Morph telco 2022-05-04, 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. Publication plans**](#_5q4h03dxajtd) **3**

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

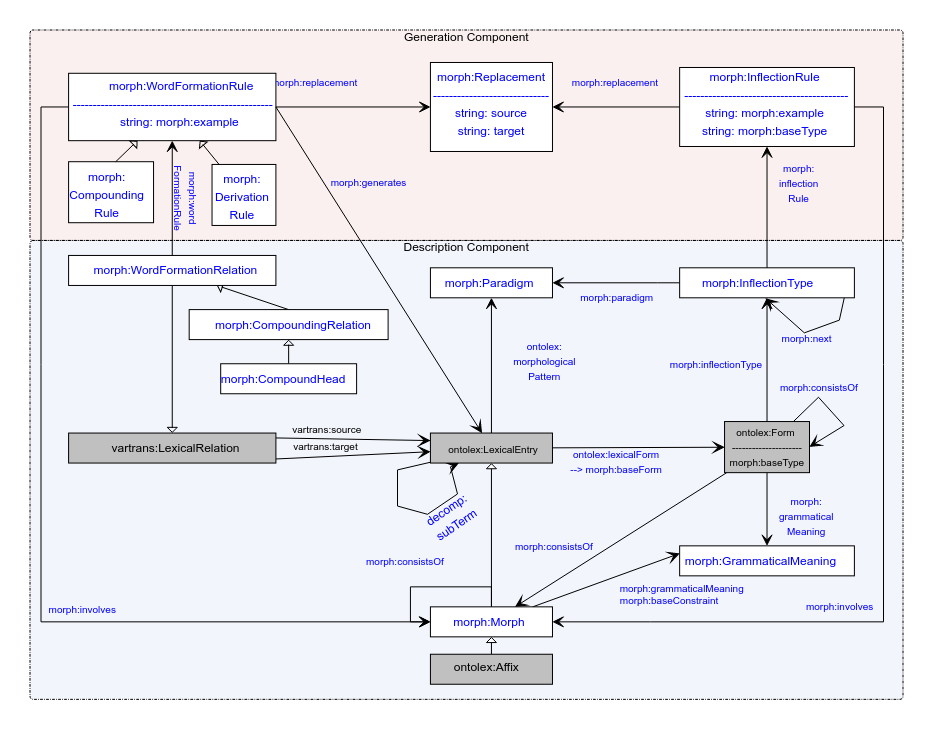
[**2.1 InflectionType**](#_letz9qce1sim) **4**

[**4. AOB**](#_4u89668ejc7q) **6**

# 

# 0. Module draft

**draft 4.14**



**implemented updates (diagram only, not necessarily in github draft):**

* morph:inflects => morph:involves (for parallel with word formation, *also in draft.md*)
* morph:paradigm (between lexical entry and paradigm) => ontolex:morphologicalPattern (we forgot about that, *also fixed in draft.md* ;)
* vartrans:LexicoSemanticRelation => vartrans:LexicalRelation (was an error, *also fixed in draft.md*)
* CompoundRelation => CompoundingRelation (in analogy to derivation)
* CompoundRule => CompoundingRule (tentative consensus)
* 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>
  + **TODO@all**: discuss and maybe come up with the solution
* decomp:subTerm added (TODO: add at definitions, too; CC: DONE)
* text: CC restructured “other data” section below

**Model draft 4.15 updates (to be discussed):**

* inflection type to be discussed
* consistsOf?

# 1. Publication plans

* COLING?
  + may 17
  + could be an altewrnative venue for the LDL submission (if rejeted or revoked) or a novel paper (if we have novel content)
    - decision postponed
  + 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
  + paper@COLING:
    - FK: update of OntoLex, incl. FrAC, Morph, MModality
* LLODREAM? (Matteo, Marco) <https://easychair.org/cfp/llodream2022>
  + 500 words
  + deadline june 15th
  + ?publication
  + PL> Focus on morph, not the overview
* 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: before May 17 (suggested at last call)
* <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
* **old todos** (discussion postponed)
  + **DONE**@CC: issue to define cardinality restrictions: <https://github.com/ontolex/morph/issues/12>
  + **DONE**@CC: update definition of ~~inflects~~ involves in GitHub draft
  + **DONE**@Matteo: read definitions until next telco
  + **DONE**@Penny: read definitions => issue
  + **DONE**@CC: define morph subclasses in LexInfo rather than OntoLex-Morph, also add equivalence axioms (lexinfo:Prefix subclassOf [ lexinfo:termElement lexinfo:prefix ])
    - **DONE**: <https://github.com/ontolex/lexinfo/pull/29>
  + @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?

## 2.1 InflectionType

* revise/abandon inflection type? (needs to be checked on data => discuss German FST data)
  + last call:
    - Max: no, this is a complete revision of generation model
    - we don’t seem to have a common understanding of what InflectionType is meant to be
      1. Max: hard to explain to other people
* 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 allopmporphic 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”)

* 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)

To be discussed next time.

# 4. AOB

next call in two weeks