**Participants:**

Bettina

Max  
Julia

Fahad

Full samples: <https://drive.google.com/drive/folders/1aONl-KDPwMpa3DmhCKAPlbprS_ToCPE0?usp=sharing>

* Diagram for the Finnish example:

<#kissa> a ontolex:Word ;

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

morph:inflects <#finnish\_noun\_type\_kala> .

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

morph:paradigm <#finnish\_noun\_type\_9> ;

morph:next <#finnish\_noun\_type\_kala\_number\_direct> ; # this means that there will be more "slots" for grammatical categories

morph:next <#finnish\_noun\_type\_kala\_number\_oblique> .

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

morph:paradigm <#finnish\_noun\_type\_9> ;

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

morph:example "kissa"@fi .

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

morph:paradigm <#finnish\_noun\_type\_9> ;

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

morph:example "kissoi-"@fi .

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

morph:paradigm <#finnish\_noun\_type\_9> ;

morph:next <#finnish\_noun\_type\_kala\_post> ;

morph:example "kissassa"@fi .

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

morph:paradigm <#finnish\_noun\_type\_9> ;

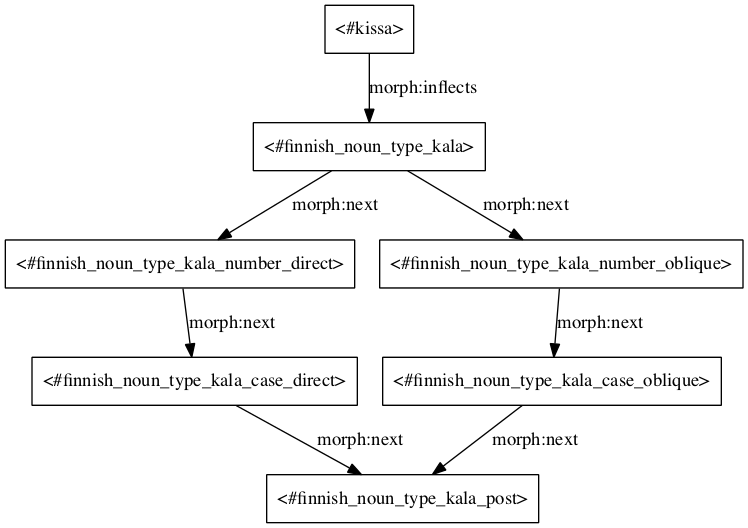
morph:next <#finnish\_noun\_type\_kala\_post> ;

morph:example "kissoissa"@fi .

<#finnish\_noun\_type\_kala\_post> a morph:InflectionType ;

morph:paradigm <#finnish\_noun\_type\_9> ;

morph:example "kissojen"@fi .



* SPARQL for forms generation:

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#>

prefix lexinfo: <http://www.lexinfo.net/ontology/2.0/lexinfo#>

prefix morph: <https://www.w3.org/community/ontolex/wiki/Morphology#>

prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select ?w\_end where {

# start with an ontolex:Word

?w a ontolex:Word ;

morph:inflects ?begin ;

ontolex:canonicalForm/ontolex:writtenRep ?w\_begin .

?begin morph:next\* ?midI .

FILTER NOT EXISTS { [] morph:next ?begin }

{

?rule\_I morph:inflectionType ?midI ;

morph:replacement ?repl\_I .

?repl\_I morph:source ?s\_I ;

morph:target ?t\_I .

}

BIND(REPLACE(?w\_begin, ?s\_I, ?t\_I) as ?w\_midI)

# go to next slot

?midI morph:next ?midJ .

{

?rule\_J morph:inflectionType ?midJ ;

morph:replacement ?repl\_J .

?repl\_J morph:source ?s\_J ;

morph:target ?t\_J .

}

BIND(REPLACE(?w\_midI, ?s\_J, ?t\_J) as ?w\_midJ)

# get to the end of the path

?midJ morph:next\* ?end .

FILTER NOT EXISTS { ?end morph:next [] }

{

?rule\_end morph:inflectionType ?end ;

morph:replacement ?repl\_end .

?repl\_end morph:source ?s\_end ;

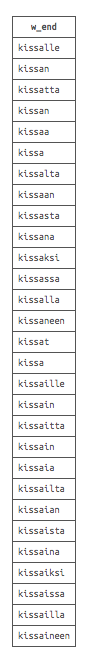
morph:target ?t\_end .

}

BIND(REPLACE(?w\_midJ, ?s\_end, ?t\_end) as ?w\_end)

}

order by ?start ?end



<#finnish\_noun\_type\_kala\_nom> a morph:Rule ;

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

lexinfo:case lexinfo:nominative ;

morph:example "kissa"@fi ; # optional

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

<#finnish\_noun\_type\_kala\_gen> a morph:Rule ;

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

lexinfo:case lexinfo:nominative ;

morph:example "kissa"@fi ; # optional

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

* Talking points:
  + There should be a connection to Form from somewhere
  + How to deal with defective paradigmata?
  + How to get linguistic information in SPARQL? Something like SELECT lexinfo:\*

Morphology Module Draft 3.0

InflectionType denotes a step that adds a certain grammatical meaning for the group of words which add the same affixes. It is a declension type without allomorphy.

Bettina: create new diagram which illustrates the generation process as a whole (input - transformation - output)

Max: create SPARQL construct query that outputs the following kinds of triples:

1. [ ] a ontolex:Form
2. [ ] a morph:Morph
3. [ontolex:Form instance] morph:consistsOf [morph:Morph instance]
4. [morph:Morph instance] morph:grammaticalMeaning [lexinfo instance]