Lexical-Functional Grammar
Architecture

Weiwei Sun

Institute of Computer Science and Technology
Peking University

September 15, 2018
**Homework:** 读书报告

挑一篇阅读材料，写一篇内容总结（中文），要求：

- 在理解的基础上概述阅读材料的最主要内容
- 不超过2000字（不包括图、公式等）
- 3月26日前电邮PDF版发送给我

**Reference**

- Chapter 1 of *Lexical-Functional Syntax*: Motivation for the LFG Architecture.
Word groups or word shapes (1)

(1) a. *The two small are chasing that children dog.
   b. *The two small are dog chasing children that.
   c. *Chasing are the two small that dog children.
   d. *That are children chasing the two small dog.
Word groups or word shapes (2)

Every permutation of the words in the sentence is possible, so long as the auxiliary tense marker occurs in the second position.

Bresnan (1998, 2001)
Morphology competes with syntax.

Absolutive: Morphological case in ergative languages for indicating subject of intransitive verbs and object of transitive verbs. Ergative: Morphological case in ergative languages for indicating agent of the transitive verbs in the basic voice.
Grammatical relations

Although Warlpiri lacks English-style phrase structure, and English lacks Warlpiri-style case and agreement forms of words, they have a common organization at a deeper level.

Binding: Subject/Object matters

(2) a. Lucy is hitting herself.
   b. *Herself is hitting Lucy.

    Napaljarri-ERG PRES-REFL hit-NONPAST.
    ‘Napaljarri is hitting herself.’
   b. *Napaljarri ka-nyanu paka-rni.
      Napaljarri.ABS PRES-REFL hit-NONPAST.
      ‘Herself is hitting Napaljarri.’
Do you like the following representation?

S

Subject  Aux  Predicate

NP  M  VP

N  may  Main  Verb  Object

sincerity  V  NP

frighten  Det  N

the  boy
Do you like the following representation?

```
S
  |--- Subject  Aux  Predicate
  |    NP       M      VP
  |      N      may    V
  sincerity    Main Verb    NP
  frighten Det N
   the boy
```
F-structure: motivation

Assumption

For any language functional syntactic concepts such as subject and object are relevant.

⇒

Use f(unctional)-structure

▶ to represent what languages have in common in wide-spread phenomena,
▶ to capture some universal properties of language

There is no advantage in representing such information as phrase-structure information.
Discussion

How should abstract grammatical relations be captured?

- **Transformational Grammar**: configurationally, using a uniform syntactic representation
- **LFG**: nonconfigurationally, using a separate syntactic representation
Discussion

How should abstract grammatical relations be captured?

- **Transformational Grammar**: configurationally, using a uniform syntactic representation
- **LFG**: nonconfigurally, using a separate syntactic representation

What does this representation mean?

(5) sincerity may frighten the boy.

```
PRED '我〈SUBJ,OBJ〉'

SUBJ PRED 'sincerity'

OBJ [PRED 'boy'

DEF +
PERS 3rd
NUM sg]
```
Lexical-Functional Grammar

The idea that words and phrases are alternative means of expressing the same grammatical relations underlies the design of Lexical-Functional Grammar (LFG).

Pioneers

Developed in the late 70s by Joan Bresnan and Ron Kaplan

- J. Bresnan: A linguist trained at the MIT
- R. Kaplan: A psycholinguist and computational linguist trained at Harvard
Demo

http://clarino.uib.no/iness/xle-web
Outline

Grammatical Functions

Syntactic Descriptions

Structural Correspondences

Configurational vs. Nonconfigurational
Grammatical functions

Universally-available grammatical functions

LFG posits a *universal* inventory of grammatical functions.

- **SUBJ**: subject
- **OBJ**: object
- **COMP**: sentential or *closed* (nonpredicative) infinitival complement.
- **XCOMP**: an *open* (predicative) complement, often infinitival, whose **SUBJ** function is externally controlled.
- **OBJθ**: a family of secondary **OBJ** functions associated with a particular, language-specific set of thematic roles
- **OBLθ**: a family of thematically restricted *oblique functions*. E.g. **OBLGOAL**, **OBSOURCE**
- **ADJ**, **XADJ**: *adjunct functions*
**Subject (1)**

**Case**

(6) he/*him broke the window

**Agreement**

(7) I am / You are / He is

**Subjecthood test**

How about Mandarin Chinese?
Object

Many languages have more than one phrase bearing an object function.

(8) a. He gave her a book.
    b. I bet you 1 million pounds you won’t click on this video. (from YouTube)

Languages allow a single thematically unrestricted object, the primary OBJ.

(9) a. I gave her a book.
    b. I gave a book to her.

Languages may allow one or more secondary, thematically restricted objects, viz OBJθ. In English, the thematically restricted object must be a theme.

(10) a. I made her a cake.
    b. *I made a cake her.
Oblique

- Obliques are with an explicit indication of the thematic role. In English, this indication is by means of prepositions: oblique arguments are PPs, while objects are bare NPs/DPs.

- The oblique argument functions include such grammatical functions as GOAL (to), BENEFACTIVE (for), SOURCE (from), INSTRUMENT (with), LOCATION (various prepositions), AGENT (by), etc.

- $\text{OBL}_\theta$: $\text{OBL}_{\text{GOAL}}$, $\text{OBL}_{\text{BEN}}$, $\text{OBL}_{\text{SOURCE}}$, $\text{OBL}_{\text{INSTRUMENT}}$, $\text{OBL}_{\text{LOC}}$, $\text{OBL}_{\text{AGENT}}$

(11) a. David gave the book $[[\text{OBL}_{\text{GOAL}} \text{ to Chris}]]$.
   b. *David gave the book $\text{to Chris to Ken}$.
The **COMP**, **XCOMP**, and **XADJ** functions are **clausal functions**.

- The **COMP** function is a closed function **containing an internal SUBJ phrase**.

  (12) a. David complained that **Chris** yawned.
      b. David wondered **who** yawned.
      c. David couldn’t believe **how big** the house **was**.

- The **XCOMP** and **XADJ** functions are open functions that **do not contain an internal subject phrase**. Their **SUBJ** must be specified externally to their phrase.

  (13) a. **David** seemed to **yawn**.
      b. Chris expected **David** to **yawn**.

  (14) **Stretching his arms**, **David** yawned.
Cross-classification of grammatical functions

Several cross-classifications are possible among grammatical functions.

- Governable functions: SUBJ, OBJ, (X)COMP, OBJ_{θ}, OBL_{θ} are governed or subcategorized for by the predicate.
- Modifiers: ADJ, XADJ modify the phrase they appear in, but they are not subcategorized for by the predicate.

- Open functions: XCOMP, XADJ
- Closed functions: SUBJ, OBJ, COMP, OBJ_{θ}, OBL_{θ}, ADJ
Classification of the argument functions

Core vs. Non-core

- **Core** arguments/terms: SUBJ, OBJ, OBJ\(\theta\)
  - More strictly grammatical functions
- **Non-core** functions/non-terms: OBL\(\theta\)
  - Oblique elements are much less active syntactically than core elements.

Restricted vs. Unrestricted

- Semantically **unrestricted** functions: SUBJ, OBJ
- Semantically **restricted** functions: OBJ\(\theta\), OBL\(\theta\)
Nongovernable grammatical functions

There are two nongovernable grammatical functions:

- **ADJ**: grammatical function of modifiers
- **XADJ**: open predicative adjuncts, whose **SUBJ** function is externally controlled.

(15) a. Having opened the window, David took a deep breath.
    b. David ate the celery **naked**.
    c. David ate the celery **raw**.

More than one adjunct function can appear in a sentence.

(16) David devoured a sandwich **at noon yesterday**.
The autonomy of functional organization

Key: GFs are primitive concepts

LFG does not assume that abstract grammatical functions are defined in terms of their phrase structural position in the sentence nor in terms of morphological properties like casemarking.

Comparison to GB

Grammatical relations are defined structurally, in terms of the tree.

- **Subject**: NP or CP daughter of TP
- **Object**: NP or CP daughter of a VP headed by a transitive verb
- **Object of preposition**: NP daughter of PP
### Examples of English grammatical functions in LFG
*(based on Asudeh and Toivonen 2015)*

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>Some people with no shame walked in and wrecked the party.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SUBJ)</td>
<td>The party was wrecked by some people with no shame.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT</th>
<th>Primary object</th>
</tr>
</thead>
<tbody>
<tr>
<td>(OBJ)</td>
<td>Ricky trashed the hotel room.</td>
</tr>
<tr>
<td></td>
<td>Ricky gave John a glass.</td>
</tr>
<tr>
<td></td>
<td>Ricky gave a glass to John.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT_θ</th>
<th>Secondary object; thematically restricted object</th>
</tr>
</thead>
<tbody>
<tr>
<td>(OBJ_θ)</td>
<td>(object theme, restricted to theme roles)</td>
</tr>
<tr>
<td></td>
<td>(obj_θ in English restricted to theme, cannot be beneficiary)</td>
</tr>
<tr>
<td></td>
<td>Sandy gave John a glass.</td>
</tr>
<tr>
<td></td>
<td>Tom baked Susan a cake.</td>
</tr>
</tbody>
</table>
### Examples of English grammatical functions in LFG  
*(based on Asudeh and Toivonen 2015)*

<table>
<thead>
<tr>
<th>Grammatical Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| OBLIQUE<sub>θ</sub> | Typically has oblique case or is a PP                                       | Julia placed the vase *on the desk.*  
Ricky gave a glass *to John.* |
| COMPLEMENT          | Closed (saturated) complement: a clausal argument which has its own SUBJECT | Peggy told Matt *that she had won the prize.*                          |
| XCOMP               | Open (unsaturated) predicate complement                                       | I told Patrick *to quit.*                                               |
|                     |                                                                            | Peggy-Sue seems *to be a complete fraud.*                              |
| ADJUNCT<sub>ADJ</sub> | A modifier, a nonargument                                                   | Mary read a *good book.*                                                |
|                     |                                                                            | Mary counted the cars *very quickly.*                                  |
|                     |                                                                            | Sally killed a bug *in the yard.*                                     |
|                     |                                                                            | *Since she had no money,* Mary was forced to get a job.*               |
### Examples of English grammatical functions in LFG
*(based on Asudeh and Toivonen 2015)*

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XADJ</strong></td>
<td>Open predicate adjunct</td>
<td><strong>Having no money</strong>, Mary was forced to get a job.</td>
</tr>
<tr>
<td><strong>POSS</strong></td>
<td>Possessor phrase</td>
<td>John’s book</td>
</tr>
</tbody>
</table>
| **TOP**  | Grammaticalized discourse function; must be identified with or anaphorically linked to another grammatical function | Bagels, Mary loves. \(\text{TOP} = \text{OBJ}\)  
As for bagels, Mary loves them. \(\text{TOP} \text{ anaphorically linked to OBJ}\) |
| **FOC**  | Grammaticalized discourse function; must be identified with or anaphorically linked to another grammatical function | Which author do the critics praise? \(\text{FOC} = \text{OBJ}\) |
Outline

Grammatical Functions

Syntactic Descriptions

Structural Correspondences

Configurational vs. Nonconfigurational
Multiple parallel structures

**Assumption:** language is made up of multiple dimensions of structure.
- LFG describes and models language by parallel structures.
- LFG also illustrates how different aspects of linguistic structure are related.

**Traditional LFG analyses focus on two structures**

Constituent/categorial structure (**c-structure**)
- *overt*, more concrete level of linear and hierarchical organization of words into phrases.

Functional structure (**f-structure**)
- *abstract functional syntactic* organization of the sentence
- syntactic predicate-argument structure and functional relations like *subject* and *object*. 
C-structure

Described by conventional PS trees

PS trees are defined in terms of syntactic categories, terminal nodes, dominance and precedence.

▶ They don’t contain unpronounced words.
▶ They reflect the structure and grouping of words and phrases in the sentence.

Not universal

▶ Languages are very different on the c-structure level.
▶ The inventory of phrasal categories is not universally fixed, but may vary from language to language.
Phrase structure rules

**CFG rules**

\[ S \rightarrow \text{NP} \text{ VP} \]

- This rule permits a node labeled \( S \) to dominate two nodes, an NP and a VP, with the NP preceding the VP.
- In LFG, phrase structure rules are interpreted as node admissibility conditions: *a phrase structure tree is admitted by a set of phrase structure rules.*

**LFG employ more flexible phrase structure rules**

- \( \text{IP} \rightarrow \{ \text{NP}|\text{PP} \} \quad \text{I'} \quad \)
  - Either a NP or a PP can appear in the specifier position.
  - \{...|...\} mark disjunction.
- \( \text{VP} \rightarrow \text{V} \ (\text{NP}) \ \text{PP}^* \)
  - \((\text{NP})\): an optional NP
  - \(*\): can repeat zero or many times.
ID/LP rules

- Immediate Dominance (ID) rules: dominance relations
- Linear Precedence (LP) rules: precedence constraints

**ID rule: Using commas**

\[
\begin{align*}
\text{VP} & \rightarrow V, \text{NP} \\
\Rightarrow & \quad \text{VP} \rightarrow V \text{ NP} \\
\Rightarrow & \quad \text{VP} \rightarrow \text{NP} \ V
\end{align*}
\]

**LP ordering constraint: Using \(\prec\)**

\[
\begin{align*}
\text{VP} & \rightarrow V, \text{NP} \quad V \prec \text{NP} \\
\Rightarrow & \quad \text{VP} \rightarrow V \text{ NP} \\
\text{VP} & \rightarrow V, \text{NP}, \text{PP} \quad V \prec \text{NP}, V \prec \text{PP} \\
\Rightarrow & \quad \{\text{VP} \rightarrow V \text{ NP} \text{ PP} \mid \text{VP} \rightarrow V \text{ PP} \text{ NP}\}
\end{align*}
\]
F-structure representation

Described by functions

- In LFG, functional information is formally represented by the f-structure.
- Mathematically, the f-structure can be thought of as a function from attributes to values,
- or equivalently as a set of ⟨attribute, value⟩ pairs.

Attribute-Value Matrix (AVM)

We can represent f-structures in tabular form

\[
\begin{bmatrix}
\text{ATTRIBUTE}_1 & \text{value}_1 \\
\text{ATTRIBUTE}_2 & \text{value}_2
\end{bmatrix}
\]
An example

Example

(17) I saw the girl.

LFG is functional!

Weiwei Sun

Lexical-Functional Grammar
Formal properties of f-structures

**Definition (F-structure)**

An f-structure is a finite set of pairs of attributes and values. An f-structures attributes can be:

- **atomic symbols**, e.g. SUBJ, OBJ, PRED

An f-structures values can be:

- **atomic symbols**, e.g. sg, 1st, +, past
- **semantic forms**, e.g. 'girl', 'see⟨SUBJ, OBJ⟩'
- **f-structures**
- **a set** of f-structures

**Attributes with the same values**

\[
\begin{bmatrix}
\text{ATT1} & 1 \\
\text{ATT2} & 1 \\
\end{bmatrix}
\begin{bmatrix}
\text{A1} & \text{v1} \\
\text{A2} & \text{v2} \\
\end{bmatrix}
\]
Example: *David yawned quietly yesterday.*

```
[SUBJ [PRED 'david']]
[PRED 'yawn<SUBJ>']
[TENSE past]
[ADJ {[PRED 'quietly'] {[PRED 'yesterday']}}]
```
**Example**: *David yawned quietly yesterday.*

\[
\begin{align*}
\text{SUBJ} & \quad [\text{PRED} \quad 'david'] \\
\text{PRED} & \quad 'yawn' \langle \text{SUBJ} \rangle' \\
\text{TENSE} & \quad \text{past} \\
\text{ADJ} & \quad \{ [\text{PRED} \quad 'quietly'] \} \\
& \quad \{ [\text{PRED} \quad 'yesterday'] \}
\end{align*}
\]
Example: *David yawned quietly yesterday.*

<table>
<thead>
<tr>
<th>SUBJ</th>
<th>PRED</th>
<th>'david'</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRED</td>
<td>'yawn(SUBJ)'</td>
<td></td>
</tr>
<tr>
<td>TENSE</td>
<td>past</td>
<td></td>
</tr>
<tr>
<td>ADJ</td>
<td>{ [ PRED</td>
<td>'quietly' ] }</td>
</tr>
<tr>
<td></td>
<td>{ [ PRED</td>
<td>'yesterday' ] }</td>
</tr>
</tbody>
</table>

Semantic form
**Example:** David yawned quietly yesterday.

```
[SUBJ [PRED 'david']
PRED 'yawn(SUBJ)'
TENSE past
ADJ { [PRED 'quietly']
      [PRED 'yesterday'] }]
```

F-structure
Example: *David yawned quietly yesterday.*

A set of f-structures
Subcategorization

- A semantic form may contain an argument list, next to its semantic predicate name, e.g.
  - 'yawn⟨SUBJ⟩'
  - 'see⟨SUBJ, OBJ⟩'
  - 'give⟨SUBJ, OBJ, OBJ THEME⟩'

- Lexical items select for grammatical functions (not for NPs, CP, etc)

- How to make sure that subcategorization requirements are fulfilled?

Well-formedness constraints on the f-structure

1. Completeness
2. Coherence
3. Consistency (uniqueness)
Principle of completeness

All governable functions present in the argument list of a semantic form must be present in the f-structure.

(18) *He devoured.

\[
\begin{align*}
\text{SUBJ} & & \left[ \text{PRED} \ 'pro' \right] \\
\text{PRED} & & '\text{devour}\langle \text{SUBJ, OBJ}\rangle' \\
\end{align*}
\]

Definition

Local Completeness: An f-structure is locally complete iff it contains all the governable functions that its predicate governs.

Completeness: An f-structure is complete iff it is locally complete and all its subsidiary f-structures are locally complete.
Principle of coherence

All governable functions present in the f-structure are also present in the argument list of the predicate.

(19) *David yawned the flower.

\[
\begin{align*}
\text{SUBJ} & \quad \text{PRED} \quad '\text{David}' \\
\text{OBJ} & \quad \text{PRED} \quad '\text{David}' \\
\text{PRED} & \quad '\text{yawn}(\text{SUBJ})'
\end{align*}
\]

Definition

**Local Coherence**: An f-structure is locally coherent iff all the governable functions it contains are governed by its predicate.

**Coherence**: An f-structure is coherent iff it is locally coherent and all its subsidiary f-structures are locally coherent.
The principle of consistency (uniqueness)

An attribute has a unique value.

**Definition**

**Consistency**: An f-structure is consistent iff all attributes have at most one value (which may be a set).

The value of the \((X)\text{ADJ}\) function is a set of f-structures:

- **PRED**: 'devour\(\langle\text{SUBJ, OBJ}\rangle\)'
- **SUBJ**: [PRED 'david']
- **OBJ**: [SPEC A [PRED 'sandwich']]
- **ADJ**: \{[PRED 'quietly'], [PRED 'at\langle\text{OBJ}\rangle'] [OBJ [PRED 'noon']]}
**Functional description**

**Function**

The *language* of functional descriptions is based on the mathematical conception of f-structures as *functions*:

\[
f(x) = x^2 \Rightarrow f(\text{SUBJ}) = g \Rightarrow (f \text{ SUBJ}) = g
\]

**Functional description**

\((f \text{ FEAT})\): the value of the FEAT feature in \(f\)

- \((f \text{ TENSE}) = \text{past}\)
- \((f \text{ SUBJ}) = g\)
- \(h \in (f \text{ ADJ})\)
How to construct f-structures?

Functional descriptions just present **partial constraints**. We find sound f-structures by solving a bunch of constraints, like solving an algebra problem.

\[
\begin{pmatrix}
2 & 1 \\
1 & 2
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
=
\begin{pmatrix}
3 \\
3
\end{pmatrix}
\implies
\begin{pmatrix}
x \\
y
\end{pmatrix}
=
\begin{pmatrix}
1 \\
1
\end{pmatrix}
\]

**Minimality**

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

**Subsumption**

A structure \( A \) subsumes a structure \( B \) iff \( A \) and \( B \) are identical or \( B \) contains \( A \) and additional information not included in \( A \).
An example

Example

\[
(f \text{ SUBJ}) = g \\
(f \text{ PRED}) = 'laugh\langle\text{SUBJ}\rangle' \\
(g \text{ PRED}) = 'david' \\
\]

\[
\downarrow \\
\F: \begin{bmatrix} 
\text{PRED} & 'laugh\langle\text{SUBJ}\rangle' \\
\text{SUBJ} & g: \begin{bmatrix} 
\text{PRED} & 'david' 
\end{bmatrix} 
\end{bmatrix} 
\]

We could enhance our *language* for functional descriptions.
Negation

An f-description can be negated; when this happens, the f-description must not be satisfied.

Example

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

justify  \( (f \text{ COMP COMPFORM}) \neq \text{IF} \)
Existential constraints

Existential constraint

An f-structure may be required to contain an attribute, but its value may be unconstrained.

Example

(21) a. the man who yawned
    b. the man who yawns
    c. the man who will yawn
    d. *the man who yawning

We can enforce the requirement for relative clauses to be tensed by means of a constraint like the following:

\( (f \text{ TENSE}) \)
Constraining equation

- Defining equations determine the minimal solution
- Constraining equations check that the minimal solution is wellformed.

Example

(22) a. Chris thought that David yawned.
    b. Chris thought David yawned.
    c. That David yawned surprised Chris.
    d. *David yawned surprised Chris.

- \((f \text{ COMPFORM}) = \text{THAT}\)
- \((f \text{ COMPFORM}) = c \text{THAT}\)
Inside-out constraint

Outside-in vs. Inside-out

- **Outside-in functional constraint:**
  - \((f \text{ PRED}) = \text{'rock'}\)
  - \((f \text{ CASE}) = \text{LOC}\)

- **Inside-out functional constraints:**
  - \(((OBL_{LOC} \ g) \text{ CASE}) = \text{ERG}\)
  - \((\text{SUBJ OBL}_{LOC} \ g)\)

**Example**

\((OBL_{LOC} \ g)\) is labeled \(f\): \(f = (OBL_{LOC} \ g)\)

\[
\begin{bmatrix}
\text{SUBJ} \\
\text{OBL}_{LOC}
\end{bmatrix}
\begin{bmatrix}
\text{CASE} & \text{ERG} \\
\text{PRED} & \text{'rock'}
\end{bmatrix}
\begin{bmatrix}
\text{CASE} \\
\text{LOC}
\end{bmatrix}
\]
Questions

1. Where can I find these functional descriptions?
2. What is the relation between c-structure and f-structure?
Weiwei Sun

Lexical-Functional Grammar

Outline

Grammatical Functions

Syntactic Descriptions

Structural Correspondences

Configurational vs. Nonconfigurational
Structural correspondences (1)

**Question**

- C-structures and f-structures represent different properties of an utterance.
- How can these structures be associated properly to a particular sentence?

**Codescription**

Simultaneously describing more than one structure: Each piece of the c-structure is directly associated with a description of part of the f-structure.

**Example**

\[
S_{f_1} \quad f_1, f_3[\text{SUBJ} \quad f_2] \\
\text{NP}_{f_2} \quad \text{VP}_{f_3}
\]
Structural correspondences (2)

**Correspondence function**

A function $\phi$ maps c-structures to f-structures $\phi : N \mapsto F$.

- $\phi(n)$: f-structure associated with $n$
- $\phi(M(n))$: f-structure associated with the parent node of $n$

**Example**

```
S
  /\  
NP VP
  /\  
N  V
```

David smiled

```
\begin{array}{l}
PRED \quad '\text{smile}'_{\text{SUBJ}} \\
TENSE \quad \text{past} \\
\text{SUBJ} \quad g_{\text{NP}}, g_{\text{N}}: \\
\quad \begin{cases} 
\text{PRED} \quad '\text{david}' \\
\text{NUM} \quad \text{sg} \\
\text{PERS} \quad 3\text{rd}
\end{cases}
\end{array}
```

$f_{S}, f_{VP}, f_{V}$:
The head convention

A constituent structure phrase and its head map to the same f-structure.

Example

- S, VP and V map to the same f-structure $f$.
- NP and N map to the same f-structure $g$.
Annotating PS-rules: heads

Consider the rewrite rule: \( VP \rightarrow V \)

Notation

\( VP \) and \( V \) have the same f-structure by annotating the \( V \)-node:

\[
\begin{align*}
VP & \rightarrow V \\
\phi(M(n)) & = \phi(n)
\end{align*}
\]

The equation indicates that the f-structure of the parent node of \( V \) (i.e. \( \phi(M(n)) \)) is equal to one of the node \( V \) (\( \phi(n) \)).

An alternative notation

\[
\begin{align*}
\uparrow & \equiv \phi(M(n)) & VP \rightarrow V \\
\downarrow & \equiv \phi(n) & \uparrow=\downarrow
\end{align*}
\]
Annotating PS-rules: grammatical functions

**Notation**

\[ S \rightarrow NP \quad \phi(M(n))_{\text{SUBJ}} = \phi(n) \quad \phi(M(n)) = \phi(n) \]

The first equation indicates that the \text{SUBJ} feature of the f-structure of the parent node of NP (i.e. \( \phi(M(n))_{\text{SUBJ}} \)) is equal to the f-structure of NP (\( \phi(n) \)).

**An alternative notation**

\[ \uparrow \equiv \phi(M(n)) \]

\[ \downarrow \equiv \phi(n) \]

\[ S \rightarrow \quad \text{(\( \uparrow \text{SUBJ}\))=} \downarrow \quad \uparrow=\downarrow \]

**Key idea**

Local description of partial structures
Instantiation

\[
S \\
\text{NP} \quad \text{VP} \\
\text{N} \quad \text{V} \\
\text{David} \quad \text{smiled}
\]

\[
f_S, f_{\text{NP}}, f_V: \\
PRED \quad '\text{smile}\langle \text{SUBJ} \rangle'$ \\
TENSE \quad \text{past} \\
\text{SUBJ} \quad g_{\text{NP}}, g_{\text{N}}: \\
PRED \quad '\text{david}' \\
\text{NUM} \quad \text{sg} \\
PERS \quad \text{3rd} \\
\]

\[
\begin{align*}
(f_S \text{ SUBJ}) &= f_{\text{NP}} \\
f_{\text{NP}} &= f_{\text{N}} \\
(f_{\text{N}} \text{ PRED}) &= '\text{david}' \\
(f_{\text{N}} \text{ NUM}) &= \text{sg} \\
(f_{\text{N}} \text{ PERS}) &= \text{3rd} \\
(f_S) &= f_{\text{VP}} \\
(f_{\text{VP}}) &= f_V \\
(f_V \text{ PRED}) &= '\text{smile}\langle \text{SUBJ} \rangle'$ \\
(f_V \text{ TENSE}) &= \text{past}
\end{align*}
\]
Adjuncts

- The attribute ADJ takes a set as its value
- The c-structure/f-structure correspondence rule expresses membership to a set.

$$N \rightarrow \text{ADJP} \quad N$$
$$\downarrow \in (\uparrow \text{ADJ}) \quad \uparrow = \downarrow$$

**Example**

- $N$ (girl)
- $A \downarrow \in (\uparrow \text{ADJ}) \quad \uparrow = \downarrow$
- pretty
- num = sg
- pers = 3rd
- adj

$$\left[ \begin{array}{c}
\text{PRED} & 'girl' \\
\text{NUM} & \text{sg} \\
\text{PERS} & 3rd \\
\text{ADJ} & \left\{ \begin{array}{c}
\text{PRED} & 'pretty' \\
\end{array} \right. \\
\end{array} \right]$$
Lexical entries

**Notation**

In lexical entries, information about the item’s f-structure is represented in the same way:

\[
\text{smiled} \quad V \quad (↑ \text{PRED}) = 'smile(\text{SUBJ})' \\
(↑ \text{TENSE}) = \text{past}
\]

**An alternative notation**

The equivalent phrase structure rule:

\[
V \quad \rightarrow \quad \text{smiled} \\
(↑ \text{PRED}) = 'smile(\text{SUBJ})' \\
(↑ \text{TENSE}) = \text{past}
\]
Lexical integrity

Principle lexical integrity

Morphologically complete words are leaves of the c-structure tree, and each leaf corresponds to one and only one c-structure node.

- Fully inflected words are the terminal elements of the c-structures.
- Every word belongs to exactly one node.
- The structural formation of words is independent of the structural formation of phrases.

Warlpiri

- The relative order of words in sentences is extremely free.
- The relative order of stems and inflections in words is fixed.
Lexical integrity: A seemly irrelevant illustration

Chomsky (1970) demonstrated that NPs based on “derived” nouns (i.e. nouns that have verbal counterparts) have exactly the syntax of NPs based on underived nouns.

Chomsky reasoned,
we need to recognize that both types are base-generated as nouns instead of attempting to derive certain NPs from clausal counterparts

---

**Target syntax argument**

To generate A directly instead of deriving it from C if there exists a pattern B that has the same target syntax as A and is clearly not derived from C.

\[ C \rightarrow A \]

B
Lexical integrity: A seemly irrelevant illustration

Chomsky (1970) demonstrated that NPs based on “derived” nouns (i.e. nouns that have verbal counterparts) have exactly the syntax of NPs based on underived nouns.

Chomsky reasoned, we need to recognize that both types are base-generated as nouns instead of attempting to derive certain NPs from clausal counterparts

Target syntax argument

To generate A directly instead of deriving it from C if there exists a pattern B that has the same target syntax as A and is clearly not derived from C.
Lexical integrity: A seemly irrelevant illustration

Chomsky (1970) demonstrated that
NPs based on “derived” nouns (i.e. nouns that have verbal counterparts) have exactly the syntax of NPs based on underived nouns.

Chomsky reasoned,
we need to recognize that both types are base-generated as nouns instead of attempting to derive certain NPs from clausal counterparts

**Target syntax argument**

To generate A directly instead of deriving it from C if there exists a pattern B that has the same target syntax as A and is clearly not derived from C.

\[
C \rightarrow A \\
C \rightarrow B
\]
**Lexical integrity: A seemly irrelevant illustration**

Chomsky (1970) demonstrated that NPs based on “derived” nouns (i.e. nouns that have verbal counterparts) have exactly the syntax of NPs based on underived nouns.

Chomsky reasoned, we need to recognize that both types are base-generated as nouns instead of attempting to derive certain NPs from clausal counterparts.

**Target syntax argument**

To generate A directly instead of deriving it from C if there exists a pattern B that has the same target syntax as A and is clearly not derived from C.

\[
\text{C} \nleftrightarrow \text{A} \\
\text{B}
\]
Inflected words (1)

According to the principle of lexical integrity

- Internal structural formation of words invisible to c-structure principles.
- But the f-structures specified by words is allowed to unify with the f-structures of the syntactic contexts.

Example: Descriptions

<table>
<thead>
<tr>
<th>lion</th>
<th>(↑  PRED) = 'lion'</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>(↑  NUM) = plurals</td>
</tr>
</tbody>
</table>

Example: Structures

\[
f_N \left[ \begin{array}{c} \text{PRED} \\ \text{NUM} \end{array} \right] \begin{array}{c} '\text{lion}' \\ \text{pl} \end{array} \]

Weiwei Sun
Lexical-Functional Grammar
Inflected words (2)

Example: Descriptions

live (↑ PRED) = ‘live⟨...⟩’
-s (↑ TENSE) = pres
(↑ SUBJ) =↓
  (↓ PERS) = 3
  (↓ NUM) = sg

Example: Structures

\[ V_{f_N} \]
\[ \xrightarrow{\ \ } \]
\[ \text{live } s_{f_s} \]

\[ f_V \]
\[ \begin{bmatrix}
\text{PRED} & \text{‘live⟨...⟩’} \\
\text{TENSE} & \text{pres} \\
\text{SUBJ} & f_s \begin{bmatrix}
\text{NUM} & \text{sg} \\
\text{PERS} & 3
\end{bmatrix}
\end{bmatrix} \]
Example 1: Basis

Example

(23) David yawned

P122. *Lexical-Functional Grammar*
On whiteboard
Outline

Grammatical Functions

Syntactic Descriptions

Structural Correspondences

Configurational vs. Nonconfigurational
Two types of encoding

Grammatical functions are encoded in different ways in different languages. Two fundamental ways for language to realize underlying concepts:

- Phrase structure (groups)
- Morphology (shapes)

Morphology competes with syntax

- English: phrase structure strategy (configurational)
- Warlpiri: morphological strategy (nonconfigurational)

Languages may tend to employ one type of encoding more heavily, but there are many cases in which a single language employs mixed strategies.
English

- SUBJ, OBJ, and OBLθ are primarily encoded configurationally.

⇒ PS rules contain specifications of particular grammatical functions.

**PS rules**

- **IP** → **NP** → **I'**
  - $(↑ \text{SUBJ}) = ↓ \quad ↑ = ↓$

- **I'** → **I** → **VP**
  - $↑ = ↓ \quad ↑ = ↓$

- **VP** → **V'**
  - $↑ = ↓$

- **VP** → **V** → **NP**
  - $↑ = ↓ \quad (↑ \text{OBJ}) = ↓$
Warlpiri (1)

Grammatical function is determined by morphological casemarking on the argument phrase

\[ GF \equiv \{ \text{SUBJ}|\text{OBJ}|\text{OBJ}_\theta|\text{OBL}_\theta|\text{COMP}|\text{XCOMP}|\text{ADJ}|\text{XADJ} \} \]

---

**PS rules**

\[
\begin{align*}
\text{IP} & \rightarrow \quad \text{NP} \quad \text{I}' \\
\uparrow & \quad (\uparrow \text{FOCUS}) = \downarrow \quad \uparrow = \downarrow \\
& \quad (\uparrow \text{GF}) = \downarrow \\
\text{I}' & \rightarrow \quad \text{I} \quad \text{S} \\
\uparrow = \downarrow & \quad \uparrow = \downarrow \\
\text{S} & \rightarrow \quad \{ \text{NP} \mid \text{V} \}^* \\
(\uparrow \text{GF}) & = \downarrow \quad \uparrow = \downarrow 
\end{align*}
\]
The Warlpiri verb specifies a great deal of information about its arguments.

**Lexical entry**

- **panti-rni**  
  V  
  \( \uparrow \text{PRED} = \text{‘spear} (\text{SUBJ,OBJ}) \)’  
  \( \uparrow \text{OBJ CASE} = \text{ABS} \)  
  \( \uparrow \text{OBJ PRED} = \text{‘pro} \)’

- **ngarrka-ngku**  
  V  
  \( \uparrow \text{PRED} = \text{‘man} \)’  
  \( \uparrow \text{CASE} = \text{ERG} \)

(24)  
ngarrka-ngku ka wawirri panti-rni  
man-ERG PRES kangaroo.ABS spear-NONPAST  
‘The man is spearing the kangaroo.’
Warlpiri: An example

Example

P129. *Lexical-Functional Grammar.*

On whiteboard
Semantic roles, syntactic constituents, and grammatical functions belong to parallel information structures of very different formal character. They are related not by proof-theoretic derivation but by structural correspondences, as a melody is related to the words of a song. The song is decomposable into parallel melodic and linguistic structures, which jointly constrain the nature of the whole. In the same way, the sentences of human language are themselves decomposable into parallel systems of constraints—structural, functional, semantic, and prosodic—which the whole must jointly satisfy.

Joan Bresnan
Summary

Main ideas

- **LFG** is a nontransformational theory.
  - The idea is fairly radical when first proposed.
- A **formal system** to model human speech: fits in the tradition of generative grammar.
- Psychological plausibility: represent a native speaker’s syntactic knowledge appropriately.
- Strong **typological (universal) basis**: analyses should capture cross-linguistic similarities.

LFG brings scholars from different fields together:
- Theoretical linguists
- Descriptive, typological linguists
- Computational linguistics
Summary

LFG is lexical

- LFG assumes that words and lexical items are as important in providing grammatical information as syntactic elements.
- The lexicon is richly structured to capture linguistic generalizations.

LFG is functional

- Functional syntactic concepts like subject and object are relevant for the analysis of every language.
- Grammatical functions like subject and object are primitives of the theory.
Design Principles

**Principle I: Variability**
External structures (modelled by LFG c-structures) vary across languages.

**Principle II: Universality**
Internal structures (modelled by LFG f-structures) are largely invariant across languages.

**Principle III: Monotonicity**
The mapping from c-structure to f-structure is not one-to-one, but it is monotonic (information-preserving).
Reading

- Chap. 1, 2, 5. *Lexical Functional Grammar*
- Chap. 4. *Lexical Functional Grammar*
- Chapter 13. *Grammatical theory: From transformational grammar to constraint-based approaches*