Morph telco 2022-06-29, 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 (LDL-2022):** <https://www.overleaf.com/4868363189kczjzdndgxwc> (folder submission/)

**CALL WILL POSSIBLY BE POSTPONED BY 2 WEEKS**

**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)

Sina Ahmadi (SA)

Elena B.

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

[**0. Module draft**](#_1kaaos1w4ub1) **3**

[**1. Publications**](#_5q4h03dxajtd) **3**

[2. definition consolidation / overview](#_n0e2ll1nl5iu) **4**

[2.1 progress of consolidation](#_op9t9tq1v0ds) 4

[2.2 overview](#_hbl1nk25ltov) 4

[2.3 open issues](#_facompgvbtnk) 4

[2.4 InflectionType](#_ytch9o1lp0of) 5

[current model](#_1t7puoeqhzeq) 6

[alternative 1: detach InflectionType](#_2z82laky7bwb) 6

[alternative 2: replace InflectionType by GrammaticalMeaning](#_28k0zvfnaaec) 6

[alternative 3: merge InflectionType with InflectionRule](#_tur9fmagixty) 6

[**2.5 morph(eme) order**](#_ki1gty3ur827) **7**

[2.6 replacement](#_37vo5slljte6) 8

[**3 open problems/other data**](#_jwteu367w2rp) **8**

[3.0 Comparison with MMoOn (Mod. Greek, Hebrew, other; unassigned)](#_uwa8n37v0ip) 8

[3.1 Samples to be modelled (all)](#_s3vk3hv6pguv) 8

[3.2 inflection tables (Fahad, others?)](#_rzzsrcthlbm3) 9

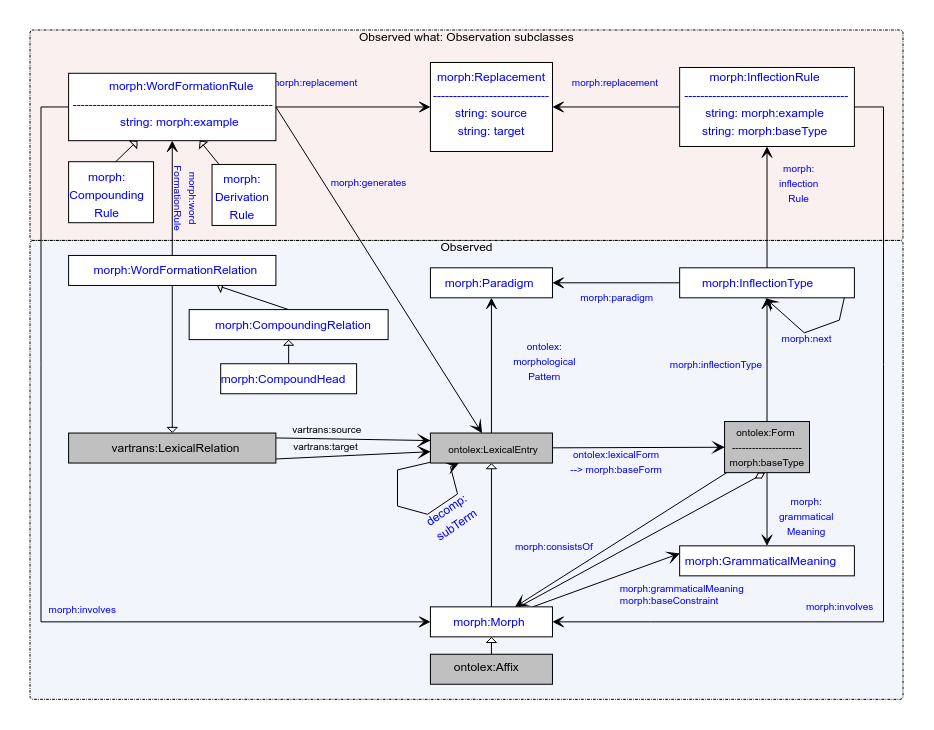
[3.3 semitic consonantal roots (unassigned)](#_w3ikvzzgmbyd) 9

[**4. AOB**](#_4u89668ejc7q) **10**

# 

# 0. Module draft

**draft 4.15 (no changes)**



**Model draft 4.16 updates (to be discussed) & open issues:**

* inflection type to be discussed

# 1. Publications

* LDL
* 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
* LLODREAM? <https://easychair.org/cfp/llodream2022>
* 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
* 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 / overview

## 2.1 progress of consolidation

* **standing TODO@all**: provide/refine/review definitions
  + under <https://github.com/ontolex/morph/blob/master/draft.md>
  + technical definitions, but linguistic explanation (“definition”) in text
  + **@all**: you can contribute suggestions by creating issues (<https://github.com/ontolex/morph/issues>), via pull requests, or by direct editing (share your GitHub username)
  + procedure: for definition refinement:
    - open an issue
    - pull request + close the issue
* status updates?

## 2.2 overview

* **TODO**(@Max?): continue model overview for Elena B. and Sina: Inflection part
  + Sample data for reduplication?
  + Sample data for cliticization? (**TODO@Sina**)
    - cf. Italian: <https://en.wiktionary.org/wiki/andiamoci>, <https://en.wiktionary.org/wiki/andarsene>

## 2.3 open issues

* **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

## 2.4 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
  + **TODO**@Max: example agglutinative
  + **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

### 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.

### 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 !?

### 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”)

Penny+Katerina (summary of last call, tests for fusional language):

* all alternatives express the neccessary information (if a direct link with grammatical meaning is added)
* prefer alternative 2
  + alternatives 1 and 3 are equivalent if a direct link with grammatical meaning is added
  + alternatives 1-3 preferred over current model in terms of verbosity
* CC: that corresponds to my personal preference, too
* CC: minor refinements (to be discussed after applicability to agglutinative language has been shown)
* 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)

**TODO@Max**: differences on agglutinative languages

## 2.5 replacement (wrapup)

conventions for replacement correspond to those of pattern matching/replacement in SPARQL, as formally defined

* in SPARQL 1.1 (<https://www.w3.org/TR/sparql11-query/>), which points to
  + the XPath function replace (<https://www.w3.org/TR/xpath-functions/#func-replace>), and
  + the XPath regex syntax (<https://www.w3.org/TR/xpath-functions/#regex-syntax>)

A more readable, informal description under

* <https://en.wikibooks.org/wiki/SPARQL/Expressions_and_Functions#REGEX>

Note that in the formal syntax definition, “\” is used to mark special characters. However, as most SPARQL engines are Java-based and Java uses “\” as an internal escape symbol, you actually have to write “\\” instead of “\” as defined in the syntax. A literal single “\”-character in a regex must thus be double escaped (i.e., “\\\\”).

Note: This syntax originates from regular expressions in Perl (<https://perldoc.perl.org/perlre>).

Except for minor differences in escaping and special characters, this is equivalent to

* the syntax of regular expressions in Java (<https://www.w3schools.com/java/java_regex.asp>)
* the syntax of regular expressions in Sed (<https://www.gnu.org/software/sed/manual/html_node/Regular-Expressions.html>) (and other Unix command-line tools, e.g., grep)

# 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