Morph telco 2022-05-17, 13:15 CEST

**Link:** [**https://meet.google.com/nsj-tbcy-yop**](https://meet.google.com/nsj-tbcy-yop) **[CHECK HERE FOR UPDATED LINK(S)]**

**Latest Definitions:** [**https://github.com/ontolex/morph/blob/master/draft.md**](https://github.com/ontolex/morph/blob/master/draft.md)

**Nexus:** [**https://nexuslinguarum.eu/the-action/join-us**](https://nexuslinguarum.eu/the-action/join-us)

**Participants [please add yourself]:**

Christian Chiarcos (CC) (excused for being 10 min late)

Max Ionov (MI)

Katerina Gkirtzou (KG)

Besim Kabashi (BK)

Fahad Khan (FK)

Khadija Ait ElFqih (KAE)

Matteo Pellegrini (MP)

Ciprian-Octavian Truică (CT)

Penny Labropoulou (PL)

Elena Simona Apostol (ESA)

Sina Ahmadi (SA)

Elena Benzoni (EB)

Petra Steiner (PS)

Theodorus Fransen (TF)

Thierry Declerck (DFKI)

Ranka Stanković (RS)

Gilles Sérasset (GS)

Mike Rosner (MR)

Table of Contents

[0. Module draft (4.17)](#_v4addnpz2vtu)

[1. Old Irish](#_3b2yols64ffs)

[2. Clitics](#_av83soukzc3q)

[3. Character/sound classes](#_3c1slk2we2i3)

[Maltese example](#_bgy7leujpwjo)

[Defining classes](#_ttwrnvbsp5y3)

[Other examples](#_1f6l1qle86kx)

[Next time](#_nk5j7w1igusi)

# 0. Module draft (4.17)



# 1. Old Irish

* [short data presentation], mainly based on [spreadsheet](https://docs.google.com/spreadsheets/d/1zDLWok03CJJD8rEY_IZtUtIo_MONGQGr1s82P0mrvuc/edit#gid=1855389609)
* Data currently on hand is expressible in Ontolex-core + Ontolex-morph
  + Qs TF:
    - [Morph vs word, compare UD](https://docs.google.com/spreadsheets/d/1zDLWok03CJJD8rEY_IZtUtIo_MONGQGr1s82P0mrvuc/edit#gid=1855389609)
      * Interlinking/interoperability issues?
    - [Can/should we segment a stem into morphs](https://docs.google.com/spreadsheets/d/1zDLWok03CJJD8rEY_IZtUtIo_MONGQGr1s82P0mrvuc/edit#gid=1733676412)?
      * MI: Should we talk about stems as Morphs? Does it mean that any morph is subdividable? Technically, it is allowed since they are all subclasses of a LexicalEntry which can consists of Morphs
* Generation rules (when they are there) are too complex to handle in Ontolex-morph
* Can be expressed with FSTs
  + FST functionality within next modules?
  + Should FST be a part of Ontolex or a completely separate vocabulary? What are the use-case for having it as a part of Ontolex?
  + If as an OntoLex module:
    - Together with diachrony?
    - Together with sound classes?

# 2. Clitics

Spanish reflexive clitic *se*:

* *Llamar* — to call, *llamarse* — be named
  + No problems here, a separate lexical entry
* When conjugated, clitic goes to the left and separated orthographically: ¿Cómo te llamas? — what is your name?
  + Still, it is one phonetic word (e.g. single stress)

:llamarse a ontolex:LexicalEntry ;

ontolex:canonicalForm :llamarse\_form\_inf ;

ontolex:otherForm :llamarse\_form\_2sg .

:llamarse\_form\_inf a ontolex:Form ;

ontolex:writtenRep “llamarse” ;

lexinfo:... .

:llamarse\_form\_2sg a ontolex:Form ;

ontolex:writtenRep “te llamas” ;

lexinfo:number lexinfo:singular ;

lexinfo:person lexinfo:second ;

… .

No problems for generation as well (rules) as well:

<llamarse\_rule\_2sg> a morph:InflectionRule ;

morph:paradigm <llamarse\_paradigm> ;

morph:involves <clitic\_2sg>, <suffix\_as\_pres\_2sg> ;

morph:replacement [

a morph:Replacement ;

morph:source "^(.\*)arse$" ;

morph:target "te \1as" ;

] .

Inflection:

:llamar a ontolex:LexicalEntry ;

ontolex:canonicalForm :llamar\_form\_inf ;

ontolex:otherForm :llamarse\_form\_2sg .

:llamar\_form\_inf a ontolex:Form ;

ontolex:writtenRep “llamar” ;

lexinfo:... .

:llamar\_form\_2sg a ontolex:Form ;

...

:llamarse\_form\_2sg a ontolex:Form ;

ontolex:writtenRep “te llamas” ;

lexinfo:number lexinfo:singular ;

lexinfo:person lexinfo:second ;

lexinfo:reflexivity?? lefinfo:reflexive??

… .

No problems for generation as well (rules) as well:

<llamarse\_rule\_2sg> a morph:InflectionRule ;

morph:paradigm <llamarse\_paradigm> ;

morph:involves <clitic\_2sg>, <suffix\_as\_pres\_2sg> ;

morph:replacement [

a morph:Replacement ;

morph:source "^(.\*)arse$" ;

morph:target "te \1as" ;

] .

**TODO**: add a pull request with an example of this to the documentation

**TODO**: check what people think about cliticization

# 3. Character/sound classes

## Maltese example

*kiteb* → *ktibt* (PERF.1SG)

With character/sound classes:

<kiteb\_perf\_1sg> a morph:InflectionRule ;

morph:paradigm <kiteb\_paradigm> ;

morph:involves <suffix\_t\_perf\_1sg> ;

morph:replacement [

a morph:Replacement ;

morph:source "(C)(V)(C)(V)(C)" ;

morph:target "\1\3i\5t" ;

] .

Without character/sound classes:

<kiteb\_perf\_1sg> a morph:InflectionRule ;

morph:paradigm <kiteb\_paradigm> ;

morph:involves <suffix\_t\_perf\_1sg> ;

morph:replacement [

a morph:Replacement ;

morph:source "(ċ|d|n|r|s|t|x|ż|z|b|f|ġ|g|għ|h|ħ|j|k|l|m|p|q|v|w)(a|e|i|o|u|ie)(ċ|d|n|r|s|t|x|ż|z|b|f|ġ|g|għ|h|ħ|j|k|l|m|p|q|v|w)(a|e|i|o|u|ie)(ċ|d|n|r|s|t|x|ż|z|b|f|ġ|g|għ|h|ħ|j|k|l|m|p|q|v|w)" ;

morph:target "\1\3i\5t" ;

] .

Technically similar albeit bulky, visually much less clear — difficult to interpret or to see differences between different rules.

## Defining classes

**Model change**: adding one class

<vowels> a <CharacterClass> ;

rdfs:label "V" ;

rdfs:member "a", "e", "i", "o", "u" .

For form generation: adding a replace for each class.

If decided not to use this in the model this will probably be used ad-hoc → rules in the datasets will be **not interoperable**.

## Other examples

* Vowel harmony rules in Turkic and Finno-Ugric languages (e.g. Turkish, Finnish)
  + Vowels in the replacement depend on vowels in the root
  + An affix can be -lla or -llä for a Finnish case

morph:replacement [

a morph:Replacement ;

morph:source "(.\*FRONT\_V.\*)$" ;

morph:target "\1llä" ;

],

a morph:Replacement ;

morph:source “(.\*BACK\_V.\*)$” ;

morph:target “\1lla” ;

] . # Finnish Adessive case

* Rules like “If the stem of the noun ends in a vowel, the buffer consonant *y* is added”

morph:replacement [

a morph:Replacement ;

morph:source "(V)$" ;

morph:target "\1ya" ;

],

a morph:Replacement ;

morph:source “(C)$” ;

morph:target “\1a” ;

] . # Turkish Accusative and Dative

**Note**: in both cases we will apply two alternative regexes in the same rule

# Next time