Lexical Functional Grammar

Mary Dalrymple
Centre for Linguistics and Philology
Oxford University

York Frameworks, 4 May 2010
The constraint-based approach

Nontransformational, constraint-based theories (Lexical Functional Grammar, Head-Driven Phrase Structure Grammar, Construction Grammar, Simpler Syntax ...):

- Different aspects of linguistic structure are realised by different but related linguistic representations. Movement/transformations do not play a role.
“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.” (Bresnan, 1990)

What theoretical architecture best reflects this view?
Theories and frameworks

Formal linguistic framework: A set of linguistic objects, rules, and/or processes, and a formal vocabulary for talking about them. Example: X-bar theory: phrase structure rules and trees.
Theories and frameworks

Formal linguistic framework: A set of linguistic objects, rules, and/or processes, and a formal vocabulary for talking about them. Example: X-bar theory: phrase structure rules and trees.

- Formally explicit: Provides a way of making systematic, clear, and testable claims about phrase structure.
Theories and frameworks

Formal linguistic framework: A set of linguistic objects, rules, and/or processes, and a formal vocabulary for talking about them. Example: X-bar theory: phrase structure rules and trees.

- Formally explicit: Provides a way of making systematic, clear, and testable claims about phrase structure.
- Embodies some assumptions about how language works: phrases (like VP) have heads (like V),
Formal linguistic framework: A set of linguistic objects, rules, and/or processes, and a formal vocabulary for talking about them. Example: X-bar theory: phrase structure rules and trees.

- Formally explicit: Provides a way of making systematic, clear, and testable claims about phrase structure.
- Embodies some assumptions about how language works: phrases (like VP) have heads (like V),
- but general enough to encompass a range of different theories of phrase structure.
Linguistic theory: A set of claims about the structure of language(s), which may (or may not) be stated with reference to a particular formal framework.
Linguistic theory: A set of claims about the structure of language(s), which may (or may not) be stated with reference to a particular formal framework.

- Example: The claim that all maximal X-bar projections have bar level 2 (there is no N″″ or V″″″″).
Theories and frameworks

Linguistic theory: A set of claims about the structure of language(s), which may (or may not) be stated with reference to a particular formal framework.

- Example: The claim that all maximal X-bar projections have bar level 2 (there is no N'' or V''''

- A well-designed formal framework guides development of theory by providing explicit representations and theoretical vocabulary, and aids the linguist in developing better intuitions about language and (hence) better theories of linguistic structure.
Alternative view (NOT LFG): the formal framework should not allow the linguist to formulate rules or describe constructions that are linguistically impossible.
Alternative view (NOT LFG): the formal framework should not allow the linguist to formulate rules or describe constructions that are linguistically impossible.

This is a very strong view; e.g. disallows standard phrase structure rules, since impossible languages can be characterised with (unconstrained) phrase structure rules (e.g., a language where every sentence is at least 3000 words long).
Alternative view (NOT LFG): the formal framework should not allow the linguist to formulate rules or describe constructions that are linguistically impossible.

This is a very strong view; e.g. disallows standard phrase structure rules, since impossible languages can be characterised with (unconstrained) phrase structure rules (e.g., a language where every sentence is at least 3000 words long).

The LFG view (also HPSG, other constraint-based theories): use a simple, clean formal framework, and formulate linguistic theory as a set of claims stated with reference to the framework.
Alternative view (NOT LFG): the formal framework should not allow the linguist to formulate rules or describe constructions that are linguistically impossible.

This is a very strong view; e.g. disallows standard phrase structure rules, since impossible languages can be characterised with (unconstrained) phrase structure rules (e.g., a language where every sentence is at least 3000 words long).

The LFG view (also HPSG, other constraint-based theories): use a simple, clean formal framework, and formulate linguistic theory as a set of claims stated with reference to the framework.

Advantage: No need to throw away or reformulate the framework when revisions are needed to the theory.
LFG framework

Formal framework of LFG:

- Different aspects of linguistic structure are represented in different ways, and are related to one another by piecewise correspondence (parts of one structure are related to parts of another structure).
Formal framework of LFG:

- Different aspects of linguistic structure are represented in different ways, and are related to one another by piecewise correspondence (parts of one structure are related to parts of another structure).

- The core of the formal framework of LFG has remained remarkably stable since its beginnings in the late 1970s.
LFG framework

Formal framework of LFG:

- Different aspects of linguistic structure are represented in different ways, and are related to one another by piecewise correspondence (parts of one structure are related to parts of another structure).

- The core of the formal framework of LFG has remained remarkably stable since its beginnings in the late 1970s.

- LFG-based theories of linguistic phenomena have evolved substantially since that time, and continue to evolve as new areas are explored and new theoretical proposals are formulated and evaluated.
Two aspects of syntactic structure:

- **Functional structure** is the abstract functional syntactic organisation of the sentence, familiar from traditional grammatical descriptions, representing syntactic predicate-argument structure and functional relations like subject and object.

- **Constituent structure** is the overt, more concrete level of linear and hierarchical organisation of words into phrases.
LFG’s c-structure and f-structure

```
IP
  NP  I'
    N   VP
      David  V'
          V  NP
            greeted  N
              Chris

[SUBJ [PRED ‘David’]]
[OBJ [PRED ‘Chris’]]
[PRED ‘GREET⟨SUBJ,OBJ⟩’]
```
C-structure and f-structure

In GB/Principles and Parameters/Minimalism:

- C-structure = PF or Spellout?
- F-structure = S-Structure or LF?
Since the inception of the theory, there has been much work on other linguistic levels and their relation to c-structure and f-structure:

- Argument structure and argument linking (Bresnan & Zaenen, 1990; Butt, 1995)

- The syntax-semantics interface: “glue” semantics (Dalrymple, 1999, 2001; Asudeh, 2004): interesting relations to categorial approaches, though with different assumptions about the relation to syntactic structure

- Information structure and its relation to syntax and semantics (Butt & King, 2000; Dalrymple & Nikolaeva, 2010)

- Prosodic structure and its relation to syntax and semantics (Mycock, 2006)
LFG as a component of other approaches

LFG has also been adopted as a component of OT and DOP:

- OT-LFG: Optimality-theoretic syntax with an LFG base (Bresnan, 2000)
- LFG-DOP: Data-Oriented Parsing with an LFG base (see http://www.nclt.dcu.ie/lfg-dop/publications.html)
What information does functional structure represent?
What information does functional structure represent?

- Abstract syntactic relations (familiar from traditional grammar) like subject, object, adjunct
What information does functional structure represent?

- Abstract syntactic relations (familiar from traditional grammar) like subject, object, adjunct
- Locus of subcategorisation
What information does functional structure represent?

- Abstract syntactic relations (familiar from traditional grammar) like subject, object, adjunct
- Locus of subcategorisation
- Criteria: anaphoric binding patterns, long-distance dependencies, control, honorification, agreement, casemarking, ...
F-structure

What information does functional structure represent?

- Abstract syntactic relations (familiar from traditional grammar) like subject, object, adjunct
- Locus of subcategorisation
- Criteria: anaphoric binding patterns, long-distance dependencies, control, honorification, agreement, casemarking, ...
- F-structure vocabulary is universal across languages
Functional structure

\[
\begin{array}{c}
\text{PRED} & \text{‘GO} \langle \text{SUBJ} \rangle \text{’} \\
\text{TENSE} & \text{PAST} \\
\text{SUBJ} & \begin{array}{c}
\text{PRED} & \text{‘DAVID’} \\
\text{NUM} & \text{SG}
\end{array}
\end{array}
\]
Functional structure

\[
\begin{bmatrix}
\text{PRED} & '\text{GO}(<\text{SUBJ}>)' \\
\text{TENSE} & \text{PAST} \\
\text{SUBJ} & \begin{bmatrix}
\text{PRED} & '\text{DAVID}' \\
\text{NUM} & \text{SG}
\end{bmatrix}
\end{bmatrix}
\]

- PRED, TENSE, NUM: attributes
Functional structure

\[
\begin{align*}
&\text{PRED} \quad \text{‘GO} \langle \text{SUBJ} \rangle \text{’} \\
&\text{TENSE} \quad \text{PAST} \\
&\text{SUBJ} \quad \begin{cases}
\text{PRED} & \text{‘DAVID’} \\
\text{NUM} & \text{SG}
\end{cases}
\end{align*}
\]

- PRED, TENSE NUM: attributes
- ‘GO} \langle \text{SUBJ} \rangle \text{’, DAVID, SG: values
Functional structure

\[
\begin{array}{l}
\text{PRED} \quad \text{‘GO(SUBJ)’} \\
\text{TENSE} \quad \text{PAST} \\
\text{SUBJ} \quad \begin{array}{l}
\text{PRED} \quad \text{‘DAVID’} \\
\text{NUM} \quad \text{SG}
\end{array}
\end{array}
\]

- **PRED, TENSE, NUM**: attributes
- **‘GO(SUBJ)’, DAVID, SG**: values
- **PAST, SG**: symbols (a kind of value)
Functional structure

\[
\begin{array}{c}
\text{PRED} \quad \text{’GO\langle\text{SUBJ}\rangle’} \\
\text{TENSE} \quad \text{PAST} \\
\text{SUBJ} \quad \begin{array}{c}
\text{PRED} \quad \text{’DAVID’} \\
\text{NUM} \quad \text{SG}
\end{array}
\end{array}
\]

- PRED, TENSE, NUM: attributes
- ’GO\langle\text{SUBJ}\rangle’, DAVID, SG: values
- PAST, SG: symbols (a kind of value)
- ’BOY’, ’GO\langle\text{SUBJ}\rangle’: semantic forms
An f-structure can be the value of an attribute. Attributes with f-structure values are the grammatical functions: SUBJ, OBJ, OBJ_θ, COMP, XCOMP, ...
A set of f-structures can also be a value of an attribute.
Sets of f-structures represent:

- adjuncts (there can be more than one adjunct) or
Sets of f-structures represent:

- adjuncts (there can be more than one adjunct) or
- coordinate structures (there can be more than one conjunct)
Describing F-structures

\[(f \text{ num}) = \text{SG}\]

is a functional equation.

\[(f \ a) = v\] holds if and only if \(f\) is an f-structure, \(a\) is a symbol, and the pair \(\langle a, v \rangle \in f\).

A set of formulas describing an f-structure is a functional description.
\[(f \text{ SUBJ NUM}) = (g \text{ NUM}) = \text{SG} \]

\[
f \begin{bmatrix}
\text{PRED} & \text{GO(SUBJ)}' \\
\text{SUBJ} & g \begin{bmatrix}
\text{PRED} & \text{DAVID'} \\
\text{NUM} & \text{SG}
\end{bmatrix}
\end{bmatrix}
\]
Hindi verbs show person, number, and gender agreement:

Ram calegaa
Ram go.\textsc{future}
‘Ram will go.’

\begin{align*}
\text{Ram} & \quad (g\ \text{PRED}) = \text{'Ram'} \\
& \quad (g\ \text{CASE}) = \text{NOM} \\
& \quad (g\ \text{PERS}) = 3 \\
& \quad (g\ \text{NUM}) = \text{SG} \\
& \quad (g\ \text{GEND}) = \text{MASC} \\
\text{calegaa} & \quad (f\ \text{PRED}) = \text{'go} \langle \text{SUBJ} \rangle' \\
& \quad (f\ \text{SUBJ} \ \text{CASE}) = \text{NOM} \\
& \quad (f\ \text{SUBJ} \ \text{PERS}) = 3 \\
& \quad (f\ \text{SUBJ} \ \text{NUM}) = \text{SG} \\
& \quad (f\ \text{SUBJ} \ \text{GEND}) = \text{MASC} \\
& \quad (f\ \text{SUBJ}) = g
\end{align*}
\( (g \text{ PRED}) = '\text{Ram}' \)
\( (g \text{ CASE}) = \text{NOM} \)
\( (g \text{ PERS}) = 3 \)
\( (g \text{ NUM}) = \text{SG} \)
\( (g \text{ GEND}) = \text{MASC} \)

\( (f \text{ PRED}) = '\text{GO}⟨\text{SUBJ}⟩' \)
\( (f \text{ SUBJ}) = g \)
\( (f \text{ SUBJ CASE}) = (g \text{ CASE}) = \text{NOM} \)
\( (f \text{ SUBJ NUM}) = (g \text{ NUM}) = \text{SG} \)
\( (f \text{ SUBJ PERS}) = (g \text{ PERS}) = 3 \)
\( (f \text{ SUBJ GEND}) = (g \text{ GEND}) = \text{MASC} \)
Formal descriptions: LFG vs HPSG

- HPSG takes a different view of formal descriptions from LFG. The HPSG view goes back to Functional Unification Grammar (Kay, 1984), where unification (an operation on structures) was used to combine structures:

- in HPSG, the constraints look (as much as possible) like the structures.

- That is why you sometimes see a set of instructions in what looks like a representation – it is actually a constraint or description in the (apparent) form of a structure.
HPSG’s Argument Realisation Principle (Sag et al., 2003, 432):

\[
\text{word: } \begin{bmatrix}
\text{SYN} \\
\text{VAL} \\
\text{GAP} \\
\text{ARG-STR}
\end{bmatrix}
\begin{bmatrix}
\text{SPR} & A \\
\text{COMPS} & B \oplus C
\end{bmatrix}
\]

\(\ominus\): list subtraction
\(\oplus\): list addition
Expressing generalisations over functional descriptions: templates (Dalrymple et al., 2004; Asudeh et al., 2008)
Expressing generalisations over functional descriptions: **templates** (Dalrymple et al., 2004; Asudeh et al., 2008)

Templates are names for bundles of functional equations that characterise a construction.
Expressing generalisations over functional descriptions: templates (Dalrymple et al., 2004; Asudeh et al., 2008)

Templates are names for bundles of functional equations that characterise a construction.

Templates can be defined in terms of other templates, giving something like the inheritance hierarchy of HPSG (but involving relations among descriptions rather than linguistic objects).
Expressing generalisations over functional descriptions: **templates** (Dalrymple et al., 2004; Asudeh et al., 2008)

Templates are names for bundles of functional equations that characterise a construction.

Templates can be defined in terms of other templates, giving something like the inheritance hierarchy of HPSG (but involving relations among **descriptions** rather than **linguistic objects**).

Templates can be associated with words or with units that are bigger than words, and are used to describe constructions in the Construction Grammar sense.
Generalisations and constructions

- Expressing generalisations over functional descriptions: templates (Dalrymple et al., 2004; Asudeh et al., 2008)

- Templates are names for bundles of functional equations that characterise a construction.

- Templates can be defined in terms of other templates, giving something like the inheritance hierarchy of HPSG (but involving relations among descriptions rather than linguistic objects).

- Templates can be associated with words or with units that are bigger than words, and are used to describe constructions in the Construction Grammar sense.

- This is a relatively recent area of exploration in LFG.
Subcategorisation requirements are imposed at f-structure (not c-structure) – a predicate specifies a set of grammatical functions, and the phrase structure grammar of the language determines where in the tree these functions can appear. Subcategorisation requirements are specified by **semantic forms**:

\[(f \text{ PRED}) = \text{‘GO}(\text{SUBJ})\’\]

Semantic forms have **argument lists** that list the arguments they require.
Grammatical functions

<table>
<thead>
<tr>
<th>Non-argument</th>
<th>TOPIC</th>
<th>Discourse function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-argument</td>
<td>FOCUS</td>
<td></td>
</tr>
<tr>
<td>Argument Core</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(governable)</td>
<td>SUBJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OBJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OBJθ</td>
<td></td>
</tr>
<tr>
<td>Non-core</td>
<td>OBLθ</td>
<td>Non-discourse function</td>
</tr>
<tr>
<td></td>
<td>COMP</td>
<td></td>
</tr>
<tr>
<td>Non-argument</td>
<td>ADJ(unct)</td>
<td></td>
</tr>
</tbody>
</table>

(from Börjars & Vincent 2004)
Completeness requires: All arguments which are listed in the semantic form must be present.

\((f \text{ PRED}) = \text{ ‘GO(SUBJ)’} \)

“Go” must have a SUBJ.
Coherence requires: No arguments which are not listed in the semantic form may be present.

\[(f \text{ PRED}) = \text{‘GO⟨SUBJ⟩’} \]

“Go” may not have a OBJ.
Coherence requires: No arguments which are not listed in the semantic form may be present.

\[(f \text{ PRED}) = \text{‘GO}(\text{SUBJ})\text{’}\]

“Go” may not have a OBJ.

Completeness and coherence are the equivalent (more or less) of the Theta Criterion of GB theory, or the Valence Principle and Root Condition of HPSG.
*wati ka parnka-mi karnta
  man.ABS PRES run-NONPAST woman.ABS
‘The man runs the woman.’ (Warlpiri)

wati \((g \text{ PRED}) = \text{‘MAN’}\)
karnta \((g \text{ PRED}) = \text{‘WOMAN’}\)

Each use of a semantic form is unique.
Conflicting Semantic Forms

\[ \text{wati} \quad (g\ \text{PRED}) = \text{‘MAN’} \]
\[ \text{karnta} \quad (g\ \text{PRED}) = \text{‘WOMAN’} \]

Ill-formed f-structure:

\[
\begin{bmatrix}
\text{PRED} & \text{‘RUN(SUBJ)’} \\
\text{TENSE} & \text{PRES} \\
\text{SUBJ} & g [\text{PRED} \quad \text{‘MAN’/‘WOMAN’}] \\
\end{bmatrix}
\]
Optionality

\[ njûchi \ zi-ná-lúm-a \quad alenje \]
bees \quad \text{SUBJ-PAST-bite-INDICATIVE} \quad \text{hunters}

‘The bees bit the hunters.’ (Chichewa)

\[ zi-ná-lúm-a \quad alenje \]
\text{SUBJ-PAST-bite-INDICATIVE} \quad \text{hunters}

‘They bit the hunters.’

\[ zi-ná-lúm-a: \quad \((f \ \text{SUBJ} \ \text{PRED}) = \text{‘PRO’})\]

\[ zi-ná-lúm-a \text{ optionally contributes a PRED for its SUBJ.} \]
Overt subject

njûchi zi-ná-lúm-a  alenje
bees  SUBJ-PAST-bite-INDICATIVE hunters
‘The bees bit the hunters.’

\[
f \left[ \begin{array}{c}
\text{PRED} & \langle \text{SUBJ}, \text{OBJ} \rangle' \\
\text{SUBJ} & \left[ \begin{array}{c}
\text{PRED} & \langle \text{BEES} \rangle' \\
\text{NOUNCLASS} & 10
\end{array} \right]
\\
\text{OBJ} & \left[ \begin{array}{c}
\text{PRED} & \langle \text{HUNTERS} \rangle' \\
\text{NOUNCLASS} & 2
\end{array} \right]
\end{array} \right]
\]
No overt subject

zi-ná-lúm-a alenje
SUBJ-PAST-bite-INDICATIVE hunters
‘They bit the hunters.’

\[
\begin{align*}
\text{f} & \left[ \begin{array}{c}
\text{PRED} & \text{‘BITE\langle SUBJ,OBJ\rangle’} \\
\text{SUBJ} & \left[ \begin{array}{c}
\text{PRED} & \text{‘PRO’} \\
\text{NOUNCLASS} & 10 \\
\end{array} \right] \\
\text{OBJ} & \left[ \begin{array}{c}
\text{PRED} & \text{‘HUNTERS’} \\
\text{NOUNCLASS} & 2 \\
\end{array} \right]
\end{array} \right]
\end{align*}
\]
Optionality: Clitics

Juan vió a Pedro.
Juan saw PREP Pedro
‘Juan saw Pedro.’ (Spanish)

Juan lo vió.
Juan ACC.MASC.SG.CLITIC saw
‘Juan saw him.’

Juan lo vió a Pedro.
Juan ACC.MASC.SG.CLITIC saw PREP Pedro
‘Juan saw Pedro.’
Optionality: Clitics

\[Pedro\] 
\[(f \ \text{pred}) = \text{‘Pedro’}\]
\[(f \ \text{gend}) = \text{masc}\]
\[(f \ \text{num}) = \text{sg}\]

\[lo\] 
\[(()(f \ \text{pred}) = \text{‘pro’})\]
\[(f \ \text{gend}) = \text{masc}\]
\[(f \ \text{num}) = \text{sg}\]
Optionality: Clitics

\[
\begin{align*}
\text{Pedro} & \quad (f \text{ PRED}) = \text{Pedro} \\
& \quad (f \text{ GEND}) = \text{MASC} \\
& \quad (f \text{ NUM}) = \text{SG} \\
\text{lo} & \quad ((f \text{ PRED}) = \text{PRO}) \\
& \quad (f \text{ GEND}) = \text{MASC} \\
& \quad (f \text{ NUM}) = \text{SG} \\
\text{lo} \text{ optionally contributes a PRED.}
\end{align*}
\]
Optionality: Clitics

Juan lo vió a Pedro.
Juan ACC.MASC.SG.CLITIC saw PREP Pedro
‘Juan saw Pedro.’
Juan lo vió a Pedro.

Juan ACC.MASC.SG.CLITIC saw PREP Pedro

‘Juan saw Pedro.’

\[
\begin{align*}
\text{PRED} & \quad \text{‘SEE\langle SUBJ,OBJ\rangle’} \\
\text{SUBJ} & \quad \text{PRED} \quad \text{‘JUAN’} \\
& \quad \text{GEND} \quad \text{MASC} \\
& \quad \text{NUM} \quad \text{SG} \\
\text{OBJ} & \quad \text{PRED} \quad \text{‘PEDRO’} \\
& \quad \text{GEND} \quad \text{MASC} \\
& \quad \text{NUM} \quad \text{SG}
\end{align*}
\]
Optionality and clitic doubling

Juan lo vió.
Juan ACC.MASC.SG.CLITIC saw
‘Juan saw him.’

\[
\begin{array}{c}
\text{PRED} & \text{‘SEE\langle SUBJ,OBJ\rangle’} \\
\text{SUBJ} & \begin{array}{c}
\text{PRED} & \text{‘JUAN’} \\
\text{GEND} & \text{MASC} \\
\text{NUM} & \text{SG}
\end{array} \\
\text{OBJ} & \begin{array}{c}
\text{PRED} & \text{‘PRO’} \\
\text{GEND} & \text{MASC} \\
\text{NUM} & \text{SG}
\end{array}
\end{array}
\]
C-structure and f-structure

IP
   NP  I'
      N  VP
          V'

David

V

greeted

NP

Chris

[ PRED 'GREET\langleSUBJ,OBJ\rangle' ]

[ SUBJ [ PRED 'DAVID' ] ]

[ OBJ [ PRED 'CHRIS' ] ]
Motivating Constituent Structure

What information does constituent structure represent?
Motivating Constituent Structure

What information does constituent structure represent?

- Represents hierarchical phrasal groupings
Motivating Constituent Structure

What information does constituent structure represent?

- Represents hierarchical phrasal groupings
- Criteria depend on surface syntactic properties, not semantic intuitions or facts about abstract functional syntactic structure
Motivating Constituent Structure

What information does constituent structure represent?

- Represents hierarchical phrasal groupings
- Criteria depend on surface syntactic properties, not semantic intuitions or facts about abstract functional syntactic structure
- Varies greatly across languages
Some theories (GB/Principles and Parameters, NOT LFG): Subjects always appear in the specifier of IP.
Constituent Structure

- Some theories (GB/Principles and Parameters, NOT LFG): Subjects always appear in the specifier of IP.

- LFG does not assume that subjects are defined in terms of phrase structure position, or that subjects must always appear in a particular position in the tree.
Some theories (GB/Principles and Parameters, NOT LFG): Subjects always appear in the specifier of IP.

LFG does not assume that subjects are defined in terms of phrase structure position, or that subjects must always appear in a particular position in the tree.

However, there are structure-function mapping generalisations which state that phrases with particular functions tend to appear in particular phrase structure positions.
Constituent Structure

- Some theories (GB/Principles and Parameters, NOT LFG): Subjects always appear in the specifier of IP.

- LFG does not assume that subjects are defined in terms of phrase structure position, or that subjects must always appear in a particular position in the tree.

- However, there are structure-function mapping generalisations which state that phrases with particular functions tend to appear in particular phrase structure positions.

- In English, the specifier of IP is associated with the subject function; in other languages, it is associated with TOPIC or FOCUS. More below.
Lexical Integrity (Bresnan, 1982): Morphologically complete words are leaves of the c-structure tree, and each leaf corresponds to one and only one c-structure node.
Lexical Integrity (Bresnan, 1982): Morphologically complete words are leaves of the c-structure tree, and each leaf corresponds to one and only one c-structure node.

English: cause to run

Japanese: hasiraseta run.CAUS.PAST

Words in one language can express the same f-structure as phrases in another language: Lexical Integrity holds at c-structure, not f-structure.
Economy of Expression (Bresnan, 2001): All syntactic phrase structure nodes are optional, and are not used unless required by independent principles (completeness, coherence, semantic expressivity).

```
CP
  \   /
  C'  
   \ /
   C  IP
    \ /
    Is NP I'
     \ /
     N  VP
      \ /
      David V
       \ /
       yawning
```
'When was Lermontov born?'
C-structure and f-structure

David greeted Chris

\[
\begin{align*}
\text{pred } & 'greet' \\
\text{subj, obj} & \\
\text{subj} & [\text{pred } 'David'] \\
\text{obj} & [\text{pred } 'Chris'] \\
\end{align*}
\]
C- and F-Structure

\[ \phi \text{ function relates c-structure nodes to f-structures.} \]

(Function: Every c-structure node corresponds to exactly one f-structure.)
Many c-structure nodes can correspond to the same f-structure.
Some f-structures have no corresponding c-structure node.
Some f-structures have no corresponding c-structure node.

These are formal, mathematical facts about the c-structure/f-structure relation. What are the linguistic facts?
C-structure heads are f-structure heads:

\[
\begin{align*}
\text{VP} & \quad \phi \\
\text{V'} & \quad \left[ \text{PRED} \ '\text{GREET}(\text{SUBJ}, \text{OBJ})' \vphantom{\text{SUBJ}} \right] \\
\text{V} & \quad \left[ \text{TENSE} \ '\text{PAST} \vphantom{\text{SUBJ}} \right] \\
greeted & \quad \text{greeted}
\end{align*}
\]
Specifiers are filled by grammaticized discourse functions SUBJ, TOPIC, FOCUS.
Specifiers are filled by grammaticized discourse functions SUBJ, TOPIC, FOCUS.

Specifier of IP in English: SUBJ

\[
\begin{array}{c}
\text{IP} \\
\text{NP} \rightarrow \text{I}' \\
\text{N} \quad \text{VP} \\
David \\
yawned
\end{array}
\]

\[
\begin{array}{c}
\text{PRED} \quad \langle \text{YAWN(SUBJ)} \rangle \\
\text{SUBJ} \\
\text{PRED} \quad \langle \text{DAVID} \rangle
\end{array}
\]
Specifier of IP in Russian: Topic or Focus

Evgenija Onegina
Eugene Onegin

napisal
wrote

Puškin
Pushkin

PRED ‘WRITE⟨OBJ,SUBJ⟩’

TOPIC

SUBJ PRED ‘EUGENE ONEGIN’

OBJ PRED ‘PUSHKIN’
Specifier of IP in Bulgarian: Focus; Specifier of CP: Topic

```
CP
  NP
    C'
      IP
        NP
          N
            Ivan
          I'
            NP
              N
                kakvo
              I
                pravi
                  what
                  does

PRED 'DO⟨SUBJ,OBJ⟩'
  TOPIC [PRED 'IVAN']
  SUBJ [PRED 'WHAT']
  OBJ
```
Specifier of CP in English: Focus

- CP
  - NP
    - N: What
    - C′
      - C: is
        - NP
          - N: David
          - I′
            - NP
              - N: VP
                - V: eating
  - IP
    - PRED: ‘EAT⟨SUBJ,OBJ⟩’
    - FOCUS
      - SUBJ: ‘WHAT’
      - OBJ: ‘DAVID’
Specifier of CP in Finnish: Focus

\[\begin{align*}
\text{Mikolta} & \\
\text{Mikko.ABL} & \\
\text{NP} & \\
\text{N} & \\
\text{C'} & \\
\text{IP} & \\
\text{NP} & \\
\text{N} & \\
\text{I'} & \\
\text{IP} & \\
\text{NP} & \\
\text{N} & \\
\text{I} & \\
\text{VP} & \\
\text{NP} & \\
\text{N} & \\
\text{subj} & \\
\text{OBJ} & \\
\text{OBLsource} & \\
\text{topic} & \\
\text{subjobj} & \\
\text{pred 'Mikko'} & \\
\text{pred 'Anna'} & \\
\text{pred 'flowers'} & \\
\text{get subj, obj, obl} & \\
\end{align*}\]
Complements: Functional Categories

Complement of functional category is f-structure co-head:

```
[ IP
  NP | I'
    | VP
  N | I | V
  David | is | yawning

[ PRED ‘YAWN⟨SUBJ⟩’
  SUBJ [ PRED ‘DAVID’ ]
```
Anna budet čitat’ knigu.

Complements: Functional Categories

- **PRED** ‘READ(SUBJ,OBJ)’
- **TENSE** FUTURE
- **TOPIC** \{ [PRED ‘Anna’] \}
- **SUBJ**
- **OBJ** [PRED ‘BOOK’]
Complements of Lexical Categories

Complement of lexical category is f-structure complement (non-subject argument):

```
[IP
  [NP
    [N David]
    [I' [NP [N greeted] [V' V]]]
    [VP]]
[SUBJ [PRED ‘GREET(SUBJ,OBJ)’]]
[OBJ [PRED ‘Chris’]]
```
Complements of Lexical Categories

```
[IP
  [NP
    [N
      David]
    [I']
  ]
  [VP
    [V
      gave]
    [NP
      [NP
        [N
          Chris]
        [Det
          a]
      ]
      [N'
        book]]
  ]

PRED 'GIVE(SUBJ,OBJ,OBJ-theme)'

SUBJ [PRED 'David']

OBJ [PRED 'Chris']

OBJ-theme [SPEC [PRED 'a']]

PRED 'book'
```
Constraining the c-structure/f-structure correspondence

\[ V' \xrightarrow{\phi} \begin{bmatrix} \text{PRED} & \text{‘YAWN(SUBJ)’} \\ \text{TENSE} & \text{PAST} \end{bmatrix} \]

\text{yawned}
Constraining the c-structure/f-structure correspondence

\[
\phi \quad [\text{PRED} \quad \text{‘YAWN} \langle \text{SUBJ} \rangle \text{’}] \\
\quad \quad [\text{TENSE} \quad \text{PAST}]
\]

\[
yawned
\]

\[
V' \quad \rightarrow \quad V
\]
Local F-Structure Reference

\[ V' \xrightarrow{\phi} \left[ \begin{array}{c} \text{PRED} \quad \text{‘YAWN} \langle \text{SUBJ} \rangle \text{’} \\ \text{TENSE} \quad \text{PAST} \end{array} \right] \]

\text{yawned}

\[ V' \quad \rightarrow \quad V \]

- the current c-structure node ("self"): \( * \)
- the immediately dominating node ("mother"): \( \hat{*} \)
- the c-structure to f-structure function: \( \phi \)
Rule Annotation

\[
\begin{array}{c}
V' \\
\downarrow \phi \\
V \\
vawned
\end{array}
\rightarrow
\begin{array}{c}
PRED \ 'YAWN\langle\text{SUBJ}\rangle' \\
TENSE \ PAST
\end{array}
\]
Rule Annotation

\[
\begin{align*}
V' & \rightarrow V \\
\phi(\ast) &= \phi(\ast) \\
\text{mother’s (V’’s) f-structure} &= \text{self’s (V’s) f-structure}
\end{align*}
\]

\[
\begin{array}{c}
V' \rightarrow V \\
\phi(\ast) = \phi(\ast) \\
\text{mother’s (V’’s) f-structure} = \text{self’s (V’s) f-structure}
\end{array}
\]
Simplifying the Notation

\[ \phi(\hat{*}) \quad \text{(mother’s f-structure)} \quad = \uparrow \]
\[ \phi(*) \quad \text{(self’s f-structure)} \quad = \downarrow \]

\[
\begin{array}{c}
\text{yawned} \\
\phi
\end{array}
\]

\[
\begin{array}{c}
\text{pred 'yawn} \\
\text{(subj)}' \\
\text{tense past}
\end{array}
\]

\[
\text{PRED 'YAWN(SUBJ)'}
\]

\[
\text{TENSE PAST}
\]
Simplifying the Notation

\[
\phi(\hat{*}) \quad (\text{mother’s f-structure}) \quad = \uparrow \\
\phi(*) \quad (\text{self’s f-structure}) \quad = \downarrow
\]

\[
\text{yawned} \\
V' \xrightarrow{\phi} V \\
\text{PRED ‘YAWN(SUBJ)’} \\
\text{TENSE PAST}
\]

\[
V' \rightarrow V \\
\uparrow = \downarrow \\
\text{mother’s f-structure = self’s f-structure}
\]
Using the Notation

\[ V' \rightarrow V \]

\[
\uparrow = \downarrow
\]

mother’s f-structure = self’s f-structure
Using the Notation

\[ V' \rightarrow V \]

\[ \uparrow = \downarrow \]

mother’s f-structure = self’s f-structure
Using the Notation

\[ V' \rightarrow V \]

\[ \uparrow = \downarrow \]

mother’s f-structure = self’s f-structure
Using the Notation

\[ V' \rightarrow V \]

\[ \uparrow = \downarrow \]

mother’s f-structure = self’s f-structure
More rules

\[ V' \rightarrow V \quad NP \]
\[ \phi(\hat{*}) = \phi(*) \quad (\phi(\hat{*}) \text{ OBJ}) = \phi(*) \]

mother's f-structure's OBJ = self's f-structure

In simpler form:

\[ V' \rightarrow V \quad NP \]
\[ \uparrow = \downarrow \quad (\uparrow \text{ OBJ}) = \downarrow \]
Using the Notation

\[ V' \rightarrow V \text{ NP} \]
\[ \uparrow = \downarrow (\uparrow \text{ OBJ}) = \downarrow \]
Using the Notation

\[ V' \rightarrow V \quad NP \quad \uparrow = \downarrow \quad (\uparrow \text{OBJ}) = \downarrow \]
Terminal nodes

\[
\begin{array}{c}
V \\
yawned
\end{array} \rightarrow \left[
\begin{array}{c}
PRED \quad ‘YAWN\langle SUBJ\rangle’ \\
TENSE \quad PAST
\end{array}\right]
\]
Terminal nodes

Expressible as:

\[ V \rightarrow \text{yawned} \]
\[ (\uparrow \text{PRED}) = \text{‘YAWN(SUBJ)’} \]
\[ (\uparrow \text{TENSE}) = \text{PAST} \]
Terminal nodes

Expressible as:

\[
V \rightarrow \ yawned
\]

\[
(\uparrow \text{PRED}) = \text{‘YAWN\langle\text{SUBJ}\rangle’}
\]

\[
(\uparrow \text{TENSE}) = \text{PAST}
\]

Standard form:

\[
yawned \ V \ (\uparrow \text{PRED}) = \text{‘YAWN\langle\text{SUBJ}\rangle’}
\]

\[
(\uparrow \text{TENSE}) = \text{PAST}
\]
Phrase structure rules: English

\[
\text{IP} \rightarrow \left( \text{NP} \quad \left( \uparrow \text{SUBJ} = \downarrow \right) \quad \left( \uparrow = \downarrow \right) \right) \\
\text{I'} \rightarrow \left( \text{I} \quad \left( \uparrow = \downarrow \right) \quad \left( \uparrow = \downarrow \right) \right) \\
\text{VP} \rightarrow \left( \text{V} \quad \left( \uparrow = \downarrow \right) \right) \\
\text{NP} \rightarrow \left( \text{N} \quad \left( \uparrow = \downarrow \right) \right)
\]
Lexical entries: English

\[ \text{yawned} \quad V \quad (↑ \text{ PRED}) = 'YAWN⟨SUBJ⟩' \]
\[ (↑ \text{ TENSE}) = \text{PAST} \]

\[ \text{David} \quad N \quad (↑ \text{ PRED}) = '\text{DAVID}' \]
Lexical entries: English

*yawned*  
\[ \text{V} (\uparrow \text{PRED}) = \text{‘YAWN(SUBJ)’} \]  
\[ (\uparrow \text{TENSE}) = \text{PAST} \]

*David*  
\[ \text{N} (\uparrow \text{PRED}) = \text{‘DAVID’} \]

(Standard LFG practice: include only features relevant for analysis under discussion.)
Analysis: English

```
IP
  NP
    ↑ subj = ↓
    ↑ = ↓
    ↓
    N
    ↑ = ↓
    ↓
    VP
    ↑ = ↓
    ↓
    V
    ↑ = ↓
    ↓
    yawned
    ↑ pred = ‘YAWN⟨SUBJ⟩’
    ↑ tense = PAST
```
Analysis: English

```
IP
   NP     I'
      ↑ = ↓    ↑ = ↓
      ↑ = ↓    ↑ = ↓
       N       VP
       ↑ = ↓    ↑ = ↓
      David    V
             ↑ = ↓
             ↑ PRED = 'YAWN(SUBJ)'
             (↑ TENSE) = PAST
```
Analysis: English

```
IP
  NP  I'
(↑ subj) = ↓  ↑ = ↓
   /
  N  VP
fun = fn  ↑ = ↓
   /
David  V
(fn pred) = ‘DAVID’  ↑ = ↓
   /
yawned
(↑ pred) = ‘YAWN⟨subj⟩’
(↑ tense) = past
```
Analysis: English

```
IP
   NP  I'
   (f_{ip} \text{ subj}) = f_{np}  \uparrow = \downarrow
   |   |  
   N  VP
   f_{np} = f_{n}  \uparrow = \downarrow
   |   |  
   David  V
   (f_{n} \text{ pred}) = \text{‘David’}  \uparrow = \downarrow
   |  
   yawned
   (\uparrow \text{ pred}) = \text{‘YAWN(SUBJ)’}
   (\uparrow \text{ tense}) = \text{ Past}
```
Analysis: English

\[
\begin{align*}
\text{IP} & \\
\text{NP} & \quad \text{I'} \\
& \quad \quad \quad (f_{ip \text{ SUBJ}}) = f_{np} \\
& \quad \quad \quad \quad \quad \uparrow = \downarrow \\
& \quad \quad \quad \quad \quad | \\
& \quad \quad \quad \quad \quad \text{N} \\
& \quad \quad \quad \quad \quad (f_{np} = f_n) \\
& \quad \quad \quad \quad \quad \quad \uparrow = \downarrow \\
& \quad \quad \quad \quad \quad | \\
& \quad \quad \quad \quad \quad \text{David} \\
& \quad \quad \quad \quad \quad (f_n \text{ PRED}) = 'DAVID' \\
& \quad \quad \quad \quad \quad \quad \uparrow = \downarrow \\
& \quad \quad \quad \quad \quad | \\
& \quad \quad \quad \quad \quad \text{V} \\
& \quad \quad \quad \quad \quad \text{yawned} \\
& \quad \quad \quad \quad \quad (f_v \text{ PRED}) = 'YAWN(SUBJ)' \\
& \quad \quad \quad \quad \quad \quad (f_v \text{ TENSE}) = \text{PAST}
\end{align*}
\]
IP
   /\   \\
 NP  I'  \\
  / \   |
 (f_{ip} \text{ SUBJ}) = f_{np} \quad \uparrow = \downarrow \\
 |   |
 N   VP  \\
  / \   |
 f_{np} = f_{n} \quad \uparrow = \downarrow \\
 |   |
 David  V  \\
  / \   |
 (f_{n} \text{ PRED}) = 'DAVID' \quad f_{vp} = f_{v} \\
 |   |
 yawned  \\
  /   |
 (f_{v} \text{ PRED}) = 'YAWN(SUBJ)' \\
 (f_{v} \text{ TENSE}) = \text{PAST}
Analysis: English

```
IP
  NP       I'
(f_{ip} \text{ SUBJ}) = f_{np}
  \uparrow = \downarrow
    N       VP
(f_{np} = f_n)
    David
(f_n \text{ PRED}) = 'DAVID'
    V
(f_{vp} = f_v)
    yawned
(f_v \text{ PRED}) = 'YAWN\langle\text{SUBJ}\rangle'
(f_v \text{ TENSE}) = \text{PAST}
```
\[
\begin{align*}
\text{IP} & \\
\text{NP} & \quad \text{I'} \\
(f_{ip} \text{ SUBJ}) & = f_{np} & f_{ip} & = f'_i \\
\mid & & \mid \\
\text{N} & \quad \text{VP} \\
(f_{np} & = f_n & f'_i & = f_{vp} \\
\mid & & \mid \\
\text{David} & \quad \text{V} \\
(f_n \text{ PRED}) & = \text{‘DAVID’} & f_{vp} & = f_v \\
\mid & & \mid \\
\text{yawned} & \quad \text{yawned} \\
(f_v \text{ PRED}) & = \text{‘YAWN(SUBJ)’} & (f_v \text{ TENSE}) & = \text{PAST}
\end{align*}
\]
Solving the Description

\[
\begin{align*}
(f_{ip} \text{ SUBJ}) &= f_{np} \\
  f_{np} &= f_n \\
  (f_n \text{ PRED}) &= \text{'David'} \\
  f_{ip} &= f_i' \\
  f_i' &= f_{vp} \\
  f_{vp} &= f_v \\
  (f_v \text{ PRED}) &= \text{'YAWN\langle\text{SUBJ}\rangle'} \\
  (f_v \text{ TENSE}) &= \text{PAST}
\end{align*}
\]

\[
\begin{align*}
  &f_{ip} \begin{bmatrix} \text{PRED} & \text{'YAWN\langle\text{SUBJ}\rangle'} \\ \text{TENSE} & \text{PAST} \end{bmatrix} \\
  &f_{i'} \begin{bmatrix} \text{SUBJ} & f_{np} \begin{bmatrix} \text{PRED} & \text{'David'} \end{bmatrix} \end{bmatrix}
\end{align*}
\]
Final result

(\text{f}_{ip}\ \text{subj}) = f_{np} \quad f_{ip} = f_{i'}

\text{David} \quad f_{np} = f_{n} \quad f_{i'} = f_{vp} \quad f_{vp} = f_{v} \quad (f_{v}\ \text{pred}) = \text{YAWN}(<\text{subj}>)

\text{yawned} \quad (f_{v}\ \text{tense}) = \text{past}

PRED 'YAWN (\text{subj})'
TENSE PAST
SUBJ \quad [\text{PRED 'David']}

Lexical Functional Grammar – 70 / 80
Warlpiri

\[ GF \equiv \{\text{SUBJ} \mid \text{OBJ} \mid \text{OBL}_\theta\} \]

\[
\begin{align*}
\text{IP} & \quad \rightarrow \quad \left( \begin{array}{c}
\text{NP} \\
(\uparrow \text{FOCUS}) = \downarrow \\
(\uparrow \text{GF}) = \downarrow 
\end{array} \right) \left( \begin{array}{c}
\text{I}' \\
(\uparrow = \downarrow) \\
\end{array} \right) \\
\text{I'} & \quad \rightarrow \quad \left( \begin{array}{c}
\text{I} \\
(\uparrow = \downarrow) \\
\end{array} \right) \left( \begin{array}{c}
\text{S} \\
(\uparrow = \downarrow) \\
\end{array} \right) \\
\text{S} & \quad \rightarrow \quad \{ \text{NP} \mid \text{V} \}^* \\
& \quad \rightarrow \quad \left( \begin{array}{c}
(\uparrow \text{GF}') = \downarrow \\
\uparrow = \downarrow 
\end{array} \right)
\end{align*}
\]
Warlpiri verbs

\[
panti-rni \ V \quad (↑ PRED) = 'SPEAR\langle SUBJ,OBJ\rangle' \\
(↑ SUBJ PRED) = 'PRO' \\
(↑ SUBJ CASE) = ERG \\
(↑ OBJ PRED) = 'PRO' \\
(↑ OBJ CASE) = ABS
\]
The image contains a diagram of a Warlpiri sentence in Lexical Functional Grammar (LFG). The sentence is "ngarrka-ngku man-erg 'man' panti-rni spear-nonpast wawirri kangaroo.abs "kangaroo.

The diagram shows the syntactic structure of the sentence, with IP (ip), NP (np), V (v), S (s), SUBJ (subj), and OBJ (obj) nodes. The focus is on the predication structure with 'spear⟨subj, obj⟩' and 'man' as the subject and object, respectively. The case relations are indicated with arrows, showing ergative case for the subject and absolutive case for the object.
Chichewa

\[
S \rightarrow \left( \begin{array}{c}
\text{NP} \\
(\uparrow \text{SUBJ}) = \downarrow
\end{array} \right), \quad \left( \begin{array}{c}
\text{NP} \\
(\uparrow \text{TOPIC}) = \downarrow
\end{array} \right), \quad \left( \begin{array}{c}
\text{VP} \\
\uparrow = \downarrow
\end{array} \right)
\]

\[
\text{VP} \rightarrow \left( \begin{array}{c}
\text{V'} \\
\uparrow = \downarrow
\end{array} \right)
\]

\[
\text{V'} \rightarrow \left( \begin{array}{c}
\text{V} \\
\uparrow = \downarrow
\end{array} \right) \left( \begin{array}{c}
\text{NP} \\
(\uparrow \text{OBJ}) = \downarrow
\end{array} \right)
\]

Comma between daughters in S rule: daughters of S are unordered
Chichewa verbs

zi-ná-wá-lum-a  V  (↑ PRED) = ‘BITE<SUBJ,OBJ>’
((↑ SUBJ PRED) = ‘PRO’)
(↑ SUBJ NOUNCLASS) = 10
(↑ OBJ PRED) = ‘PRO’
(↑ OBJ NOUNCLASS) = 2
Chichewa

S

NP

(↑ subj) = ↓

njûchi
bees

(↑ pred) = 'bees'

(↑ nounclass) = 10

VP

↑ =↓

V′

↑ =↓

V

zi-ná-wá-lum-a

SUBJ-PAST-OBJ-bite-INDICATIVE

(↑ pred) = 'BITE\langle subj, obj\rangle'

((↑ subj pred) = 'pro')

(↑ subj nounclass) = 10

(↑ obj pred) = 'pro'

(↑ obj nounclass) = 2

PRED

'BITE\langle subj, obj\rangle'

SUBJ

PRED 'bees'

NOUNCLASS 10

OBJ

PRED 'pro'

NOUNCLASS 2
Chichewa

zi-ná-lum-a

SUBJ-PAST-OBJ-bite-INDICATIVE

(↑ PRED) = ‘BITE(SUBJ,OBJ)’

((↑ SUBJ PRED) = ‘PRO’)

(↑ SUBJ NOUNCLASS) = 10

alenje

hunters

(↑ PRED) = ‘HUNTER’

(↑ NOUNCLASS) = 10
For more information

- For more on LFG, visit the LFG website: http://www.essex.ac.uk/linguistics/LFG/


- SOAS, Essex, and Oxford hold student-oriented meetings each term for discussion of issues in LFG, including student presentations: http://se-lfg.tk/
References


