Morph telco 2022-05-18, 13:00 CET

**Link:** [**https://meet.google.com/nsj-tbcy-yop**](https://meet.google.com/nsj-tbcy-yop) **[check here for link updates if it doesn’t work]**

**Latest Definitions:**  <https://github.com/ontolex/morph/blob/master/draft.md>

**Latest Paper (submitted to LDL-2022):** <https://www.overleaf.com/4868363189kczjzdndgxwc> (folder submission/)

**Participants [please add yourself]:**

Christian Chiarcos (CC) [christian.chiarcos@gmail.com](mailto:christian.chiarcos@gmail.com)

Max Ionov (MI)

Katerina Gkirtzou (KG)

Fahad Khan (FK)

Matteo Pellegrini (MP)

Ciprian-Octavian Truică (CT)

Penny Labropoulou (PL)

Elena Simona Apostol (ESA)

Agenda (please add, *but* do not edit table of contents directly, but add sections below and then update here):

[**0. Module draft**](#_1kaaos1w4ub1) **2**

[**1. Publications**](#_5q4h03dxajtd) **2**

[2. definition consolidation](#_n0e2ll1nl5iu) **3**

[**2.1 InflectionType**](#_letz9qce1sim) **4**

[**2.2 morph(eme) order**](#_ki1gty3ur827) **6**

[**3 open problems/other data**](#_jwteu367w2rp) **6**

[3.0 Comparison with MMoOn (Mod. Greek, Hebrew, other; unassigned)](#_uwa8n37v0ip) 6

[3.1 Samples to be modelled (all)](#_s3vk3hv6pguv) 6

[3.2 inflection tables (Fahad, others?)](#_rzzsrcthlbm3) 7

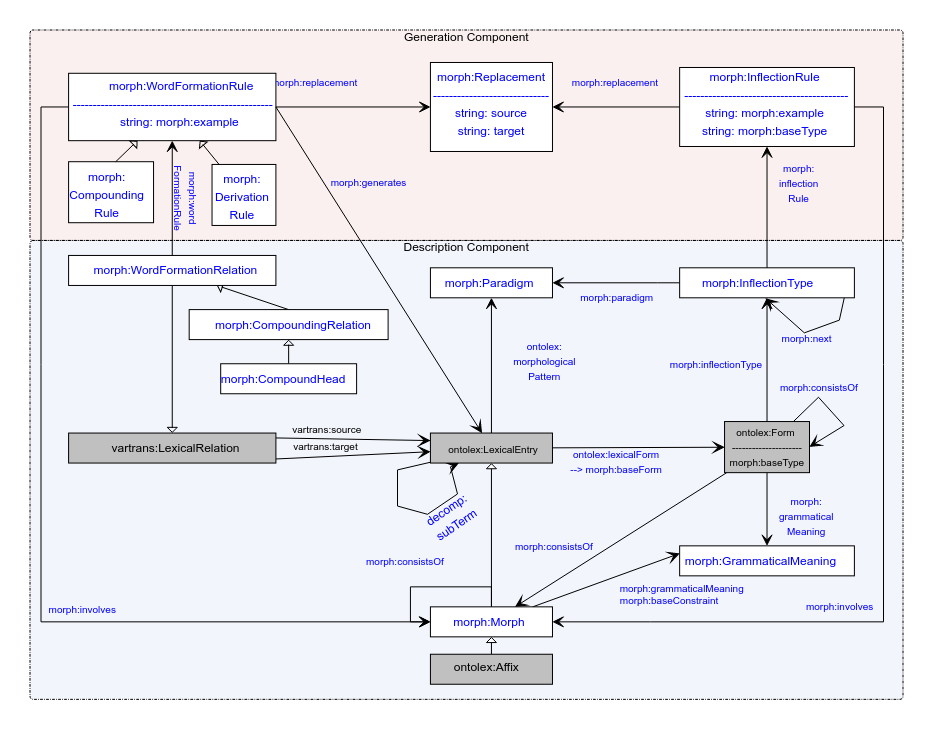
[3.3 semitic consonantal roots (unassigned)](#_w3ikvzzgmbyd) 7

[**4. AOB**](#_4u89668ejc7q) **8**

# 

# 0. Module draft

**draft 4.14 (no changes)**



**Model draft 4.15 updates (to be discussed) & open issues:**

* Head/CompoundingHead/CompoundHead stays → it’s okay to change it for homogenity
  + Matteo: Head misleading, could be suffix in derivation
  + Maybe we could rename the class completely, not using “Head”. **DONE@CC**: create an issue on GH: <https://github.com/ontolex/morph/issues/10>
  + majority vote: stay with CompoundHead
  + **TODO@all**: discuss and maybe come up with the solution
* inflection type to be discussed
* consistsOf?
  + tentative consensus:
    - drop consistsOf between morph and morph
    - -”- between form and form
    - keep between form and morph
    - add aggegator relation between form and morph (=> rdfs:\_1 etc. to encode position); => redefine ontolex:Form as a Container
    - **TODO@all**: think about whether you ever used one of the dropped consistsOf properties

# 1. Publications

* LDL
  + notification => approved, reviewer comments in overleaf
  + deadline: May 23 => **TODO**: coordinate via Slack
* general OntoLex overview
  + ?ESWC: Deadline?
    - ESWC: 2 Dec 2021 for ESWC2022 -> there are no dates for the ESWC2023 CFP as the ESWC2022 is between 29th May and 2nd June
  + update of OntoLex, incl. FrAC, Morph, MModality
* paper on word formation?
  + idea for novel paper: word formation in OntoLex-Lemon
    - not original content, but more like a survey and documentation of best practices?
    - can be helpful to consolidate/revise word formation part of the module
    - possible input from LiLa
    - **TODO@all:** think about possible venues
* LLODREAM? <https://easychair.org/cfp/llodream2022>
  + OntoLex-Morph status update
  + 500 words
  + deadline june 15th
  + ?publication
    - postproceedings (lith. journal or university internal)
  + contributors (order tbd)
    - Christian
    - Max
    - Fahad
    - Matteo
    - Marco
    - Penny
    - Ciprian
    - Katerina
    - Elena
  + topic: motivation / general overview
    - sth more specific?
* later journal paper
  + After the final publication
  + Or: an overview of the current state. Frac + Morph or Frac separately, Morph separately?
  + (at some point) a book?

# 2. definition consolidation

* internal deadline: **TODAY** !
  + pull requests by Matteo and Penny
  + **TODO@CC,MI**: review and merge within the next 10 days
* <https://github.com/ontolex/morph/blob/master/draft.md>
* you can contribute suggestions by creating issues (<https://github.com/ontolex/morph/issues>), via pull requests, or by direct editing (share your GitHub username)
* definition refinement
  + procedure:
    - open an issue
    - pull request + close the issue
  + Penny: technical or linguistic definitions?
    - technical definitions, but linguistic explanation (“definition”) in text
* open issues (postponed)
  + **OPEN**: define cardinality restrictions: <https://github.com/ontolex/morph/issues/12>
    - suggestion: when finalizing the vocabulary
  + **CHECK STATUS**: define morph subclasses in LexInfo rather than OntoLex-Morph, also add equivalence axioms (lexinfo:Prefix subclassOf [ lexinfo:termElement lexinfo:prefix ])
    - <https://github.com/ontolex/lexinfo/pull/29>
    - not merged yet
  + **unassigned**: describe the relation between decomp and CompoundRelation
    - **suggestion**: do this as part of writing a designated paper [venue?]
    - **TODO**@unassigned: document relation between both modules in appendix
      * there is an alternative reification with decomp:Component, but this is less well-suited for compound analysis, because it doesn’t relate to lexicosemantic relations.
      * the current modelling of decomp is oriented towards an analysis of synsem (semantic) roles within a compound. in morphology, we normally don’t have that, what we have, instead, are relations between lexemes and morphemes.
  + describe grouping of lexical (sub-) entries
    - LiLa: “flexeme”, sub-entries with different paradigms, but identical in meaning, etc.
      * + suggestion: model the grouping by lexicog, have both the overarching lexical entry and the flexemes as separate lexical entries, no vocabulary extension needed, but a usage note in the report
        + **tbc:** by LiLa
    - Penny: sub-entries of the same lexical entry to mark contracted and non-contracted versions of the same paradigm
      * + can be partially modelled by means of “markers”, i.e., lexinfo usage properties, instead
        + **todo@Penny**: tbc. whether lexinfo needs to be extended for that

domain: LexicalSense

**TODO**: ask John

if these properties are added, no sub-groups necessary

* + @all: think about metadata properties for LexInfo (hypothetical/unattested form, etc.) => tentative consensus, but details to be discussed
    - Penny: could work, but domain is ontolex:LexicalSense. Can this be changed?
    - **TODO**: ask John

## 2.1 InflectionType

* current definitions:
  + Class **morph:InflectionType** represents a single slot for a single grammatical category for all its possible values (e.g. all the cases)
    - Book analogy: a column from a paradigm table without allomorphy/alternative variants for just a single morpheme
  + property **morph:inflectionType** assigns an inflectional pattern of a form as belonging to a morphological pattern of a lexical entry
* CC (offline): this definition *does not work* for the current diagram, if one inflection type represents the position for \*all\* cases, we cannot associate the form for, say, dative with the rule for dative via inflection type (thanks to Matteo for pointing that out).
  + <https://github.com/ontolex/morph/issues/11>
* Comparing alternatives:
  + **TODO**@Max: graphics [**postponed**]
  + **TODO**@Max: example agglutinative [**postponed**]
  + **TODO**@Katerina+Penny: example fusional

### current model

* + - 1. Form -inflectionType-> InflectionType
      2. Paradigm <-paradigm- InflectionType
      3. InflectionType -inflectionRule-> InflectionRule
      4. InflectionType -next-> InflectionType
    - alternative 0: keep current model, one inflection type per paradigm and rule

**pro**: backward-compatible

**con**: unneccessarily verbose: what is the difference to inflection rule then?

**con**: still contradicts current definition

#### Example for fusional language:

**Notes**:

1. we have assumed that grammatical meaning (single feature or bundle of features) is a property on inflection rule based on previous discussions and example at <https://docs.google.com/document/d/1iCv865GtEksO_wd0WC7bfU-at1dEKOOea9HSlFcTTkA/edit>

2. Question for cardinality of inflection type: can it take multiple inflectional paradigms? This is useful in case the same combination of stem and ending with the same grammatical features is used in other inflectional paradigms: for instance, for Greek the inflectional paradigm "vrahos" is similar to "efyvos" but with only one stem; so, the form for nominative singular is derived from exactly the same combination (stem 1 & ending "ος") for both paradigms.

### lexical entry with stems and inflectional paradigm ##

<anthropos> a ontolex:Word ;

lexinfo:partOfSpeech lexinfo:noun ;

ontolex:morphologicalPattern <efyvos\_paradigm> ;

morph:baseForm <anthrop1\_form>;

morph:baseForm <anthrop2\_form>;

rdfs:label "άνθρωπος"@el ;

##### Generated by applying the inflection rules ####

ontolex:canonicalForm <anthropos\_form> ;

ontolex:otherForm <anthropou1\_form> ;

ontolex:otherForm <anthropou2\_form> .

########### Generated forms ###################

<anthropos\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπος"@el ;

morph:inflectionType <efyvos\_inflection\_type\_masgno> ;

morph:grammaticalMeaning <MaSgNo> .

<anthropou1\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπου"@el ;

morph:inflectionType <efyvos\_inflection\_type\_masgge1> ;

morph:grammaticalMeaning <MaSgGe> .

<anthropou2\_form> a ontolex:Form ;

ontolex:writtenRep "ανθρώπου"@el ;

morph:inflectionType <efyvos\_inflection\_type\_masgge2> ;

morph:grammaticalMeaning <MaSgGe>

##usage features to be added here##.

############ Stems #############

<anthrop1\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπ"@el ;

morph:baseType "1" .

<anthrop2\_form> a ontolex:Form ;

ontolex:writtenRep "ανθρώπ"@el ;

morph:baseType "2" .

##### Inflectional paradigm, inflection types, inflection rules and replacements ########

<efyvos\_paradigm> a Paradigm ;

rdfs:label "Inflectional paradigm of nouns like έφηβος" .

<efyvos\_inflection\_type\_masgno> a morph:InflectionType ;

morph:paradigm <efyvos\_paradigm> ;

morph:inflectionRule <inflRule\_MaSgNo> .

<efyvos\_inflection\_type\_masgge1> a morph:InflectionType ;

morph:paradigm <efyvos\_paradigm> ;

morph:inflectionRule <inflRule\_MaSgGe1> .

<efyvos\_inflection\_type\_masgge2> a morph:InflectionType ;

morph:paradigm <efyvos\_paradigm> ;

morph:inflectionRule <inflRule\_MaSgGe2> .

<inflRule\_MaSgNo> a morph:InflectionRule ;

morph:baseType "1" ;

morph:replacement <repl\_os> ;

morph:grammaticalMeaning <MaSgNo> .

<inflRule\_MaSgGe1> a morph:InflectionRule ;

morph:paradigm <efyvos\_paradigm> ;

morph:baseType "1" ;

morph:replacement <repl\_ou> ;

morph:grammaticalMeaning <MaSgGe> .

<inflRule\_MaSgGe2> a morph:InflectionRule ;

morph:paradigm <efyvos\_paradigm> ;

morph:baseType "2" ;

morph:replacement <repl\_ou>

morph:grammaticalMeaning <MaSgGe>

# usage features for variants at the same node with grammatical features;

# values to be added with lexinfo properties and instances .

<repl\_os> a morph:Replacement ;

morph:source "$" ;

morph:target "$ος"@el .

<repl\_ou> a morph:Replacement ;

morph:source "$" ;

morph:target "$ου"@el .

<MaSgNo> a morph:GrammaticalMeaning ;

rdfs:label "MaSgNo" ;

lexinfo:gender lexinfo:masculine ;

lexinfo:number lexinfo:singular ;

lexinfo:case lexinfo:nominative .

<MaSgGe> a morph:GrammaticalMeaning ;

rdfs:label "MaSgGe" ;

lexinfo:gender lexinfo:masculine ;

lexinfo:number lexinfo:singular ;

lexinfo:case lexinfo:genitive .

### alternative 1: detach InflectionType

* + - 1. Form -inflectionRule-> InflectionRule
      2. Paradigm <-paradigm- InflectionRule
      3. InflectionRule -inflectionType-> InflectionType
      4. InflectionType -next-> InflectionType

**pro:** we basically keep all the information we have, incl. finite state modelling and agglutination

**con:** inflection type won’t be used for fusional languages and probably fall out of use

**con:** terminologically, the finite state use case is still a bit of a stretch, a better name?

**note**: paradigms should be allomorphy-free, then (this is at odds with traditional usage of “paradigm”. in inflection tables, it normally includes allomorphic variants.

#### Example for fusional language

**Notes:**

1. as pointed by Christian, for fusional languages, inflection type indeed is not used.

2. similar question on cardinality as for current model: can we have multiple paradigms on inflection rule?

### lexical entry with stems and inflectional paradigm (same as in current model) ##

<anthropos> a ontolex:Word ;

lexinfo:partOfSpeech lexinfo:noun ;

ontolex:morphologicalPattern <efyvos\_paradigm> ;

morph:baseForm <anthrop1\_form>;

morph:baseForm <anthrop2\_form>;

rdfs:label "άνθρωπος"@el ;

##### Generated by applying the inflection rules ####

ontolex:canonicalForm <anthropos\_form> ;

ontolex:otherForm <anthropou1\_form> ;

ontolex:otherForm <anthropou2\_form> .

########### Generated forms (different from current model) ###################

<anthropos\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπος"@el ;

morph:inflectionRule <inflRule\_MaSgNo> ;

morph:grammaticalMeaning <MaSgNo> .

<anthropou1\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπου"@el ;

morph:inflectionRule <inflRule\_MaSgGe1> ;

morph:grammaticalMeaning <MaSgGe> .

<anthropou2\_form> a ontolex:Form ;

ontolex:writtenRep "ανθρώπου"@el ;

morph:inflectionRule <inflRule\_MaSgGe2> ;

morph:grammaticalMeaning <MaSgGe>

##usage features to be added here##.

############ Stems (same as in current model) #############

<anthrop1\_form> a ontolex:Form ;

ontolex:writtenRep "άνθρωπ"@el ;

morph:baseType "1" .

<anthrop2\_form> a ontolex:Form ;

ontolex:writtenRep "ανθρώπ"@el ;

morph:baseType "2" .

##### Inflectional paradigm, inflection rules and replacements ########

<efyvos\_paradigm> a Paradigm ;

rdfs:label "Inflectional paradigm of nouns like έφηβος" .

<inflRule\_MaSgNo> a morph:InflectionRule ;

morph:baseType "1" ;

morph:replacement <repl\_os> ;

# From minutes of 01/12/2021 :: In workflow say “copy lexinfo properties from Rule”

morph:grammaticalMeaning <MaSgNo> .

<inflRule\_MaSgGe1> a morph:InflectionRule ;

morph:paradigm <efyvos\_paradigm> ;

morph:baseType "1" ;

morph:replacement <repl\_ou> ;

morph:grammaticalMeaning <MaSgGe> .

<inflRule\_MaSgGe2> a morph:InflectionRule ;

morph:paradigm <efyvos\_paradigm> ;

morph:baseType "2" ;

morph:replacement <repl\_ou>

morph:grammaticalMeaning <MaSgGe>

# usage features for variants at the same node with grammatical features;

# values to be added with lexinfo properties and instances .

<repl\_os> a morph:Replacement ;

morph:source "$" ;

morph:target "$ος"@el .

<repl\_ou> a morph:Replacement ;

morph:source "$" ;

morph:target "$ου"@el .

<MaSgNo> a morph:GrammaticalMeaning ;

rdfs:label "MaSgNo" ;

lexinfo:gender lexinfo:masculine ;

lexinfo:number lexinfo:singular ;

lexinfo:case lexinfo:nominative .

<MaSgGe> a morph:GrammaticalMeaning ;

rdfs:label "MaSgGe" ;

lexinfo:gender lexinfo:masculine ;

lexinfo:number lexinfo:singular ;

lexinfo:case lexinfo:genitive .

### alternative 2:

replace InflectionType by GrammaticalMeaning

* 1. Form -inflectionRule-> InflectionRule
  2. Paradigm <-paradigm- InflectionRule
  3. InflectionRule -grammaticalMeaning-> GrammaticalMeaning
  4. GrammaticalMeaning -next-> GrammaticalMeaning

**pro**: we basically keep all the information we have, incl. finite state modelling and agglutination

**pro:** we eliminate one class and we address a feature request by Penny

**pro:** slot information *can* be plausibly a part of grammatical meaning (or, better, structure)

**con**: no explicit data structures for slots, researchers would need to “discover” that from comments => rename next to nextSlot?

**con**: for FST, this is very opaque, a better name? => we could introduce a designated subclass “FiniteState” of GrammaticalMeaning !?

#### Example for fusional language

**Notes:** for fusional languages, this is the same as for alternative 1, since the inflection type is not used.

### alternative 3: merge InflectionType with InflectionRule

* 1. Form -inflectionRule-> InflectionRule
  2. Paradigm <-paradigm- InflectionRule
  3. InflectionRule -grammaticalMeaning-> GrammaticalMeaning
  4. InflectionRule -next-> InflectionRule

**pro**: we keep all the information we have, incl. finite state modelling and agglutination

**pro**: we eliminate one class and address a feature request

**pro**: “rule” is more relatable to what a finite state does than “inflection type” (which sounds static)

**con**: no explicit data structures for slots, researchers would need to “discover” that from comments

**con**: in agglutinating languages, the sequence is not over replacement rules, but classes of morphemes, so we lack a formal data structure for slots

**con**: for FST, this conflates states and replacements, normally one state can have different replacements (“rules”)

#### Example for fusional language

**Notes:** for fusional languages, this is the same as for alternative 1, since the inflection type is not used.

CC (before public discussion): my favorite would be alternative 2, with the following modifications:

* rename GrammaticalMeaning to FeatureBundle (a “slot” is described as a bundle of features, so that makes sense, and finite states are informally associated with some kind of function, but typically not a specific grammatical meaning, esp. for morphophonological processes)
* introduce a subclass FiniteState of FeatureBundle (we would informally capture the finite state itself as a feature, and the bundle would consist of exactly one such feature)

Penny+Katerina:

* alternatives 1-3 are equivalent (because of the direct link with grammatical meaning => alternative 2)
* all preferred over current model in terms of verbosity

tbc: differences on agglutinative languages

**TODO@Max**: next meeting

## 2.2 morph(eme) order

* CC: this is an old problem we always postponed, forgot to mention last time
* consistsOf
  + currently
    - ambiguous: form -> form, form -> morph, morph -> morph
    - doesn’t represent order
  + suggestion:
    - replace by aggregation (rdfs:Seg or rdfs:Bag)
    - define a form as an aggregate of morphs
    - no such relation between form -> form and morph -> morph
    - instead of morph -> morph: because morph is a lexical entry, we go from morph -> form, and then use “consistsOf” equivalent at form level
    - instead of form -> form: use form -> morph and assign the “sub-forms” as lexicalForm to the morph
  + has the original “consistsOf” ever been used by anyone before?
* tentative consensus:
  + drop consistsof between forms and morphs, keep between form and morph
  + plus: define ontolex:Form as container (seq) of morphs
  + NB: we need consistsOf to mark *morphological* segments in case somebody else defiunes a form as an aggegrate of sth else

## 2.3 replacement

(If we have misunderstood, and the following are already covered, feel free to remove.)

Taken from the Finnish example <https://docs.google.com/document/d/1iCv865GtEksO_wd0WC7bfU-at1dEKOOea9HSlFcTTkA/edit>: morph:replacement [morph:source "$", morph:target "ssa"]

This looks more like a concatenation of source and target string, rather than a replacement.

We suggest that the pattern in the source string is repeated at the target string,

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

This will allow for supporting (1) the addition of a string *before* and *after* the source pattern at the same time and (2) replacement of the source string with a totally new string.

**Case 1:** The motivation for this comes from "augmentation" in modern Greek, which is an extra character added at the beginning of verbs in past tenses (cf. <http://www.xanthi.ilsp.gr/filog/ch6/gram/vpc3.asp>) together with a suffix in the generated form. Although in the LEXIS lexicon, we treat past forms with augmentation as simply a different stem, similar cases may also exist in other languages. A simplified example for 1st person, past imperfective ('e-treh-a') and past perfective ('e-trex-a') forms, is:

<inflRule\_Id01PaIm> a morph:InflectionRule ;

morph:paradigm <trehw\_paradigm> ;

morph:baseType "1" ;

morph:replacement

# [ morph:source ".\*" ; morph:target "έ&α"@el ] .

[ morph:source "^(.\*)$" ; morph:target "έ\1α"@el ] .

morph:grammaticalMeaning <Id01PaIm> .

<inflRule\_Id01PaPe> a morph:InflectionRule ;

morph:paradigm <trehw\_paradigm> ;

morph:baseType "2" ;

morph:replacement

# [ morph:source ".\*" ; morph:target "έ&α"@el ] .

[ morph:source "^(.\*)$" ; morph:target "έ\1α"@el ] .

morph:grammaticalMeaning <Id01PaPe> .

**Case 2**: For specific words with irregular inflectional paradigms, e.g. the definite article "o", cf. <https://www.foundalis.com/lan/definart.htm> it looks like we have two options: add all inflected forms on the lemma, without generating them, or add an inflectional paradigm with one stem (the lemma itself) that appears as the source string replaced by all other forms as target strings. Simplified paradigm:

## stem: "ο" generates "o"

<inflRule\_MaSgNo\_defart> a morph:InflectionRule ;

morph:paradigm <defart\_paradigm> ;

morph:baseType "1" ;

morph:replacement

[ morph:source "$" ; morph:target "$"@el .

morph:grammaticalMeaning <MaSgNo> .

## stem: "ο" generates "του"

<inflRule\_MaSgGe\_defart> a morph:InflectionRule ;

morph:paradigm <defart\_paradigm> ;

morph:baseType "1" ;

morph:replacement

[ morph:source "$" ; morph:target "του"@el .

morph:grammaticalMeaning <MaSgGe> .

# 3 open problems/other data

## 3.0 Comparison with MMoOn (Mod. Greek, Hebrew, other; unassigned)

* Greek : <https://link.springer.com/chapter/10.1007/978-3-030-98876-0_34>
* Bettina’s data (link?)

## 3.1 Samples to be modelled (all)

* most sample data originally on GDrive (where is the link?)
  + now (also) on GitHub: <https://github.com/ontolex/morph/tree/master/data/gdrive>
    - CC: can we fully move there?
* samples @ GitHub
  + Latin (word formation<LiLa: tbc: is that covered?)
  + Sumerian (agglutination<CC: open requirement: slots)
  + Old High German (word formation<CC: open requirement: tree structures)
  + Italian (word formation<Stefania; tbc: is that covered?)
  + Italian (inflection<WHOM?; tbc: is that covered?)
  + Inuktitut (incorporation, polypersonal agreement, assimilation/allomorphy; generation/parsing<CC: open requirement
  + UniMorph (inflection<CC: todo: apply modelling), cf. <https://github.com/acoli-repo/acoli-morph/tree/main/unimorph>
  + Finnish (generation<Max: todo: to be updated): <https://github.com/ontolex/morph/blob/master/data/generation/dataset-generation-example.ttl>
  + ?FST (FOMA, Quechua): <https://github.com/ontolex/morph/tree/master/data/foma/quechua>
* external:
  + LEXIS (Greek Parole-Simple dict@Penny)
  + DeriNet/UDer/Universal Derivations
    - Latin@LiLa ?
    - German@Christian: <https://github.com/acoli-repo/acoli-morph> (UDer 0.5 only)
  + SFST: <https://github.com/acoli-repo/acoli-morph> (Morphisto@Christian, German; inflection only)
  + GermaNet compounds: <https://github.com/acoli-repo/acoli-morph> (German@Christian)
  + Morph@Thierry (= Italian samples?)
  + **TODO**: (please list your data, unless described in separate section)
  + **Open requirements**: IGT/ToolBox/FLeX data, inflection tables!

## 3.2 inflection tables (Fahad, others?)

* Latin (?)
* Old English (Fahad): sample data: coman/quoman example, cf. <https://en.wiktionary.org/wiki/cuman#Old_English>
  + issues with dialects (reference dialect vs. other dialects) and diachrony (phonological processes); treatment of syncopation, suppletion, fusion of different roots [=> variants?] ?
* postponed until Fahad has some progress on modelling

## 3.3 semitic consonantal roots (unassigned)

* + from the same consonant cluster, we can generate different POSes
  + cf. <https://en.wikipedia.org/wiki/K-T-B>, <https://en.wiktionary.org/wiki/%D9%83_%D8%AA_%D8%A8>)
    - this cannot (always) be modelled as inflection, as OntoLex requires (at most) one POS per lexical entry
    - note that this page describes vowelized words as “derivatives”: can we model this as derivation ? (but the process occurs in inflection, too)
      * given a real dictionary, can be easily distinguish derivation and inflection?
  + cf. Arabic example from <https://en.wikipedia.org/wiki/Dictionary_of_Modern_Written_Arabic> (from Max)
    - dictionary organized by roots, but root is not made explicit
    - **todo@unassigned**: put an example into GitHub
* discussion postponed until we have a Semitic speaker
  + - Ilan?
      * but first, check Bettina’s conversion of KDictionaries’ Hebrew dict

# 4. AOB

next call in two weeks