Grammar Customization
Morphotactics in the Grammar Matrix
Lab 3 Phenomena

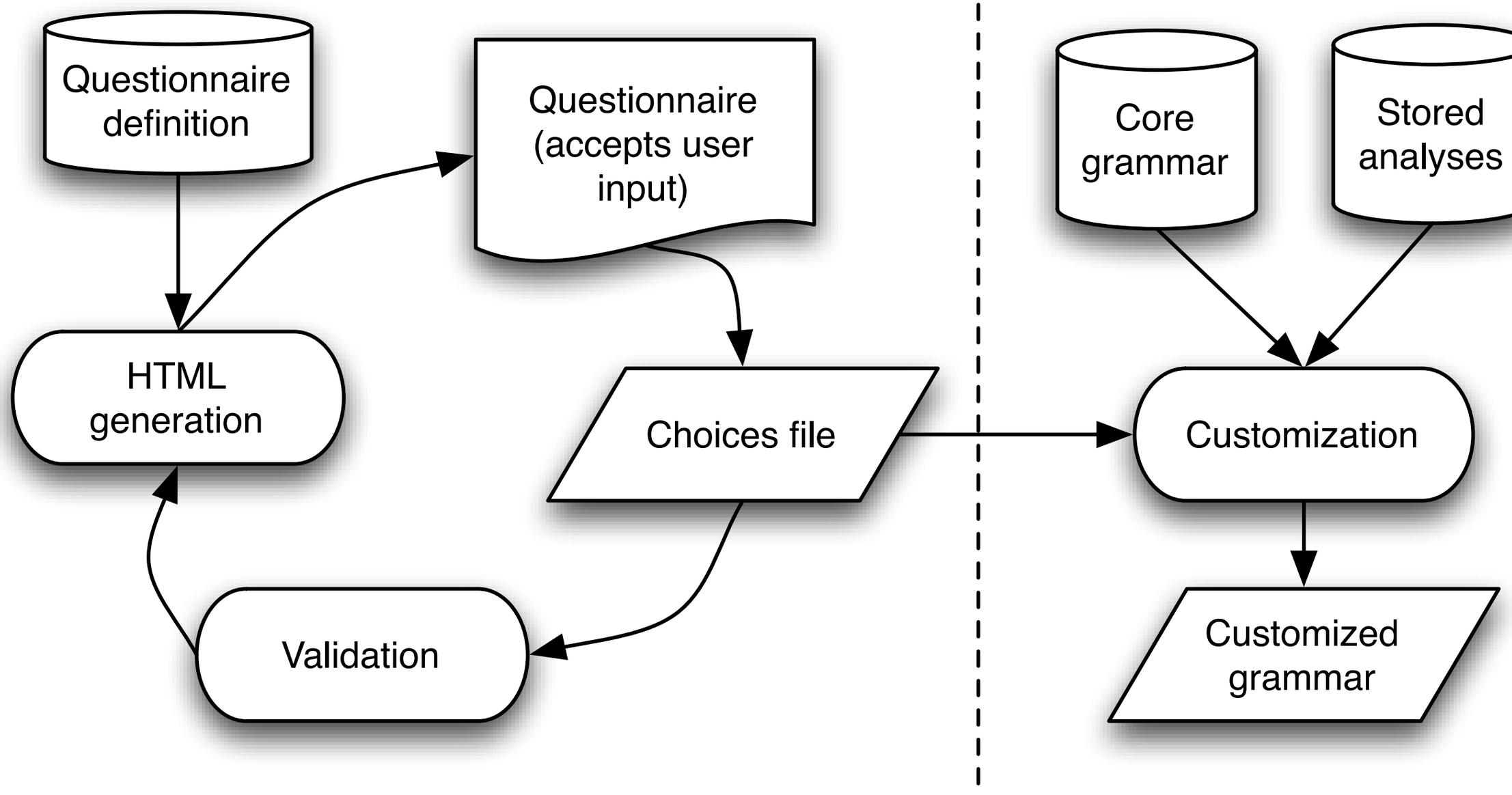
Ling 567
April 11, 2017

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Elicitation of typological information

Grammar creation



(Bender et al 2010)

Creating a library for the customization system

- Choose phenomenon
- Review typological on phenomenon
- Refine definition of phenomenon
- Conceptualize range of variation within phenomenon
- Review HPSG (& broader syntactic) literature on phenomenon
- Pin down target MRSs
- Develop HPSG analyses for each variant
- Implement analyses in tdl
- Develop questionnaire
- Run regression tests
- Test with pseudo-languages
- Test with illustrative languages
- Test with held-out languages
- Add tests to regression tests
- Add to MatrixDoc pages

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Morphology: Basics

- Morpheme: The smallest meaningful unit of language/smallest pairing of “form” and “meaning”
- But:
    - “form” can be lots of things, including empty but also messy changes to word form
    - “meaning” can be just syntactic features
- Morphotactics: Which morphemes can combine, in what order
- Morphophonology: Relationship between underlying word forms and surface forms
- Morphosyntax: Relationship between morphemes and syntactic and semantic features

Morphology: Example

slolmáyaye
slol-ma-ya-yÁ
know-1SG.PAT-2SG.AGT-know

‘you know/knew me’ [lkt]

- Infixation, vowel harmony: Morphophonology
- Relative order of PAT and AGT marker, optionality of same: Morphotactics
- Mapping to constraints that the patient argument be 1sg and the agent 1pl: Morphosyntax
- Actually parsing the string: priceless!

What morphophonolgy can the LKB & the customization system handle?

	LKB	Customization System
polite concatenative morphology	✓	✓
zero morphemes	✓	✓
morphologically conditioned allomorphy	✓	✓
phon. chnages at morpheme boundary	✓	
ablaut		
infixation		
vowel harmony		
suppletion		

Assume a morphophonological analyzer...

- Morphophonological analyzers map surface forms to underlying strings of morphemes

- FSTs are up to the task (except for open-class reduplication)
  - XFST (Beesley & Karttunen 2003) is a very linguist-friendly set up; FOMA (Holden & Algeria 2010) is a open-source package with similar functionality

- But you don't need to build one for this class!

- Use the morpheme segmented line of your IGT to represent what it would map to, and then (if you have any interesting morphophonology) have that line be the target for your grammar.

Morphophonology/morphosyntax boundary: Where to draw the line?

- Underlying morphemes can be represented as a sequence of phonemes or as symbols representing morphological features.

- A canonical XFST-derived analyzer will also include POS tags as a morphological feature in the underlying form.

- From the point of view of the LKB:

  - The POS tag adds nothing

  - Spelling the morphemes as morphological features adds nothing: we still need a lexical rule that maps those strings to constraints on avms

Morphophonology/morphosyntax boundary: Where to draw the line?

- On the other hand: for XFST/FOMA, the POS tags (and maybe features) can be useful intermediate stages in processing

- The features can make it easier to create gloss lines automatically.

- On the third hand: using sequences of morphemes might make LKB input/output comprehensible to speakers

- So what should the upper tape have?

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Basic concepts

- Position class: A supertype to lexical rules which fit in the same slot
- Lexical rule type: *lex-rule* and its subtypes, all have DTR feature
- Lexical rule instance: A grammar entity (manipulatable by the LKB) which inherits from a lexical rule type and specifies a spelling change (including no change).
- Forbids constraint: A specification in the customization system stating that a stem lexical rule type (including a position class) cannot co-occur with another lexical rule type, instance, pc or stem.
- Requires constraint: A specification in the customization system stating that a stem lexical rule type (including a position class) must co-occur with another lexical rule type, instance, pc or stem.

Position classes, inputs and lexical rule hierarchies

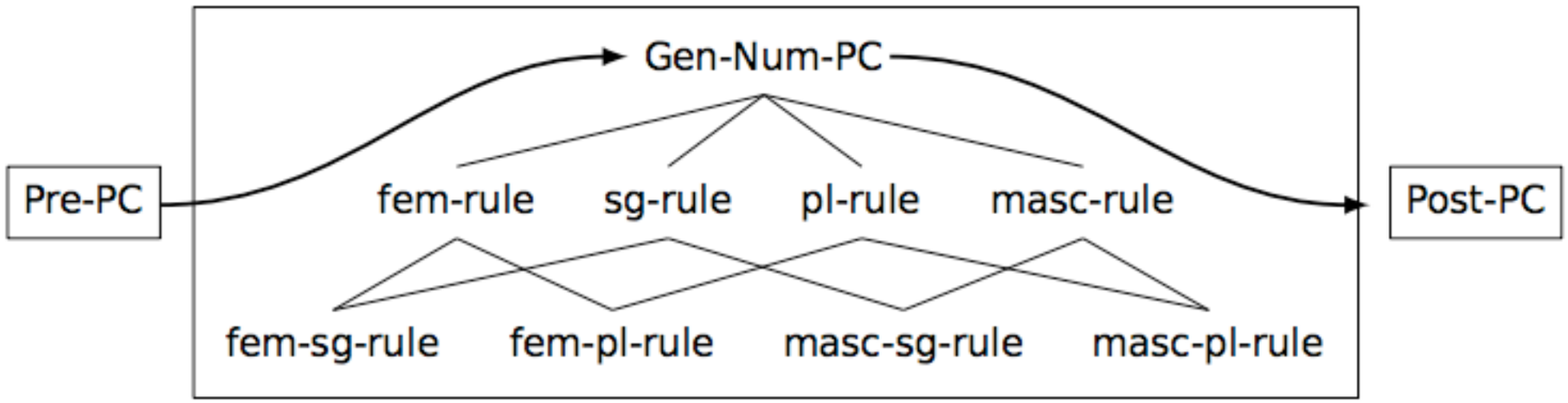


Figure 9: Example lexical rule type hierarchy in a position class

(Goodman 2013)

To define a position class

- Required:
    - Whether or not it is obligatory
    - Possible inputs and prefix/suffix
        - = position in the string
- Optional:
    - Requires/forbids constraints

To define a lex rule type

- Required
    - Nothing (though defaults fill in)
- Optional
    - Name
    - Supertype (if it doesn’t inherit directly from its position class)
    - Feature/value pairs (optional, but this is usually the point!)
    - Requires/forbids constraints

To define a lex rule instance

- Required
  - Affix v. no affix
  - Spelling for affix
- Optional
  - Nothing

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

tdl files

- matrix.tdl: Supertypes for lex-rules, which handle the copying up of everything you're not changing

- my_language.tdl: Position classes and lex rule types defined through the customization system; features for inside INFLECTED

- lrules.tdl: Instances for non-spelling-changing lex rules (zero morphemes)

- irules.tdl: Instances for spelling-changing lex rules

Handling of morphotactics

- Rule order handled through super types and typing the DTR feature
- Requires/forbids through the INFLECTED feature

```
case-lex-rule-super := Representative-rule-dtr &
                       add-only-no-ccont-rule &
                       noun-telic-rule-dtr &
[ INFLECTED [ CASE-FLAG +,
              INNER-NEGATION-FLAG #inner-negation,
              NUMBERED-FLAG #numbered ],
  DTR case-rule-dtr &
      [ INFLECTED [ INNER-NEGATION-FLAG
                    #inner-negation,
                    NUMBERED-FLAG #numbered ] ] ].
```

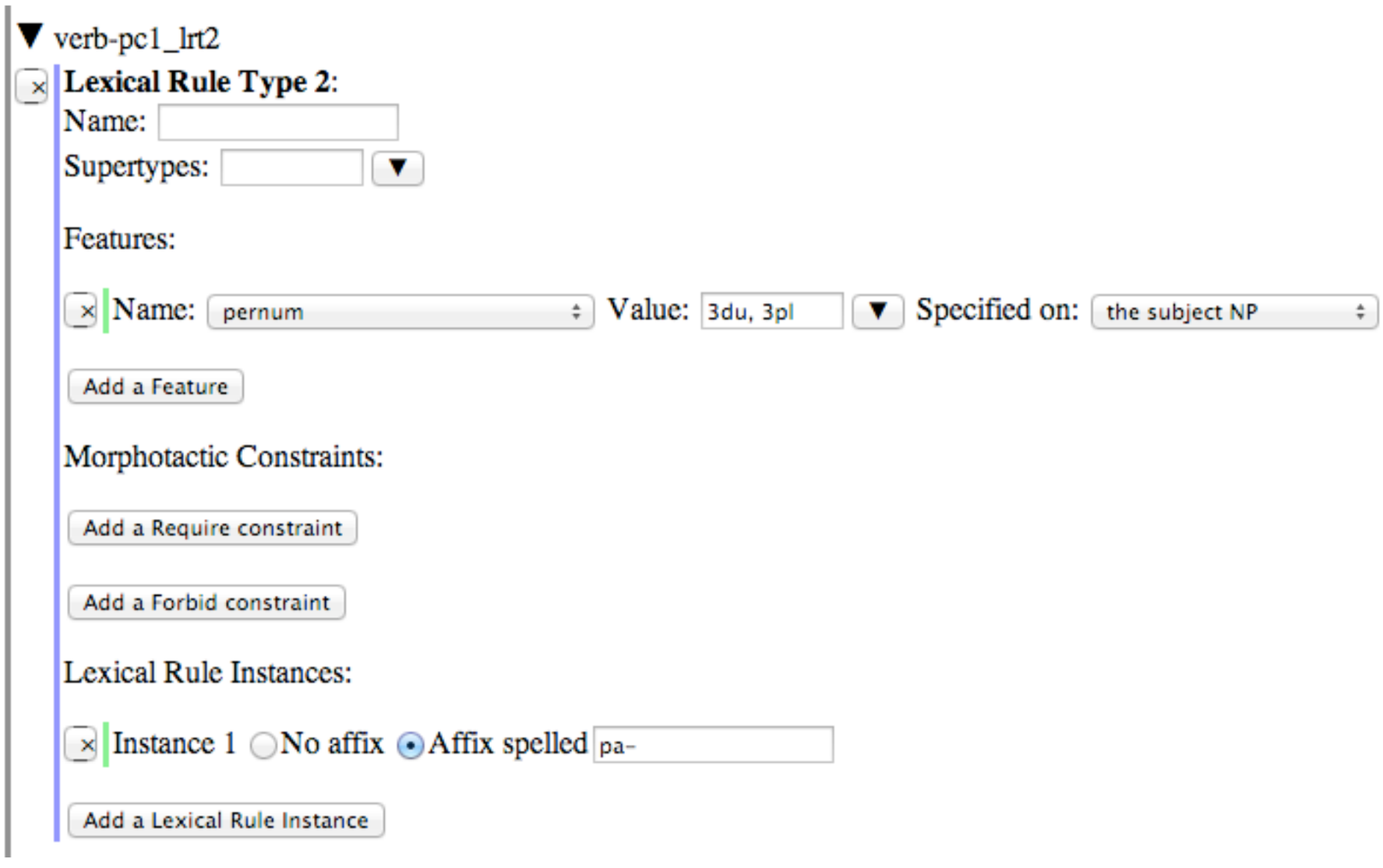
Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Agreement: Verbs and arguments

- The SHAC doesn't scale:
  - Verbs can agree with more than one argument
  - Many agreement features we actually want represented in the semantics
- Solution:
  - Agreeing verbs constrain the PNG features of their arguments
  - Typically implemented via lexical rules

Agreement: Example



Agreement: Nouns and determiners

- Determiners often agree with nouns in gender, number, and case.
- Sometimes the only overt mark of these features is on the determiners.
- Gender is usually inherent in nouns (= define on the lexical classes).
- Case comes from lexical rules applying to nouns (if it's overtly marked there) and/or constraints on the SUBJ and COMPS lists of the selecting verb.
- In the customization system, constraints on features of determiners are actually applied to the SPEC value so they enforce agreement with the nouns.

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Argument Optionality

- Most (all?) languages allow selected arguments to be unexpressed in certain circumstances
- The variation comes in what those circumstances are:
    - Subjects only/all arguments
    - Only if there is agreement marking on the verb/regardless of agreement marking on the verb
        - Some languages disallow the markers on the verb if the arguments aren’t dropped
    - Certain tense/aspect or person/number combinations

Argument optionality: HPSG analysis

- The analysis provided by the customization system involves non-branching phrase structure rules which shorten a valence list without realizing an overt dependent

- The application of the rules is controlled by the feature OPT:
  - [OPT -] arguments may not be dropped
  - (SPR values that must be realized by a bare-np rule are [OPT +]...)

- Lexical rules for agreement (or lack thereof) can constrain the OPT value and/or shorten the SUBJ/COMPS lists: this is handled with the pseudo-feature OVERT-ARG

Overview

- Grammar customization
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Argument optionality
- Tense/aspect

Tense

- Tense encodes the three-way relationship between speech time, event time and an anaphorically determined reference time.
    - Past/present/future
    - Past/non-past
    - Future/non-future
    - Recent/remote past/future

Aspect

- Aspect encodes the description of the internal temporal properties of the event and how it's viewed.
    - Internal temporal properties: Aktionsart or situation aspect
    - How it’s viewed: Viewpoint aspect
- Perfective/imperfective
    - NB: Perfective != perfect
- Inceptive, continuative, progressive, habitual, iterative…

Lab 3

Test Suite

The first task is to create positive and negative example sentences illustrating the following phenomena, to the extent that they are relevant for your language. All grammars should do the following:

- tense/aspect
- negation

And then you should choose at least three of the following (provided you can find info for three). That is, you may skip one:

- agreement
- argument optionality
- possessives
- valence changing lexical rules

Before you start, read the general instructions for testsuites and the formatting instructions.

Back to top

Starter grammar

The second task is to update your starter grammar by filling out the required sections of the Grammar Matrix customization questionnaire. The goal here is to get as much coverage as you can over your test suite using only the customization system (no hand-editing of tdl files yet). In particular, you'll need to address these sections:

- Tense and aspect
- Word order (if you need to add auxiliaries)
- Argument optionality
- Sentential negation
- Morphology
  - Rules for agreement affixes
  - Rules for tense/aspect markers
  - Rules for any other affixes that are required so you get fully inflected forms
  - Rules for your valence changing affix (if you went that route)
- Lexicon
  - Stem forms (rather than full forms) of verbs
  - Auxilairies, if necessary
  - Additional nouns/verbs if you have expanded the vocabulary in your testsuite

Begin by uploading your choices file from lab 2, and modify from there. Ask lots of questions!

Things to discuss

- Reminders: Test early & test often!
- Keeping the scope manageable, especially in the face of tempting tables and paradigms
- When the customization system can’t handle something
    - What should go in the testsuite?
    - What should go in the choices file?
    - What should go in your write up?
- Clitics v. affixes

Things to discuss

- Homophony (e.g. demonstrative pronouns and demonstrative determiners/ adjectives; subject and object markers)
- Disjunctive feature specifications as a way to reduce homophony
- Which PCs should be “obligatory”?
- Send me things to demo by noon on Thursday!