**Participants:**

Bettina

Max

Fahad

John

Maria

MI: Note, that different languages and different linguistic traditions describe morphology and paradigms in different ways. Most notably, there are 3 established models of morphology:

1. Item and Arrangement, IA
2. Item and Process, IP
3. Word and Paradigm, WP

*Paradigm tables*, like the Latin one above are common for the 3rd one, WP. They are rooted in the *European linguistic traditions*, since they are suited well for traditional Indo-European languages like Ancient Greek and Sanskrit. Since there are a lot of fusion, it is quite difficult and often quite useless to try and separate affixes and endings.

For *agglutinative languages*, it is more common to *separate affixes and simply give them as morphological information (so, like the first table)*, so, IA model. And it is often tricky to give full forms in paradigms because there are a lot of combinations (even for inflection). Example: a common Turkic verbal inflection has more than 100 forms. One can argue that some of these forms are derivational, but another one can argue against that.

Also, for some agglutinative languages and other languages with complex morphophonological processes, it is often very difficult to reconstruct forms from tables like these.

Take, for instance, Turkic once again, or Uralic languages, with vowel synharmony. Vowels in affixes affect vowels in the root. This can be, in principle, be modeled with several variants of a root, but this is not the only case these processes happen.

Morphologies for these languages are often described with various IP models. Computationally, usually with some variant of finite state transducers (xfst, foma, 2-level morphology (pc-kimmo), etc.)

The problem with this approach is that for specifying even a small part of morphology, sometimes a lot of rules should be implemented. Also, the rules are not necessarily regular, most are context-free, and sometimes even context-free.

Example: Paul Meurer implementation of Abkhaz morphology spans over thousands lines of rules, and it takes 64Gb or RAM to compile the automata (and a lot (but less) to load).

**An approach for the automatic generation (CC + Max)**

* All names suggested below are preliminary. Unclear to us whether they overlap with previously proposed notions such as “Paradigm” or “Pattern”.
* *morph:Paradigm*: theoretically motivated type of inflection (~ “*-i-stem nouns”* in the first Latin example above). Cf. morphemes, theoretical construct, may contain allomorphs, exceptions, etc. Can have multiple inflection types linked to it (at least one).
  + Idea: a paradigm is a generalization over different inflection types. In particular, different assimilation rules may apply to different stem forms. In a paradigm, these are not distinguished
* *morph:InflectionType*: a group of flexia for a group of words. No allomorphy. (~ “*Parisyllabic rule*”, “*double consonant rule*”, etc.in the second Latin example table above)
  + Idea: If we associate an inflection type with a stem, we can unambiguously generate all forms
  + Paradigms are collections of inflection types
* *morph:Rule* (?pattern?)
  + Inflection types are collection of rules, that associate grammatical features (say, case and number) with a particular string transformation
  + Datatype property *morph:pattern*: A transformation instruction (Perl-like regex) that describes how to generate the inflected form from the stem, e.g., “*s/is$/em*/” for generating *amnem* (second Latin example above) from *amnis*
    - If there is more than one morph:pattern, this means that two equivalent transformations can apply, e.g., for *amnim* from *amnis*. Normally, this should be one.
  + Rules roughly correspond to the entries of the first Latin table above.
* If paradigms are not fully fleshed out but described by examples, create an inflection type for each example (say, *amnis*, *pars* etc. for the second Latin example above)
  + in the inflection type we have datatype properties
    - *morph:example* (no more than 1): one dictionary form of the word that is inflected according to this inflection type (string)
  + Rules (or pattern, at least 1):
    - Grammatical categories (nom, sg, etc.)
    - PCRE-compatible Regex as a string
    - Example (string)

**An approach for the automatic generation (John)**

<#cat> a ontolex:Word ;

ontolex:canonicalForm [ontolex:writtenRep "cat"@en] ;

ontolex:morphologicalPattern <#regular\_english\_noun> .

<#english\_plural> morph:belongsToMorphPattern <#regular\_english\_noun> ;

lexinfo:number lexinfo:plural ;

morph:consistsOf [a morph:StemMorph] ,

[a morph:AffixMorph ; morph:representation "s"@en] .

<#english\_genitive> morph:belongsToMorphPattern <#regular\_english\_noun> ;

lexinfo:number lexinfo:singular; lexinfo:case lexinfo:genitive ;

morph:consistsOf [a morph:StemMorph] ,

[a morph:AffixMorph ; morph:representation "’s"@en] .

Each paradigm generates a unique form (exactly one form).

A morphological pattern is a set of paradigms

A paradigm is a set of forms sharing a fixed set of grammatical category values (e.g. gender and number) but deviate in another grammatical category value (e.g. different cases).

A form is composed of morphs

amnis = ontolex:LexicalEntry

third declension i-stem = morph:MorphologicalPattern

femine singular = morph:InflectionalParadigm or

feminine singular nominative = morph:InflectionalParadigm

aminis, amnem, amnim, aminis... = ontolex:Form

|  | **Third declension paradigm**  **(*i*-stem nouns)** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Masculine &**  **Feminine** | | **Neuter** | |  |
| **Singular** | **Plural** | **Singular** | **Plural** |  |
| **Nominative** | — | -ēs | — | -ia |  |
| **Accusative** | -em  -im | -ēs  -īs |  |
| **Genitive** | -is | -um  (-ium) | -is | -um  (-ium) |  |
| **Dative** | -ī | -ibus | -ī | -ibus |  |
| **Ablative** | -e  -ī |  |