LFG

Syntactic Theory
Winter Semester 2009/2010

Antske Fokkens

Department of Computational Linguistics
Saarland University

17 November 2009

Outline

1 Introduction

2 F-structures
- Motivation
- Formal properties of f-structures
- grammatical functions in LFG
- well-formedness conditions

Outline

1 Introduction

2 F-structures
- Motivation
- Formal properties of f-structures
- grammatical functions in LFG
- well-formedness conditions

Lexical Functional Grammar, Introduction

- Developed in the late 70s by Joan Bresnan and Ron Kaplan
- LFG brings scholars from different fields together:
  - Theoretical linguists
  - Descriptive, typological linguists
  - Computational linguistics
- Main ideas:
  - A formal system to model human speech (fits in the tradition of generative grammar)
  - Psychological plausibility: the formalism should be able to represent a native speaker's syntactic knowledge appropriately
  - Strong typological basis: analyses should capture cross-linguistic similarities

Main levels of representation

A Lexical Functional Grammar represents expressions in (minimally) two levels of representation:

- **constituent structure** (c-structure):
  - a tree which represents phrase structure configurations
  - it indicates the superficial arrangements of the words in the sentence, i.e. it serves as an input for the phonological interpretation of the string
  - languages differ radically on a c-structure level
- **functional structure** (f-structure):
  - an attribute-value matrix represents surface grammatical functions, i.e. traditional syntactic relations such as subject, object, complement and adjunct
  - It serves as the sole input to the semantic component
  - languages are similar on a f-structure level

Lexical Functional Grammar

- LFG is **lexical** because of the assumption that words and lexical items are as important in providing grammatical information as syntactic elements
- LFG is **functional** because grammatical information is represented by lexical functions (f-structure), rather than by phrase structure configurations

  i.e. LFG is **nonconfigurational**

Orginizations of the coming lectures

- An overview of the architecture of LFG
  - F-structures: formal definition and basic properties
  - C-structures: basic properties
  - Mapping between c- and f-structures
  - Example analysis
- Phenomena and constraints in LFG
  - How to integrate and use constraints in LFG analyses
  - Some basic phenomena and their analyses in LFG

Outline

1 Introduction

2 F-structures
- Motivation
- Formal properties of f-structures
- grammatical functions in LFG
- well-formedness conditions

F-structure: motivation

- Assumption: for any language functional syntactic concepts such as subject and object are relevant
- The f-structure can represent what languages have in common in wide-spread phenomena, no matter how radically different languages may be on the surface

  e.g. passives
- The f-structure can capture some universal properties of language

  e.g. the Keenan-Comrie Hierarchy for relative clauses:
  SUBJ > DOBJ > IOBJ > OBL > GEN > OCOMP
  - A language may sets its border for acceptable and unacceptable relative clauses anywhere on the hierarchy: those elements above the boundary can be relativized.
  - Processing becomes more difficult when going down the hierarchy

Examples of relative clauses

- **Subject**: That's the man [who ran away]. The girl [who came late] is my sister.
- **Direct object**: That's the man [I saw yesterday]. The girl [Kate saw] is my sister.
- **Indirect object**: That's the man [to whom I gave the letter]. The girl [who I wrote a letter to] is my sister.
- **Oblique**: That's the man [I was talking about]. The girl [who I sat next to] is my sister.
- **Genitive**: That's the man [whose sister I know]. The girl [whose father died] told me she was sad.
- **Obj of Comp**: That's the man [I am taller than]. The girl [who Kate is smarter than] is my sister.

An example of an F-structure

- Example: the f-structure of *I saw the girl*:

$$
\begin{bmatrix}
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'pro'} \\ \text{PERS} & 1 \\ \text{NUM} & \text{SG} \end{bmatrix} \\
\text{TENSE} & \text{PAST} \\
\text{PRED} & \text{'see}\left\langle (\uparrow\text{SUBJ}),(\uparrow\text{OBJ}) \right\rangle\text{'} \\
\text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{DEF} & + \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{bmatrix}
\end{bmatrix}
$$

Formal properties of F-structures

- An F-structure is a finite set of pairs of attributes and values
- An F-structures attributes may be
  - A: atomic symbols, e.g. SUBJ, OBJ, PRED
- An F-structures values may be:
  - A: atomic symbols, e.g. SG, 1, +, PAST
  - S: semantic forms, e.g. ’girl’, ’see$<$($\uparrow$SUBJ)($\uparrow$ OBJ)$>$’
  - F: f-structures
- F-structures are defined by the following recursive domain equation:

  $F = (A \rightarrow_f F \cup A \cup S)$

Examples of simple F-structures

- $$f: \begin{bmatrix} \text{PRED 'David'} \\ \text{NUM SG} \end{bmatrix}$$

  Description:
  (f PRED) = 'David'
  (f NUM) = SG

- $$g: \begin{bmatrix} \text{PRED 'yawn(SUBJ)'} \\ \text{TENSE PAST} \\ \text{SUBJ } f \begin{bmatrix} \text{PRED 'David'} \\ \text{NUM SG} \end{bmatrix} \end{bmatrix}$$

  Description:
  (g PRED) = 'yawn(SUBJ)'
  (g TENSE) = PAST
  (g SUBJ) = f

A Functional structure

- Mathematically, the f-structure can be is seen as a function from attributes to values, hence its name
- A function assigns a unique value to its argument
- In other words:
  - if (f q) = t and (f q) = v, then t = v
  - *attr → v1, *attr → v2 $\quad v1 \neq v2$
- The value of an attribute can be a set:

  (We'll see more examples later)
  - $\begin{bmatrix} \text{attr1} & \text{v1} \\ \text{attr2} & \{\text{v2,v3}\} \end{bmatrix}$ e.g. *we*: $\begin{bmatrix} \text{PRED} & \text{'pro'} \\ \text{PERS} & \{\text{H,S}\} \\ \text{NUM} & \text{PL} \end{bmatrix}$

symbols and semantic forms

- Symbols are unbroken strings of alphanumeric characters $\longrightarrow$ the choice of symbols belongs to a particular theory of linguistics
- Semantic forms are special: the single quotes around semantic form values indicate that this form is unique. E.g. each instance of the word *girl* is a uniquely instantiated occurrence of the semantic form 'girl'

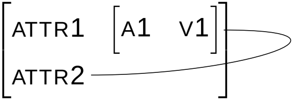
Some Linguistic terminology (Bresnan 1982)

- an attribute-value pair where the value is a symbol is called a **feature**
- an attribute-value pair where the value is an f-structure is called a **grammatical function**
- an attribute whose value is a semantic form is called a **semantic feature**

Attributes with the same values

- Two attributes within the same f-structure can have the same value
- This can be represented in several ways:

$$\begin{bmatrix} \text{ATTR1} & \begin{bmatrix} \text{A1} & \text{V1} \end{bmatrix} \\ \text{ATTR2} & \begin{bmatrix} \text{A1} & \text{V1} \end{bmatrix} \end{bmatrix}$$

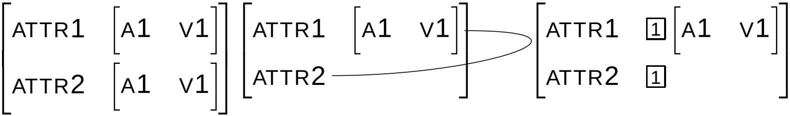


$$\begin{bmatrix} \text{ATTR1} & \boxed{1} \begin{bmatrix} \text{A1} & \text{V1} \end{bmatrix} \\ \text{ATTR2} & \boxed{1} \end{bmatrix}$$

- Note:
  - Semantic forms are unique: two instances of 'lion' in a sentence does not necessarily mean two attributes have the same value: co-indexation is required

Attributes with the same values

- Two attributes within the same f-structure can have the same value
- This can be represented in several ways:



- Note:
  - Semantic forms are unique: two instances of 'lion' in a sentence does not necessarily mean two attributes have the same value: co-indexation is required
  - Identity in LFG differs from identity in HPSG: no type/token distinction!

Grammatical functions in LFG

LFG proposes the following inventory of grammatical functions, which is universally available:

- SUBJect
- OBJect
- OBJ$_\theta$
- COMP
- XCOMP
- OBLique$_\theta$
- ADJunct
- XADJunct

Cross-classification of grammatical functions

Several cross-classifications are possible among grammatical functions:

- Governable functions: SUBJ, OBJ, XCOMP, COMP, OBJ_θ, OBL_θ
  Modifiers: ADJ, XADJ
  - Core arguments/terms: SUBJ, OBJ, OBJ_θ
    Non-term/oblique functions: OBL_θ
  - Semantically unrestricted functions: SUBJ, OBJ
    Semantically restricted functions: OBJ_θ, OBL_θ
- Open functions: XCOMP, XADJ
  Closed functions: SUBJ, OBJ, COMP, OBJ_θ, OBL_θ, ADJ

$\rightarrow$ we will only consider the distinction between governable functions and modifiers for now

Governable grammatical functions

- SUBJ, OBJ, XCOMP, COMP, OBJ_θ and OBL_θ are *governed* or *subcategorized for* by the predicate, hence the name **governable grammatical functions**
- ADJ and XADJ modify the phrase they appear in, but they are not subcategorized for by the predicate. The term **modifiers** applies to these functions

The value of ADJ and XADJ

- In principle, there is no limit to the number of modifiers that can appear within a phrase: the value of the ADJ or XADJ feature is the set of all modifiers that are present, e.g. *David yawned quietly* (*yesterday*):

$$\left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{'David'}\end{array}\right] \\ \text{PRED} & \text{'yawn}<(\uparrow \text{SUBJ})>\text{'} \\ \text{ADJ} & \left\{\left[\begin{array}{ll}\text{PRED} & \text{'quietly'}\end{array}\right]\right\} \end{array}\right] \left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{'David'}\end{array}\right] \\ \text{PRED} & \text{'yawn}<(\uparrow \text{SUBJ})>\text{'} \\ \text{ADJ} & \left\{\begin{array}{l}\left[\begin{array}{ll}\text{PRED} & \text{'quietly'}\end{array}\right] \\ \left[\begin{array}{ll}\text{PRED} & \text{'yesterday'}\end{array}\right]\end{array}\right\} \end{array}\right]$$

- Typically, the values of governable functions are not sets

Identifying governable grammatical functions I

- Dowty (1982) proposes the following tests to distinguish between governable functions and modifiers
  - *Entailment test*: does the predicate entail existence of the argument?
    but:
    - many predicates entail time and place
    - predicates such as *seek* don't entail existence of their arguments, the same holds for semantically empty arguments such as *it* in *it rains*
  - *Subcategorization test*: modifiers can be omitted, arguments cannot
    but:
    - Some verbs have optional arguments (or ambiguous subcategorization frames), such as *eat*
    - In pro-drop languages arguments can generally be dropped

Identifying governable grammatical functions II

- These tests provide good indications for the governable function/modifier distinction, but cannot always correctly differentiate between arguments and modifiers

Some additional tests (1/2)

- *Multiple occurrence*: (Kaplan and Bresnan 1982): modifiers may be multiple specified, arguments cannot:

  The girl saw the baby <u>on Tuesday</u> <u>in the morning</u>

  * David saw <u>Tony</u> <u>George</u> <u>Sally</u>

- *Order dependence*: (Pollard and Sag 1987) relative order of modifiers may change truth-conditions, this is not the case for arguments
  - Kim jogged <u>for twenty minutes</u> <u>twice a day</u>
  - Kim jogged <u>twice a day</u> <u>for twenty years</u>

Some additional tests (2/2)

- *Anaphoric binding*: (Hellan 1988, Dalrymple 1993, for Norwegian)

(1) Jon fortalte meg om seg selv.
Jon told me about self
"Jon told me about himself"

(2) * Hun kastet meg fra seg selv
she threw me from self
"she threw me away from herself"

→ Languages may provide different kind of evidence for such distinctions

Subcategorization

- A semantic form may contain an argument list, next to its semantic predicate name, e.g.
  - 'yawn$<$($\uparrow$ SUBJ)$>$'
  - 'see$<$($\uparrow$ SUBJ), ($\uparrow$ OBJ)$>$'
  - 'give$<$($\uparrow$ SUBJ), ($\uparrow$ OBJ), ($\uparrow$ OBJ2)$>$'
- Note that lexical items select for grammatical functions (not for NPs, CP, etc)
- How to make sure that subcategorization requirements are fulfilled?
  $\rightarrow$ well-formedness constraints on the f-structure: completeness and coherence

Principle of completeness

- The principle of completeness requires that all governable functions present in the argument list of a semantic form must be present in the f-structure
- This excludes ungrammatical expressions such as

* He devoured

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'pro'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{pred} & \text{'devour}<(\uparrow\text{SUBJ}),(\uparrow\text{OBJ})>\text{'} \end{bmatrix}$$

$\longrightarrow$ the object is missing: incomplete f-structure!

Principle of Completeness: 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

- The principle of coherence requires that all governable functions present in the f-structure are also present in the argument list of the predicate
- This excludes ungrammatical examples such as

  * David yawned the flower

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'David'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'flower'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{PRED} & \text{'yawn}<(\uparrow \text{SUBJ})>\text{'} \end{bmatrix}$$

$\longrightarrow$ the OBJ *the flower* is not governed by the predicate: incoherent f-structure!

Principle of Coherence: 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

Principle of Consistency (uniqueness)

- The principle of consistency states what we have already seen in the f-structures formal properties: an attribute has a unique value
- It excludes ungrammatical examples such as

  * David sleep

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'David'} \\ \text{NUM} & \text{SG/PL} \end{bmatrix} \\ \text{PRED} & \text{'yawn}<(\uparrow \text{SUBJ})>\text{'} \end{bmatrix}$$

$\longrightarrow$ 'David' is singular, but the verb form states that the subject's number is plural: inconsistent f-structure!

definition: An f-structure is consistent iff all attributes have at most one value

F-structures, recap I

- F-structures represent the grammatical relations of expressions
- Languages are similar on this level: allows to explain cross-linguistic properties of phenomena
- Formally, an f-structure is a set of attribute-value pairs
- LFG posits a universal inventory of grammatical functions (where we distinguish governable functions and modifiers (among other properties))
- F-structures must be
  - complete
  - coherent
  - consistent

Bibliography I

- Bresnan, Joan (2000). *Lexical Functional Syntax*. Blackwell Publishers: Malden, USA/Oxford UK.
- Dalrymple, Mary, Ron M. Kaplan, John T. Maxwell III and Annie Zaenen (eds.). (1995) *Formal Issues in Lexical-Functional Grammar*. CSLI Publications: Palo Alto, USA.
- Dalrymple, Mary (2001). *Lexical Functional Grammar*. Academic Press: San Diego, USA/London, UK.
- Kaplan, Ron (1995). The formal architecture of Lexical-Functional Grammar. In: Dalrymple et al. (1995).
- Kordoni, Valia (2008a). Syntactic Theory Lectures 5. Course slides.
- Schneider, Gerold (1998). *A Linguistic Comparison of Constituency, Dependency and Link Grammar. Lizentiatsarbeit, Institut für Informatik der Universität Zürich*. http://www.ifi.unizh.ch/cl/study/lizarbeiten/lizgerold.pdf.