Grammar Development with LFG and XLE

Miriam Butt
University of Konstanz

Last Time

- LFG and XLE basics
- C-structure and f-structure
- Functional annotation
- Unification/Consistency, Completenes and Coherence
- Templates
- XLE Walkthrough

This Time: Lesson 3

1. Lexical Rules
   - Passive
   - English Dative Alternation (Dative Shift)
   - Interactions among Lexical Rules
2. Different types of functional equations/constraints

Lexical rules (vs. Transformations)

- A feature that LFG is very well known for is the *Lexical Rule.*
- At the time LFG was invented, generalizations between certain types of sentences were thought of in terms of syntactic *transformations.*
- A famous example involved the passive.
- Linguistic Observation: active clauses are related to passive clauses via a generalizable rule.
    - » Active: *The tiger chased the cat.*
    - » Passive: *The cat was chased by the tiger.*

Transformations

- For example, within Transformational Grammar the rule for the English passive looked something like this:

  NP1 V NP2 → NP2 AUX V by NP1

- In our example:

  NP1 = the tiger   NP2 = the cat

  V = chased   Aux = was

- Over time, however, it was realized that this was not the best way to express what happens with passives across languages.

Lexical rules

- Work by David Perlmutter and Paul Postal showed that the relationship between active and passive was best understood in terms of grammatical relations.

- In LFG terms, this was formulated in terms of a Lexical Rule:
  - OBJ → SUBJ
  - SUBJ → Adjunct or OBL-AG (OBL agent)

  - Verbs which allow for the passive encode this rule as part of their lexical entry.

Lexical rules

- Not all verbs allow for passivization.
- Passives are generally formed with agentive (di)transitive verbs.
- But not stativees, for example.
  - » *The dog had a bone.*
  - » **A bone was had by the dog.*
  - So: LEXICAL property of verbs.
  - And encoded in terms of Lexical Rules.

Lexical rules in XLE

- Lexical rules are a special kind of template.
- They always take a subcategorization frame as their (single) parameter.
- They contain at least two disjuncts, namely on for the canonical subcategorization frame and (an)other(s) for derived subcategorization frame(s).
- Relations between original and derived grammatical functions are stated as follows:

```
(^ OBJ)-->(^ SUBJ)
```

Example of a lexical rule

```
PASSIVE(FRAME) =
    { FRAME
      (^ PASSIVE) = -
    | FRAME
      (^ PASSIVE) = +
      (^ OBJ)-->(^ SUBJ)
      { (^ SUBJ)-->(^ OBL-AG)
      | (^ SUBJ)--> NULL
      }
    }.
```

Demo

grammar2.lfg
testsuite2.lfg

passives
parse-testfile

Testsuites (again)

- Note that testsuite2.lfg builds on testsuite1.lfg
- ALWAYS work with a testsuite.
- ALWAYS test the previous sentences you had already implemented (or excluded from being parsed).
- Make sure you incrementally increase the coverage of your grammar, rather than simply changing which phenomena it can cover.

Argument Alternations

- The Passive is an example of what is more generally known as an *Argument Alternation.*
- Another famous Argument Alternation is the English *Dative Shift.*
  - » *The girl gave a bone to the dog.*
  - » *The girl gave the dog a bone.*
- The Dative Shift can also be treated via a Lexical Rule.
- Again, not all verbs readily allow Dative Shift.
  - » *The girl pulled the bone to the dog.*
  - » **The girl pulled the dog a bone.*

Argument Alternations

- The Passive and the Dative Shift interact
  - » *The girl gave a bone to the dog.*
  - » *A bone was given to the dog (by the girl).*

  - » *The girl gave the dog a bone.*
  - » *The dog was given a bone (by the girl).*

- In LFG this can be modeled by an interaction between two Lexical Rules.
- The ditransitive template calls up the Passive and Dative Shift (the order needs to be right).

Dative Shift

- In the Dative Shift
  - the SUBJ stays the same
  - the OBJ and the OBL (PP Argument) undergo the alternation

    » *The girl gave a bone to the dog.*
    SUBJ OBJ OBL-TO

    » *The girl gave the dog a bone.*
    SUBJ OBJ OBJ2

- Lexical Rule: OBJ --> OBJ2
  OBL-TO --> OBJ

Dative Shift lexical rule

```
DAT-SHIFT(FRAME) =
    { FRAME  "base case"
     | FRAME   "or dative shift"
      (^ OBJ)--> (^ OBJ2)
      (^ OBL-TO)--> (^ OBJ)
     }.
```

Unification

- LFG is based on a unification formalism.
- The information specified via the functional annotations is unified into an f-structure representation.
- Thus, different parts of the grammar can specify information about the same feature-value pair.
- However, this information must unify.
- Example: subject-verb agreement.
  - Subject Noun: (^ NUM) = sg
  - Verb: (^ SUBJ NUM) = sg

Various Types of Constraints

- Most of the equations we have seen so far have been *Defining Equations*.
- However, the LFG formalism allows for various other types of equations.
- Several others seen so far as well:
  - existential constraint (as part of the Count Noun Template in Lesson 2)
  - constraining equation (as part of the demo of grammar2.lfg in this lesson)
  - negative constraint (as part of subject verb agreement)

Various Types of Constraints

- Defining equations:

  Contribute a value for the specified attribute

  Notation: `(^ ATTRIBUTE) = value`

- Constraining equations:

  Check whether the specified attribute has the specified value, but do **not** contribute/introduce that value

  Notation: `(^ ATTRIBUTE) =c value`

  Example: Check in the Passive Lexical Rule that the form of the verb is indeed a past participle.

Various Kinds of Constraints (cont'd)

- Negated constraints:

  Enforce that the specified attribute does **not** have the specified value.

  Notation: `(^ ATTRIBUTE) ~= value`

  Example: Base-form entry of English verb may state that `(^ SUBJ PERS) ~= 3` if `(^ SUBJ NUM) = sg`.

- Existential constraints:

  Enforce that the specified attribute has some value, without specifying which value.

  Notation: `(^ ATTRIBUTE)`

  Example: Singular entry of English count noun may state that `(^ DEF)`.

Example of a constraining equation

- Passive Lexical Rule

```
PASS(FRAME) = { FRAME   "base case"
              | FRAME    "passive"
                (^ PASSIVE) = +
                (^ PARTICIPLE) =c past
       "make sure to have a past participle"
                (^ OBJ)-->(^ SUBJ)
                { (^ SUBJ)-->(^ OBL-AG)
              | (^ SUBJ)--> NULL }
            }.
```

Example of a negated constraint

- The gorillas devour the bananas.

```
devour V * PRED='devour<(^SUBJ)(^OBJ)>'
           (^TENSE) = pres
           (^MOOD) = indicative
           { (^SUBJ NUM) = pl
           | (^SUBJ NUM) = sg
             (^SUBJ PERS) ~= 3
           }.
```

Example of an existential constraint

- The gorilla ate *(the) banana.

```
banana V * PRED='banana'
           (^ NUM) = sg
           (^ DEF).
```

Practical Work

- This concludes Lesson 3.
- The practical work you should do now is detailed in Exercise 3.
- You will practice with
  - templates
  - lexical rules
  - feature unification and different types of constraints
  - testsuites