**Participants:**

Bettina Klimek (BK)

Jakub Šimek

Stefania Racioppa (DFKI)

Matteo Pellegrini

Fahad Khan

Katerina Gkirtzou

1. **Module draft 4.3**



Adaptions of module draft 4.3 to be included for next telco:

* turn morph:example and morph:replacement into datatype properties (rdf:string)
* remove morph:Rule superclass and interconnect morph:generates with the two subclasses as domain

**General notes:**

this is one way of distinguishing morph, morpheme, allomorph

<http://www.sfu.ca/person/dearmond/323/323%20/323.morph.htm>

* discuss: connection between morph:Morph and morph:Rule
* add as recommendation that input of morph:InflectionRule should be an ontolex:LexicalEntry with ontolex:canonicalForm

1. **Review** [**Stefania’s example**](https://docs.google.com/document/d/1stKL92xx-ONhDw9hRshIS65VHY7rkk5Hy6hhHd08BJg/edit?usp=sharing)**, proposed solution to problem nb. 2 (Max)**

MI: regarding nb.1: in my proposal there was a link from a word to the inflection type of its first affix. Was it deliberately removed?

<#kissa> a ontolex:Word ;

ontolex:canonicalForm [ontolex:writtenRep "kissa"@fi];

morph:inflectionType <#finnish\_noun\_type\_kissa> .

# this is more verbose

:lex\_abdicatio a ontolex:LexicalEntry ;

lexinfo:gender lexinfo:feminine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :la-noun\_3 ;

morph:inflectionType :la-noun\_3n-stem-var ; # this is what I added

morph:inflectionType :la-noun\_3n-no-stem-var ;

ontolex:evokes :a00031 ;

ontolex:canonicalForm [ ontolex:writtenRep "abdicatio"@la ] .

:la-noun\_3n-stem-var a morph:InflectionType ;

morph:hasParadigm :la-noun\_3 ;

morph:inflects :la-noun\_3n\_other\_stem ,

:la-noun\_3n\_form\_1 ,

[...] ;

morph:next :la-noun\_3n-infl .

:la-noun\_3n\_other\_stem a morph:InflectionRule ;

morph:generates [ :stem :oblique ] ;

morph:replacement [ morph:source "$" ;

morph:target "n" ] .

:la-noun\_3n-infl a morph:InflectionType ;

morph:hasParadigm :la-noun\_3 ;

morph:inflects :la-noun\_3n\_abl\_m-f\_pl ,

:la-noun\_3n\_abl\_m-f\_sg ,

[...] .

1. **German compound example data for evaluation (Stefania)**

Source

Frauenbewegung <6> :0: Bewegung <6> ## (1) Frauenbewegung

Frauenbewegung <6> :3: Frau <6> ## (1) Frauenbewegung

Abendgesellschaft <6> :0: Gesellschaft <6> ## (1) Abendgesellschaft

Abendgesellschaft <6> :3: Abend <7> ## (1) Abendgesellschaft

RDF output

:lex\_frauenbewegung a ontolex:LexicalEntry ;

**decomp:component :lex\_bewegung,**

**:lex\_frau ;**

lexinfo:partOfSpeech lexinfo:noun ;

ontolex:canonicalForm [ ontolex:writtenRep "Frauenbewegung"@de ] ;

ontolex:morphologicalPattern :de-noun\_008 .

:lex\_abendgesellschaft a ontolex:LexicalEntry ;

**decomp:component :lex\_abend,**

**:lex\_gesellschaft ;**

lexinfo:gender lexinfo:feminine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :de-noun\_008 ;

ontolex:canonicalForm [ ontolex:writtenRep "Abendgesellschaft"@de ] .

Components

:lex\_abend a ontolex:LexicalEntry ;

lexinfo:gender lexinfo:masculine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :de-noun\_017 ;

ontolex:canonicalForm [ ontolex:writtenRep "Abend"@de ] .

:lex\_bewegung a ontolex:LexicalEntry ;

lexinfo:gender lexinfo:feminine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :de-noun\_008 ;

ontolex:canonicalForm [ ontolex:writtenRep "Bewegung"@de ] .

:lex\_frau a ontolex:LexicalEntry ;

lexinfo:gender lexinfo:feminine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :de-noun\_008 ;

ontolex:canonicalForm [ ontolex:writtenRep "Frau"@de ] .

:lex\_gesellschaft a ontolex:LexicalEntry ;

lexinfo:gender lexinfo:feminine ;

lexinfo:partOfSpeech lexinfo:noun ;

morph:hasParadigm :de-noun\_008 ;

ontolex:canonicalForm [ ontolex:writtenRep "Gesellschaft"@de ] .

Inflections

:de-noun\_008 a morph:Paradigm ;

rdfs:comment "German noun inflection" .

:de-noun\_008-infl a morph:InflectionType ;

morph:hasParadigm :de-noun\_008 ;

morph:inflects :de-noun\_008\_acc-dat-gen-nom\_pl,

:de-noun\_008\_acc-dat-gen-nom\_sg .

:de-noun\_008\_acc-dat-gen-nom\_pl a morph:InflectionRule ;

morph:generates [ lexinfo:case lexinfo:accusative,

lexinfo:dative,

lexinfo:genitive,

lexinfo:nominative ;

lexinfo:number lexinfo:plural ] ;

morph:replacement [ morph:source "$" ;

morph:target "en" ] .

:de-noun\_008\_acc-dat-gen-nom\_sg a morph:InflectionRule ;

morph:generates [ lexinfo:case lexinfo:accusative,

lexinfo:dative,

lexinfo:genitive,

lexinfo:nominative ;

lexinfo:number lexinfo:singular ] ;

morph:replacement [ morph:source "$" ;

morph:target "" ] .

[similar for :de-noun\_017]

Fahad:

so in the case of passerby we would have a relationship between passer and by as lexical entries but then we would have a relationsjip between morph(emes) for the plural 'passersby'

\*word formation relationship

1. **Finnish example data for evaluation (Max)**

# Words

<#kissa> a ontolex:Word ;

ontolex:canonicalForm [ontolex:writtenRep "kissa"@fi];

morph:inflectionType <#finnish\_noun\_type\_kissa\_number>, <#finnish\_noun\_type\_kissa\_number\_obl> . # the first category after the stem

# Inflection types

# They are combined with the `:next` property.

# This is actually quite usual for linguists: the same approach is used in the `lexc` formalism widely used in computational morphology (incl. xfst and foma)

<#finnish\_noun\_type\_kissa\_number> a morph:InflectionType ;

morph:hasParadigm <#regular\_finnish\_noun> ;

morph:next <#finnish\_noun\_type\_kissa\_case> ;

morph:inflects <#finnish\_noun\_type\_kissa\_sg>, <#finnish\_noun\_type\_kissa\_pl> .

<#finnish\_noun\_type\_kissa\_case> a morph:InflectionType ;

morph:hasParadigm <#regular\_finnish\_noun> ;

morph:inflects <#finnish\_noun\_type\_kissa\_nom>, <#finnish\_noun\_type\_kissa\_ine> .

<#finnish\_noun\_type\_kissa\_nom> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_case> ;

morph:generates [ lexinfo:case lexinfo:nominative ] ;

morph:replacement [ morph:source "$"; morph:target "" ] .

<#finnish\_noun\_type\_kissa\_sg> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_number> ;

morph:generates [ lexinfo:number lexinfo:singular ] ;

morph:replacement [morph:source "$", morph:target ""] . # we can actually create nodes for widely used replacements, like \_no replacement\_

<#finnish\_noun\_type\_kissa\_ine> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_case> ;

morph:generates [ lexinfo:case lexinfo:inessive ]; # it's a convenient example even though the inessive of a cat is a bit weird

morph:replacement [morph:source "$", morph:target "ssa"] .

<#finnish\_noun\_type\_kissa\_pl> a morph:InflectionRule ;

morph:inflectionType <#finnish\_noun\_type\_kissa\_number> ;

morph:generates [ lexinfo:number lexinfo:plural ] ;

morph:replacement [morph:source "$", morph:target "t"] .

Todos:

* when CC is present discuss if morph:Morph and morph:Rule are different and then examine a direct interconnection between both classes by looking at Inuktitut data
* Bettina: add telco URL and meeting cycle/time to Morphology Module Wiki page