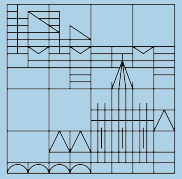
Universität
Konstanz



Grammar Development with LFG and XLE

Miriam Butt
University of Konstanz

Last Time

- Integration of OT-Marks
  - Parsing
  - Generation
- Pronouns

This Time: Lesson 6

1. Imperatives
   - empty nodes (e)
   - The LFG/XLE take on “Constructions”
2. Coordination
   - Regular Expression Macros
   - Metarulemacros
3. (Proper Names)

Non-Overt Arguments

- Languages allow arguments to be non-overt.
- For example, in the imperative, the addressee subject is usually omitted.
- But many languages also allow arguments to be omitted in other situations.
- This phenomenon is known as *pro-drop*.
- Several of the ParGram grammars have implemented *pro-drop* (Japanese, Urdu).
- Gives rise to massive ambiguities so non-trivial, but possible.

Null Argument in Imperatives

- LFG does not posit empty categories.
- However, XLE allows for an empty node "e".
- This node is useful if you need a place to put information in the c-structure but have no good node to place it on.
- Example: the null argument in imperatives.
  - » Take the exam!
- General ParGram strategy:
  - Introduce a separate category (construction): **Simp**
  - Have this contain an empty category with the relevant information.

Null Arguments

- Example for imperative:

```
Simp --> e: (^ SUBJ PRED) = 'pro'
            (^ SUBJ PERS) = 2;
         VP
         EXCL.
```

- The EXCL stands for “exclamation mark” and integrates this punctuation mark into the grammar.
- The e denotes an empty node. This does not show up in the c-structure.

ROOT vs. S

- So far we have had “S” be the top category in the grammar.
- But languages tend to contain several different types of sentences:
  - declaratives
  - imperatives
  - questions
  - ...
- Solution (ParGram): change the root category from S to ROOT in the configuration section of the grammar.

ROOT vs. S

- Let ROOT expand to different sentence types.
  - S (normal declarative sentence)
  - Simp (imperative)
  - Sint (interrogative)
- In a sense, this approach implements the idea of “Constructions” as propagated by Construction Grammar (CG), for example.
- However, these constructions have no theoretical status in LFG (unlike in CG).
- They are an engineering solution developed within XLE.

Constructions at c-structure

- An alternative solution would be to have just a single S (or CP or IP, depending on the theoretical orientation).
- The different types of S would then be encoded at the functional level (f-str).
- But in grammar writing one should always keep the following in mind:
  - c-structure is about context-free rules and hence computationally “cheap”.
  - f-structure is context sensitive. It involves constraint checking and feature unification and is computationally expensive.

Demo

grammar5.lfg
testsuite5.lfg

imperative (implement)
empty category e
ROOT category
(regeneration)

Coordination

- Recall: every attribute can only have one value.
- So what do we do with coordinated constituents?

Example: gorillas [climb trees] and [eat bananas]

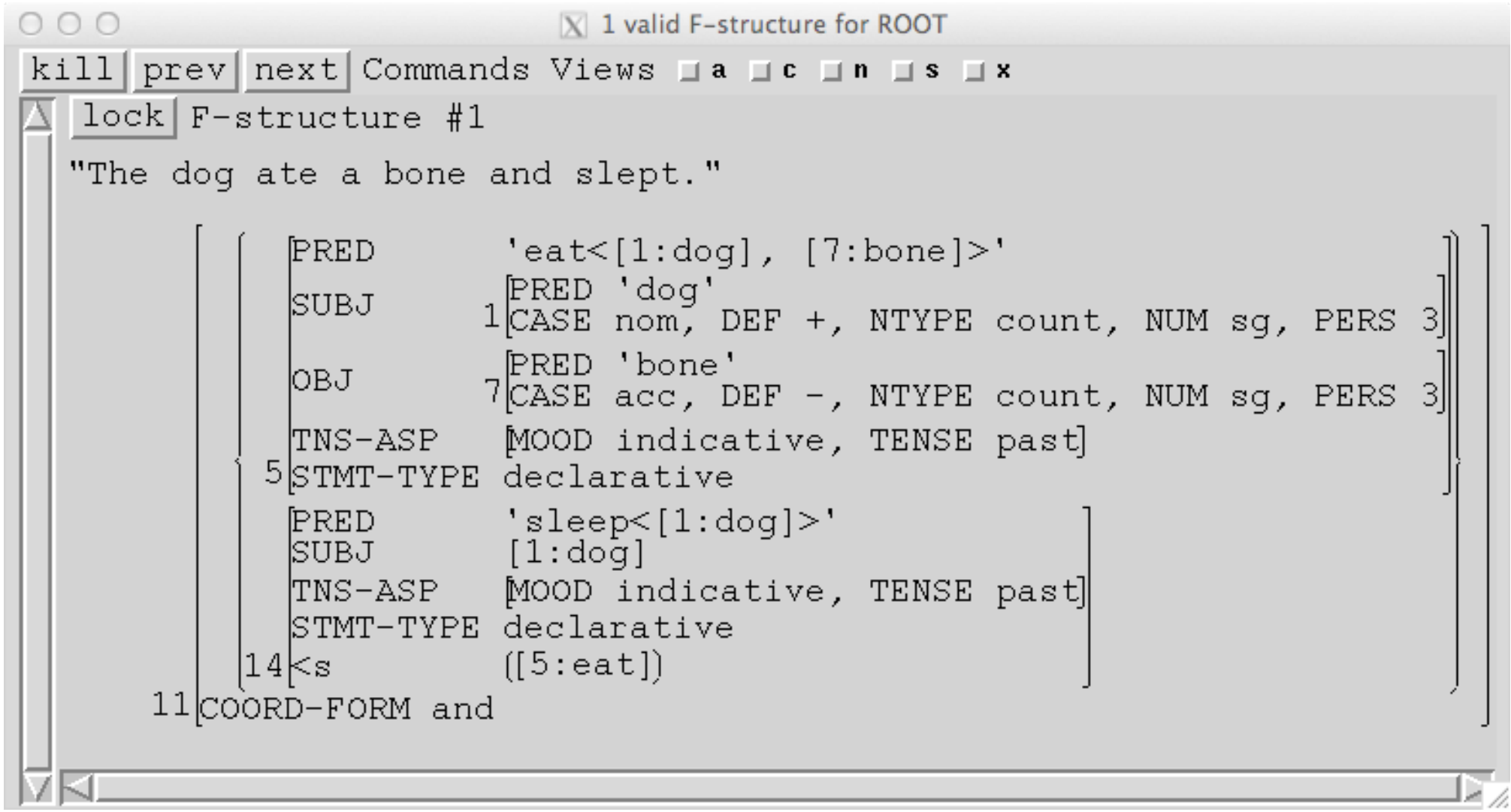
```
VP --> { …
       | VP: ! $ ^
         CONJ
         VP: ! $ ^
       }.
```

- Answer: put them into a set ($\downarrow \in \uparrow$)

Coordination – Sets

- Advantage of sets:
  - can have multiple instances
  - no feature clash
- Disadvantage:
  - Coordinated items are in an unstructured “bag”.
  - Do not know which came first linearly unless one looks back at the c-structure.
  - This can become important for calculating **scope relations.**
- Solution:
  - register the linear order (scope) at f-str via <s

Coordination – Example



Coordination

Coordination can happen at any level of c-str.

Example: the gorillas [peel and eat] the bananas

```
V --> { …
      | V: ! $ ^
        CONJ
        V: ! $ ^
      }.
```

Coordination

- Basically every category can be coordinated.
- Known as **Same Category Coordination.**

Example: the gorillas eat the bananas [in the cage and in the garden]

```
PP --> { …
       | PP: ! $ ^
         CONJ
         PP: ! $ ^
       }.
```

Coordination

How can we capture these generalizations?

Via regular-expression macros!

```
SCCOORD(CAT) = CAT: ! $ ^;
                 CONJ
                 CAT: ! $ ^.
PP --> { ...
       | @(SCCOORD PP)
       }.
```

Nominal coordination

- NP, N, etc. coordination is special.
- The NUM attribute should typically have the value pl.
- Even when the individual set members are singular.

Mary **likes** bananas.

Mary and the gorilla **like** bananas.

*Mary and the gorilla **likes** bananas.

The boys and girls like bananas.

Nondistributives

- In the configuration section of the grammar NONDISTRIBUTIVES are specified.
- Recall that the SUBJ was distributed over both conjuncts in our example.
- In grammar5.lfg, NUM, PERS are specified as being nondistributives.
- The values of these attributes are **not** distributed across each conjunct – every conjunct can have an individual value.

Mary and I **like** bananas.

Nominal coordination

```
NPCOORD(CAT) = CAT: ! $ ^;
                    CONJ: ^ = !
                                (^ NUM) = pl;
                    CAT: ! $ ^.
NP --> { ...
       | @(NPCOORD NP)
       }.
N --> { ...
      | @(NPCOORD N)
      }.
```

Nominal coordination

```
NP-CONJUNCT = "person resolution"
      { "if either conjunct is 1st person; the NP is"
        "EX: the boys and me}"
        (! PERS)=c 1
        (^ PERS)=1
       |"if a conjunct is 2nd person and the NP is not
           already 1st person, make it 2nd person"
        (! PERS)=c 2
        { (^ PERS)=c 1 "one conjunct was 1st person"
                       "EX: you and I}"
          |(^ PERS)=2 } "else assign 2nd person"
                       "EX: you and the boys}"
         |"else 3rd person, Ex: the boys and her}"
          (^ PERS)=3}.
```

METARULEMACRO

- Macros are a useful way of stating generalizations across types of rules.
- But, it is tedious to amend almost all rules so that either the SCCOORD or the NPCOORD macro are invoked (e.g., PPs, NPs, VPs, Vs, ...).
- XLE therefore implemented a special macro called the **METARULEMACRO**.
- Every rule goes through the METARULEMACRO unless specified otherwise.
- It encodes a meta statement about the entire grammar.

METARULEMACRO

- Takes three arguments: _CAT, _BASECAT, and _RHS
- _CAT is the category on the left-hand side of the rule
- _BASECAT is the same as _CAT unless you are dealing with a complex-category rule
- _RHS is the right-hand side of the rule

METARULEMACRO

```
METARULEMACRO(_CAT _BASECAT _RHS)=

    { _RHS
    | e: _CAT $ { N NP };
      @(NPCOORD _CAT)
    | e: _CAT ~$ { N NP };
      @(SCCOORD _CAT)
    }.
```

Demo

grammar-coord.lfg
testsuite-coord.lfg

coordination

Practical Work

- This concludes Lesson 6.
- The practical work you should do now is detailed in Exercise 6.
- You will practice with
  - imperatives (empty categories)
  - coordination (metarulemacro)
  - proper nouns

More on NP-CONJUNCT

- The NP-CONJUNCT template reflects crosslinguistic generalizations.
- However, not all languages are the same.
- The person resolution can generally be determined via verb agreement.
- The next example is from Spanish – only first person plural is acceptable.

José y yo hablamos/*habláis/$hablan.

Jose and I speak.1.Pl/2.Pl/3.Pl

‘Jose and I speak.’