared" type="xsd:boolean"/>	
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/hlds.xsd	

Attribute macro/@name

Namespace	No namespace	
Type	macroName	
Properties	use:	required
Facets	pattern	@\S+
Used by	Element	macro
Source	<xsd:attribute name="name" type="macroName" use="required"/>	
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd	

Attribute morph/@name

Namespace	No namespace	
Type	xsd:string	
Properties	content:	simple
Used by	Element	morph
Source	<xsd:attribute name="name" type="xsd:string"/>	
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/morph.xsd	

Attribute atomcat/@type

Namespace	No namespace
Annotations	The type of the category, e.g. "np".
Type	xsd:NMTOKEN
Properties	use: required
Used by	Element atomcat
Source	<xsd:attribute name="type" type="xsd:NMTOKEN" use="required"> <xsd:annotation> <xsd:documentation>The type of the category, e.g. "np".</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute slash/@dir

Namespace	No namespace
Annotations	The direction of the slash. The direction can be forward (/), backward (\) or both (|).

	Defaults to both.
Type	restriction of xsd:string
Properties	content: simple
Facets	enumeration /, \, |
Used by	Element slash
Source	<xsd:attribute name="dir"> <xsd:annotation> <xsd:documentation>The direction of the slash. The direction can be forward (/), backward (\) or both (|). Defaults to both.</xsd:documentation> </xsd:annotation> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="/"/> <xsd:enumeration value="\"/> <xsd:enumeration value="|"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute slash/@mode

Namespace	No namespace
Annotations	The mode of the slash. The possible values are: all (.), application only (*), associative (^), permutative (x), permutative right (x>), permutative left (<x), associative permutative right (>), and associative permutative left (<). Defaults to all. See Jason's dissertation for details.
Type	restriction of xsd:string
Properties	content: simple
Facets	enumeration ., *, ^, x, x>, <x, >, <
Used by	Element slash
Source	<xsd:attribute name="mode"> <xsd:annotation> <xsd:documentation>The mode of the slash. The possible values are: all (.), application only (*), associative (^), permutative (x), permutative right (x>), permutative left (<x), associative permutative right (>), and associative permutative left (<). Defaults to all. See Jason's dissertation for details.</xsd:documentation> </xsd:annotation> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="."/> <xsd:enumeration value="*"/> <xsd:enumeration value="^"/> <xsd:enumeration value="x"/> <xsd:enumeration value="x>"/> <xsd:enumeration value="<x"/> <xsd:enumeration value=">"/> <xsd:enumeration value="<"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute slash/@varmodality

Namespace	No namespace
Annotations	A variable over modalities.
Type	xsd:NMTOKEN
Properties	content: simple
Used by	Element slash
Source	<xsd:attribute name="varmodality" type="xsd:NMTOKEN">

	<xsd:annotation> <xsd:documentation>A variable over modalities.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `slash/@ability`

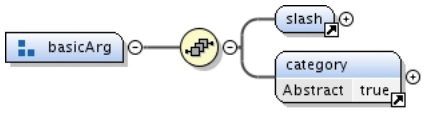
Namespace	No namespace
Annotations	The ability of the slash, either inert or active. Defaults to either. This is used to implement antecedent government; see Jason's dissertation for details.
Type	restriction of xsd:string
Properties	content: simple
Facets	enumeration inert, active
Used by	Element slash
Source	<xsd:attribute name="ability"> <xsd:annotation> <xsd:documentation>The ability of the slash, either inert or active. Defaults to either. This is used to implement antecedent government; see Jason's dissertation for details.</xsd:documentation> </xsd:annotation> <xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:enumeration value="inert"/> <xsd:enumeration value="active"/> </xsd:restriction> </xsd:simpleType> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Attribute `dollar/@name`

Namespace	No namespace
Annotations	The name of the dollar variable, for coindexation purposes.
Type	xsd:NMTOKEN
Properties	content: simple
Used by	Element dollar
Source	<xsd:attribute name="name" type="xsd:NMTOKEN"> <xsd:annotation> <xsd:documentation>The name of the dollar variable, for coindexation purposes.</xsd:documentation> </xsd:annotation> </xsd:attribute>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element Groups

Element Group `basicArg`

Namespace	No namespace
Diagram	
Used by	Element setarg
Model	slash , category
Children	category, slash
Source	<xsd:group name="basicArg"> <xsd:annotation> <xsd:documentation>A basic arg pairs a slash and a category.</xsd:documentation> </xsd:annotation>

	<xsd:sequence> <xsd:element ref="slash"/> <xsd:element ref="category"/> </xsd:sequence> </xsd:group>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd

Element Group `dollarArg`

Namespace	No namespace
Diagram	
Model	slash{0,1} , dollar
Children	dollar, slash
Source	<xsd:group name="dollarArg"> <xsd:annotation> <xsd:documentation>A dollar arg pairs an optional slash and a dollar variable. The slash defaults to the most general slash.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element ref="slash" minOccurs="0"/> <xsd:element ref="dollar"/> </xsd:sequence> </xsd:group>
Schema location	file:/zusatz/Magister/openCCG/openccg/grammars/categories.xsd