Lexical-Functional Grammar (LFG)

L614
Spring 2010

Based heavily upon Dalrymple (2001) and to a lesser degree on Austin (2001)

Motivation for LFG

- **Lexical** = (not transformational) richly structured lexicon, where relations between, e.g., verbal alternations, are stated
- **Functional** = (not configurational) abstract grammatical functions like subject and object are primitives, i.e., not defined by the phrase structure configurations

LFG in a nutshell

LFG (minimally) distinguishes two kinds of representation:

- **c-structure** (constituent structure):
  overt linear and hierarchical organization of words into phrases
- **f-structure** (functional structure):
  abstract functional organization of the sentence, explicitly representing syntactic predicate-argument structure and functional relations

These are two completely different formalisms: trees (c-structure) and attribute-value matrices (f-structure)

(We will largely ignore A-structure and σ-structure here.)

Part I: F-structure

F-structure maps more closely to meaning and encodes abstract grammatical relations like subject and object as primitives, i.e. not reducible to anything else (e.g., tree structure)

Motivation:

- Study of grammatical relations predates modern linguistic theory
- Categories like subject and object are cross-linguistic → languages vary less in their f-structure
- e.g., Keenan-Comrie Hierarchy (for relative clause formation) is supposedly universal

\[\text{subj} > \text{do} > \text{io} > \text{obl} > \text{gen} > \text{ocomp}\]

Grammatical functions

Inventory: **subject**, **object**, **obl**, **comp**, **xcomp**, **oblique**, **adjunct**, **xadjunct**

- Terms (core functions): **subj**, **obj**, **obl**
- Semantically restricted: **obl**, **obl**
  - Thematic restrictions (θ) placed on function
  - **obl**: secondary **obj** functions associated with thematic roles: **obl**
  - **obl**: thematically restricted oblique functions, often corresponding to adpositions
- Open clausal functions (no internal subject): **xcomp**, **xadj**
  - **comp**: sentential or closed (nonpredicative) infinitival complement
  - **xcomp**: open (predicative) complement with subject externally controlled

Governable & non-governable grammatical functions

- **Governable functions**: **subj**, **obj**, **obl**, **comp**, **xcomp**, **obl**
  - A predicate can govern these functions (i.e., subcategorize for them)
- **Non-governable functions**: **adj**, **xadj**
  - **adj**: David devoured a sandwich yesterday.
  - **xadj**: Having opened the window, David took a deep breath.
Subcategorization

Subcategorization is done at f-structure
• Verbs select for grammatical functions
• Use the pred (predicate) feature to specify the semantic form, e.g.,
  - yawn: pred 'yawn<subj>'
  - hit: pred 'hit<subj,obj>'
  - give: pred 'give<subj,obj,objtheme>'
  - eat: pred 'eat<subj,(obj)>'

F-structure representation: Simple F-structures

F-structure is a function from attributes to values
• For the proper noun David, pred and num are attributes; 'DAVID' and sg are the corresponding values
  (1) pred 'DAVID'
      num sg
• F-structures within f-structures: David yawned
  (2) pred 'yawn<subj>'
      tense past subj
      pred 'David'
      num sg

Semantic forms

Semantic forms are actually uniquely instantiated, so the previous f-structure can look like:
(3) pred 'yawn<subj>'
    tense past subj
    pred 'David'
    num sg

This makes it more clear that each word makes a unique contribution to the f-structure.
• Generally, not a crucial issue, so we leave the indices off most f-structures

F-structure features

What sorts of features can be used?
• Ultimately, that’s up to the grammar writer
• Commonly used features in LFG include ASPECT, PRONTYPE, VFORM, etc. (see (17) in Dalrymple (2006))

Important note:
• LFG does not define a set of features or values which must be included in an f-structure
• So, one verb may define vform, while another might leave it undefined.
  – This is different from HPSG, as we’ll see.

F-structure representation: Sets

Values can be sets, in order to handle phenomena with an unbounded number of elements, e.g. adjuncts, coordinates
David yawned quietly yesterday.
(4) pred 'yawn<subj>'
    tense past subj
    pred 'David'
    num sg
    adj
    pred 'quietly'
    pred 'yesterday'

David and Chris yawned.
(5) pred 'yawn<subj>'
    tense past subj
    num pl
    pred 'David'
    num sg
    pred 'Chris'
    num sg

Sets can also have additional properties, i.e. have attributes and values which apply over the whole set—hybrid object
• Or, properties can distribute over elements of the set (e.g., num feature below)
David and Chris yawned.
**F-structure representation: Attributes with Common Values**

Attributes can share the same values, to describe phenomena such as raising; notated in different ways, e.g., for David seemed to yawn:

More traditional notation:  
\[
\begin{align*}
\text{PRED} & \quad \text{SEEM} < \text{XCOMP} > \text{SUBJ} \\
\text{TENSE} & \quad \text{PAST} \\
\text{SUBJ} & \quad \text{PRED} < \text{DAVID} > \\
\text{XCOMP} & \quad \text{PRED} < \text{YAWN} < \text{SUBJ} > \\
\end{align*}
\]

More HPSG-like notation:  
\[
\begin{align*}
\text{PRED} & \quad \text{SEEM} < \text{XCOMP} > \text{SUBJ} \\
\text{TENSE} & \quad \text{PAST} \\
\text{SUBJ} & \quad \text{PRED} < \text{DAVID} > \\
\text{XCOMP} & \quad \text{PRED} < \text{YAWN} < \text{SUBJ} > \\
\end{align*}
\]

---

**The nature of f-structure**

An f-structure is restricted by the principles of

- **completeness**: a predicate and all its arguments be a part of the structure
- **coherence**: all arguments in the structure must be required by a predicate
- **uniqueness (consistency)**: every attribute has a single value

---

### Completeness

- Argument list of a semantic form = list of governable grammatical functions
- Completeness: All governable grammatical functions mentioned in the predicate must be present in the f-structure.

(7) a. pred 'DEVOUR<OBJ,OBJ>'

b. *David devoured.

**Definition**: An f-structure is **locally complete** iff it contains all the governable grammatical functions that its predicate governs. An f-structure is **complete** iff it and all its subsidiary f-structures are locally complete.

---

### Coherence

- Coherence: All governable grammatical functions present in the f-structure must be mentioned in the argument list of the predicate.
- Like completeness, but in the other direction.

(8) a. *David yawned the sink.

b. David yawned the sink.

**Definition**: An f-structure is **locally coherent** iff all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is **coherent** iff it and all its subsidiary f-structures are locally coherent.

---

### Uniqueness (consistency)

- Avoid conflicting values, e.g. plural noun and singular verb

(9) a. *The boys yawns.

b. [ PRED 'YAWN<SUBJ>' SUBJ 'BOYS' NUM SG/PL ]

**Definition**: In a given f-structure, a particular attribute may have at most one value.

---

### Constraining f-structures

We use functional equations (defining equations) on words and phrases to describe acceptable f-structures.

F-description with a single equation:

(10) \((g \text{ NUM}) = \text{ SG}\)

Different f-structures which satisfy this f-description:

(11) a. [ NUM SG ]

b. [ PRED 'CHARLIE' GEND MASC NUM SG ]
Functional Constraints (formally)

(12) \((fa) = v\) holds iff \(f\) is an f-structure, \(a\) is a symbol, and the pair \(\langle a, v \rangle \in f\)

\(\Rightarrow\) The f-structure for an utterance is the minimal solution satisfying the constraints introduced by the words and phrase structure of the utterance.

minimal solution: satisfies all constraints in the f-description and has no additional structure

Constraining equations

Can also use constraining equations to check the properties of the minimal solution

- e.g., the \(\text{SUBJ}\) of \(f\) must meet certain conditions: \((f \text{ SUBJ NUM}) = _e \text{ SG}\)

Defining equations and constraining equations are formally very similar, so we will treat them in the same way

Functional constraints example

Lexical constraints:

- **John**
  - (g pred) = 'John'
  - (g num) = SG
- **runs**
  - (f pred) = 'run<subj>'
  - (f subj case) = NOM
  - (f subj num) = SG

Phrasal constraints (more on this later):

- (f subj) = g

Combining lexical and phrasal constraints, we have:

- (f subj) = g
- (g pred) = 'John'
- (g num) = SG
- (f pred) = 'run<subj>'
- (g case) = NOM
- (g num) = SG

Minimal solution:

\[
\begin{bmatrix}
\text{PRED} & 'run<subj>' \\
\text{SUBJ} & g \\
\text{CASE} & \text{NOM} \\
\text{NUM} & \text{SG}
\end{bmatrix}
\]

More functional constraints

We want more ways to define the set of acceptable f-structures

- Disjunction
- Negation
- Existential Constraints
- Optionality

Disjunction

Disjunction: different options can be used to satisfy an f-description

(13) I met/have met him.

- Lexical entry for met:
  - (f pred) = 'meet<subj,obj>'
  - ((f tense) = past | (f form) = pastpart}
**Negation**

Negation: an f-description is specified that cannot be true

(14) a. I know whether/if David yawned.
    b. You have to justify whether/*if your journey is really necessary.

⇒ if is not allowed with justify (unlike know)

- justify \( \mathbf{V} (f \text{comp compform}) \neq \text{if} \)

**Existential Constraints**

Existential constraint: an f-structure must have some attribute, but the value of that attribute is unconstrained.

(15) a. The man who yawns/yawned/will yawn.

⇒ In a relative clause, yawn must be tensed, but which tense is not important

- Relative clause constraint is simply: \( (f \text{tense}) \)

Can also specify negative existential constraints, e.g., \( \neg (f \text{tense}) \)

**Optionality**

Optionality: an f-description may but doesn’t need to be satisfied

(16) a. Juan vió a Pedro.
    b. Juan lo vió.
    c. Juan lo vió a Pedro.

⇒ If the semantic information contributed by lo is optional, that explains how both Pedro and lo can appear in the same sentence.

- Pedro \( N (f \text{pred}) = \text{Pedro} \)
- lo \( \text{Pro} ((f \text{pred}) = \text{pro}) \)

**Part II: C-structure**

Having examined f-structure, we can now turn to c-structure

C-structure corresponds to a fairly traditional notion of phrase structure

- X-Bar Theory: lexical heads with specifier and complements
- Adjunction: another permissible configuration
- Categories: lexical (N, P, V, A, Adv) and functional (I, C) categories—not universally fixed

Slightly different notions:

- Endocentric category S: has no lexical head (for “internal subject” languages)
- Optionality: all constituent structure positions are optional

**Example of c-structure**

(17) kogda rodilsja Lermontov?
    "When was Lermontov born?"

⇒ Optionality: Specifier of IP is not in tree, and VP has no V head.
C-structure rules

Like phrase structure rules, but

• interpreted as node admissibility conditions, i.e., trees must meet PSR descriptions
• allow for regular expressions (Kleene star, disjunction, optionality, etc.) on the right-hand side.

We can also employ the use of

• Metacategories
• ID/LP rules

Metacategories

A metacategory represents several different sets of categories

(18) a. XP ≡ \{NP | PP | VP | AP | AdvP\}
    b. VP ≡ V NP

Note that using the metacategory VP given in (18b) in the rule S → NP VP results in the following tree:

```
   S
   / \  /
NP  V  NP
```

ID/LP Rules

Rules can be written in ID/LP format: ID = immediate dominance, LP = linear precedence

(19) No LP rules:
    a. VP → V, NP
    b. VP → \{V NP | NP V\}

(20) One LP rule:
    a. VP → V, NP
    b. VP → V NP

(21) Interacting LP rules:
    a. VP → V, NP, PP V < NP, V < PP
    b. VP → \{V NP PP | V PP NP\}

The Head Convention

Multiple c-structures can map onto the same f-structure → this allows nodes to inherit properties from their head

```
VP
  / \  /
V  V'
```

\[\phi(V) = \phi(V') = \phi(V)\]

```
Yawned
```

F-structure/C-structure Regularities

Can have set mappings for particular positions, e.g., the specifier of IP in English maps to SUBJ (the same position in Russian maps to TOPIC and in Bulgarian to FOCUS)

```
  IP
  / \  /
NP  V  VP
  / \  /
N  V  David
```

\[\phi(IP) = \phi(NP) = \phi(VP)\]
**Alternate Notation**

A way to specify this constraint on the specifier of IP is the following:

\[
(22) \text{IP} \rightarrow \text{XP} \quad \text{I}'
\]

\[↑\text{subj} = ↓\text{I}' \quad ↑=↓ \]

- This says: The value of subj for XP’s mother is equal to XP’s f-structure
- IP and I’ have the same f-structure

**Annotated Phrase Structure Rules**

\[ \text{(23) V'} \rightarrow \text{V} \text{ NP} \quad ↑ = ↓ \text{NP} (↑\text{obj}) = ↓ \]

\[ \text{(24) VP} \rightarrow \text{V} \text{ NP} \quad ↑ = ↓ \text{NP} (↑\text{obj}) = ↓ \text{NP} (↑\text{obj2}) = ↓ \]

\[ \text{(25) VP} \rightarrow \text{V} \text{ NP} \quad ↑ = ↓ \text{NP} (↑\text{obj2}) = ↓ \text{PP} (↑\text{obj}) = ↓ (↑\text{pform}) = \text{to} \]

**Lexical Entries**

Can use the same notation to express lexical entries

\[ (26) \text{a. yawned} \quad \text{V} \quad (↑\text{pred}) = \text{'yawn<subj>}' \]

\[ (↑\text{tense}) = \text{past} \]

\[ \text{b. David} \quad \text{N} \quad (↑\text{pred}) = \text{'David'} \]

The setup is best illustrated with an example or two . . .

For example, how do we get the final f-structure for David yawned?

**An example grammar: The c-structure rules with annotations**

(based on Kaplan and Bresnan 1995)

\[ (27) \text{a. S} \rightarrow \text{NP} \quad \text{VP} \quad (↑\text{subj}) = ↓ \text{VP} (↑\text{obj}) = ↓ \text{NP} (↑\text{obj2}) = ↓ \text{NP} (↑\text{obj2}) = ↓ \]

\[ \text{b. NP} \rightarrow \text{Det} \quad (↑\text{spec}) = \text{the} \quad (↑\text{num}) = \text{sg} \quad (↑\text{pred}) = \text{'toy'} \]

\[ \text{f. toy} \quad \text{N} \quad (↑\text{num}) = \text{sg} \quad (↑\text{pred}) = \text{'toy'} \]

**A sentence licensed by the example grammar**

\[ f_1: \text{S} \]

\[ f_3: \text{VP} \quad (↑\text{subj}) = \text{Det} \quad (↑\text{obj}) = \text{N} \quad (↑\text{pform}) = \text{to} \]

\[ f_5: \text{NP} \quad (↑\text{spec}) = \text{A} \quad (↑\text{num}) = \text{sg} \quad (↑\text{pred}) = \text{'baby'} \]

\[ f_4: \text{NP} \quad (↑\text{spec}) = \text{A} \quad (↑\text{num}) = \text{sg} \quad (↑\text{pred}) = \text{'baby'} \]
Part IV: A few notes

We’ll finish up the unit on LFG by looking quickly at:

- Extraction
- The syntax-semantics interface
- Computational issues

Extraction: Functional uncertainty

The way extraction is handled in LFG is by functional uncertainty: a functional equation sets up a relation between some initial, extracted object with a grammatical function (GF) later in the sentence. Which GF is left unspecified, e.g.:

(29) CP → XP
    (↑focus) = ↓
    (↑focus) = (↑comp* GF)

This says that the focus element is equated with some GF after a path of comp values.

Extraction example

(30) What do you think Chris bought?

The principle of completeness ensures that bought has a realized object, and the functional equation fills it in.

Extraction example 2

(32) What do you think Chris hoped David bought?

The syntax-semantics interface

F-structures are fairly closely linked to the semantics of a sentence, but the details still have to be worked out:

One theory of the syntax-semantics interface is glue semantics:

- A compositional form of semantics, which maps from the f-structure to a semantic formula
  - The σ function performs the mapping
- Employs linear logic
  - Each premise (i.e., word) must be used once and can only be used once
  - It is thus resource-driven, requiring every word to contribute to the meaning
Computational issues

Processing c-structure by itself is essentially equivalent to processing CFGs, which is very efficient.

How does one account for f-structures?

- Can be interleaved
- Can post-process c-structures with f-structure constraints

It has been shown that if an f-structure is acyclic, the set of strings it corresponds to are equivalent to a context-free language.

- This can help constrain both parsing and generation

Summary

- LFG is split into f-structure and c-structure, with a mapping between them
- F-structure is a rich feature-based way of encoding functional relations
- C-structure is a basic constituent structure

References

