Morphotactics in the Grammar Matrix
Lab 3 Phenomena

Ling 567
Jan 23, 2024

Overview

- Reminders while grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

While grading lab 2

- Make the test sentences even simpler
- Read the directions very carefully
- Ask me more questions

- Testsuite and grammar are independent
    - Just because we didn’t do wh questions in the grammar in Lab 2 doesn’t mean there shouldn’t be ungrammatical examples for wh questions

Overview

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

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

2 Morphology: Introduction 11

#7 Morphemes are the smallest meaningful units of language, usually consisting of a sequence of phones paired with concrete meaning. 11
#8 The phones making up a morpheme don't have to be contiguous. 11
#9 The form of a morpheme doesn't have to consist of phones. 13
#10 The form of a morpheme can be null. 13
#11 Root morphemes convey core lexical meaning. 14
#12 Derivational affixes can change lexical meaning. 16
#13 Root+derivational affix combinations can have idiosyncratic meanings. 17
#14 Inflectional affixes add syntactically or semantically relevant features. 18
#15 Morphemes can be ambiguous and/or underspecified in their meaning. 19
#16 The notion 'word' can be contentious in many languages. 20
#17 Constraints on order operate differently between words than they do between morphemes. 21
#18 The distinction between words and morphemes is blurred by processes of language change. 22

#19 A clitic is a linguistic element which is syntactically independent but phonologically dependent. 23
#20 Languages vary in how many morphemes they have per word (on average and maximally). 24
#21 Languages vary in whether they are primarily prefixing or suffixing in their morphology. 25
#22 Languages vary in how easy it is to find the boundaries between morphemes within a word. 26

3 Morphophonology 29

#23 The morphophonology of a language describes the way in which surface forms are related to underlying, abstract sequences of morphemes. 29
#24 The form of a morpheme (root or affix) can be sensitive to its phonological context. 29
#25 The form of a morpheme (root or affix) can be sensitive to its morphological context. 31
#26 Suppletive forms replace a stem+affix combination with a wholly different word. 32
#27 Alphabetic and syllabic writing systems tend to reflect some but not all phonological processes. 33

4 Morphosyntax 35

#28 The morphosyntax of a language describes how the morphemes in a word

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

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

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

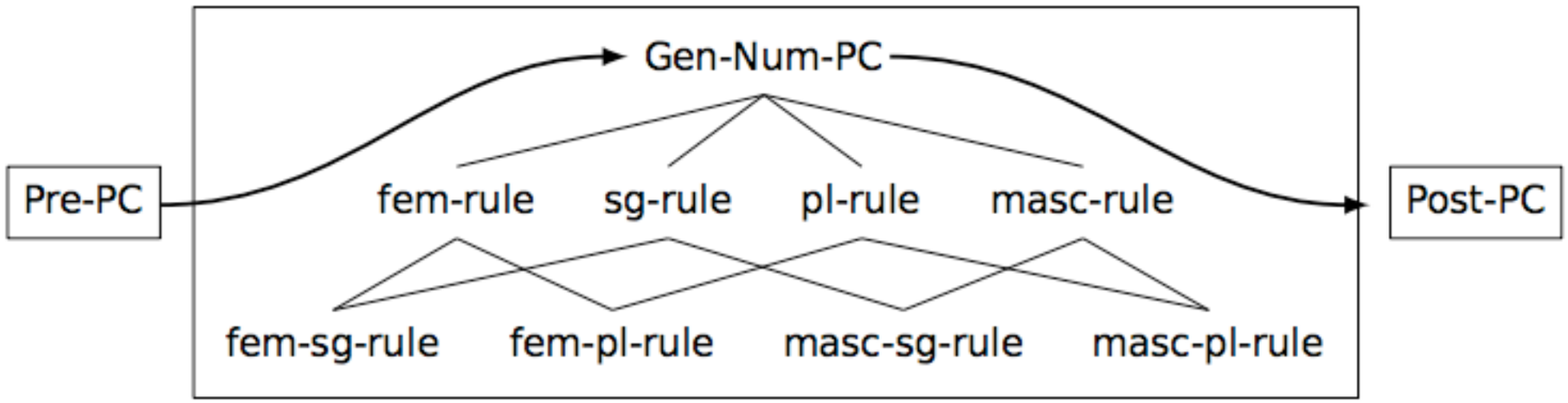


Figure 9: Example lexical rule type hierarchy in a position class

(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 (use sparingly!)

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 (use sparingly!)

To define a lex rule instance

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

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

Overview

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

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 ] ] ].
```

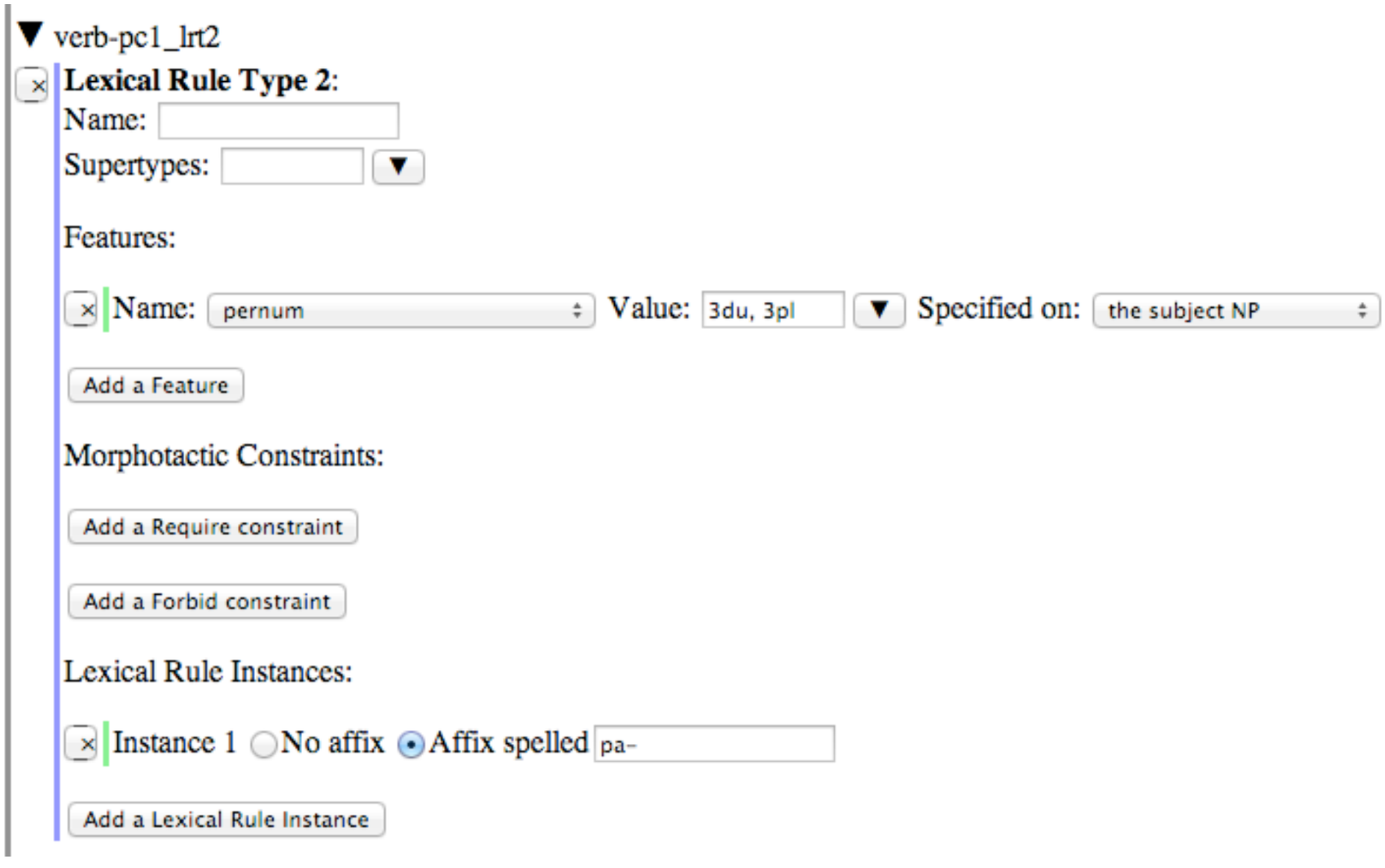
Overview

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

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

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

Coordination

- *And* coordination only (not *with*)
- How many marks?
- What kind of mark?
- Where does the mark go?
- What types of constituents are coordinated with the same strategy?

Agreement in Coordination

- ‘nouny’ coordination only
- Feature resolution
- Closest conjunct

Overview

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

Negation

- Sentential negation only
- What kinds of marks?
- How many marks?

Overview

- Reminders after grading Lab 2
- Morphology: Who's job is it anyway?
- Morphotactics in the customization system
- Morphotactics in customized grammars
- Agreement
- Coordination
- Negation

Lab 3: To web page