Syntactic Theory

Lecture 5a (10.12.2010)

PD Dr.Valia Kordoni

Email: kordoni@coli.uni-sb.de

http://www.coli.uni-saarland.de/~kordoni/courses/syntactic-theory/2010/

Lexical-Functional Grammar (LFG)

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.)

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

  SUBJ > DO > IO > OBL > GEN > OCOMP

Grammatical functions

LFG inventory: SUBJect, OBJect, OBJ_θ, COMP, XCOMP, OBLique_θ, ADJunct, XADJunct

- Governable functions: SUBJ, OBJ, OBJ_θ, COMP, XCOMP, OBL_θ
- Terms (core functions): SUBJ, OBJ, OBJ_θ (agreement, anaphora, control)
- Semantically restricted: OBJ_θ, OBL_θ
- Open clausal functions (no internal subject): XCOMP, XADJ

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,$\text{OBJ}_{\text{theme}}$>'
  - *eat*: PRED 'EAT<SUBJ>OBJ'

F-structure representation: Simple F-structures

F-structure is a function from attributes to values – also can be viewed as sets of pairs

- For the proper noun *David*, PRED and NUM are attributes; 'DAVID' and SG are the corresponding values

(1) $\begin{bmatrix} \text{PRED} & \text{'DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix}$

- F-structures within f-structures: *David yawned*

(2) $\begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix}$

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.

(3)
$$\begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{ADJ} & \left\{ \begin{bmatrix} \text{PRED} & \text{'quietly'} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \text{'yesterday'} \end{bmatrix} \right\} \end{bmatrix}$$

F-structure representation: Sets

Sets can also have additional properties, i.e. have attributes and values which apply over the whole set—*hybrid object*

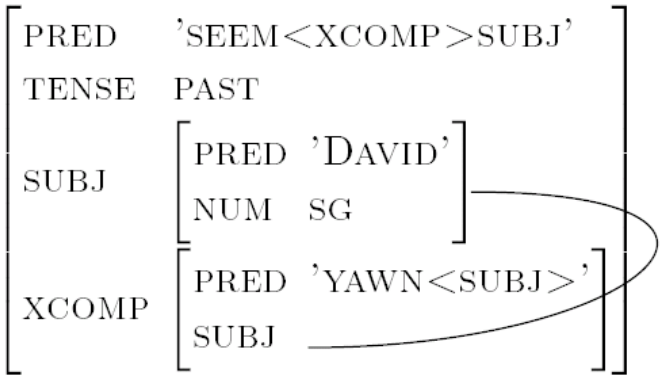
David and Chris yawned.

(4)
$$\begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \left\{ \begin{bmatrix} \text{PRED} & \text{'DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix}, \begin{bmatrix} \text{PRED} & \text{'CHRIS'} \\ \text{NUM} & \text{SG} \end{bmatrix} \right\} \end{bmatrix}$$

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*:

(5) More traditional way:



Attributes with Common Values (cont.)

(6) More HPSG-like:

$$\begin{bmatrix} \text{PRED} & \text{'SEEM<XCOMP>SUBJ'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \boxed{1}\begin{bmatrix} \text{PRED} & \text{'DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{SUBJ} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

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<SUBJ,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.
$$\begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'DAVID'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'SINK'} \end{bmatrix} \end{bmatrix}$$

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 yawn.

b.
$$\begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'BOYS'} \\ \text{NUM} & \text{SG/PL} \end{bmatrix} \end{bmatrix}$$

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

Constraining f-structures

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

F-description with a single equation:

(10) $(g\ \textsc{num}) = \textsc{sg}$

Different f-structures which satisfy this f-description:

(11) a. $\begin{bmatrix} \textsc{num} & \textsc{sg} \end{bmatrix}$

b. $\begin{bmatrix} \textsc{pred} & \text{'Charlie'} \\ \textsc{gend} & \textsc{masc} \\ \textsc{num} & \textsc{sg} \end{bmatrix}$

Functional Constraints (formally)

(12) $(f\,a) = v$ 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

Can also specify longer paths in the f-structure, e.g., $(g\ \text{SUBJ}\ \text{NUM}) = \text{SG}$

Functional constraints example

Lexical constraints:

- *John*
  - $(g\ \text{PRED}) = \text{'JOHN'}$
  - $(g\ \text{NUM}) = \text{SG}$
- *runs*
  - $(f\ \text{PRED}) = \text{'RUN<SUBJ>'}$
  - $(f\ \text{SUBJ CASE}) = \text{NOM}$
  - $(f\ \text{SUBJ NUM}) = \text{SG}$

Phrasal constraints (more on this later):

- $(f\ \text{SUBJ}) = g$

Functional constraints example (cont.)

Combining lexical and phrasal constraints, we have:

- $(f\ \text{SUBJ}) = g$
- $(g\ \text{PRED}) = \text{'JOHN'}$
- $(g\ \text{NUM}) = \text{SG}$
- $(f\ \text{PRED}) = \text{'RUN<SUBJ>'}$
- $(g\ \text{CASE}) = \text{NOM}$
- $(g\ \text{NUM}) = \text{SG}$

Functional constraints example (cont.)

Minimal solution:

(13) $$f:\begin{bmatrix} \text{PRED} & \text{'RUN<SUBJ>'} \\ \text{SUBJ} & g:\begin{bmatrix} \text{PRED} & \text{'JOHN'} \\ \text{CASE} & \text{NOM} \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix}$$

More functional constraints

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

- Disjunction
- Optionality
- Negation
- Existential Constraints

Disjunction

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

(14) I met/have met him.

- Lexical entry for *met*:

- $(f\ \text{PRED}) = \text{'MEET<SUBJ,OBJ>'}$
  $\{(f\ \text{TENSE}) = \text{PAST} \mid \{(f\ \text{FORM}) = \text{PASTPART}\}$

Optionality

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

(15) a. Juan vió a Pedro.
Juan saw PREP Pedro
'Juan saw Pedro.'

b. Juan lo vió.
Juan him.ACC.SG.CLITIC saw
'Juan saw him.'

c. lo Juan vió a Pedro.
Juan him.ACC.SG.CLITIC saw PREP Pedro
'Juan saw Pedro.'

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

Optionality (cont.)

The lexical entries, showing that *lo* is optional:

- *Pedro* N $(f\ \text{PRED}) = \text{'PEDRO'}$
- *lo* Pro $((f\ \text{PRED}) = \text{'PRO'})$

Negation

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

(16) 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* 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.

(17) a. The man who yawns/yawned/will yawn.
b. *The man who yawning.

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

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

The nature of 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:

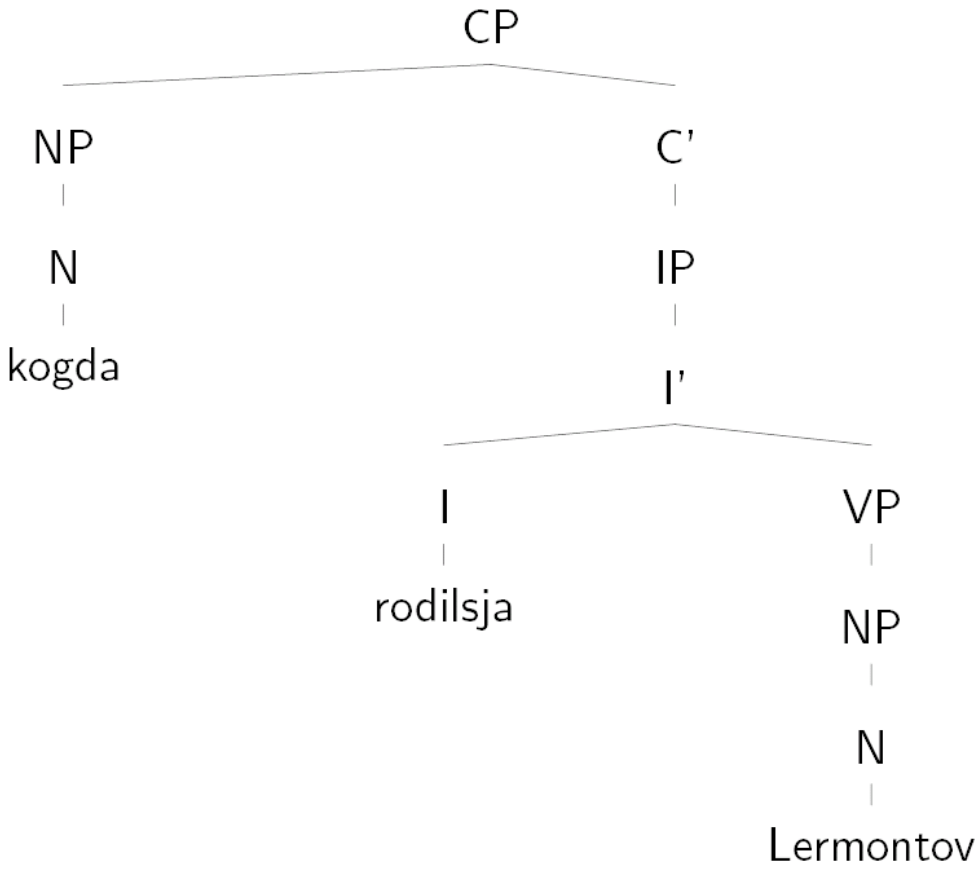
- Endocentric category S: has no lexical head (for "internal subject" languages)
- Optionality: all constituent structure positions are optional

Example of c-structure

(18) kogda rodilsja Lermontov?
when born Lermontov
'When was Lermontov born?'

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

Example c-structure



C-structure rules

Like phrase structure rules, but

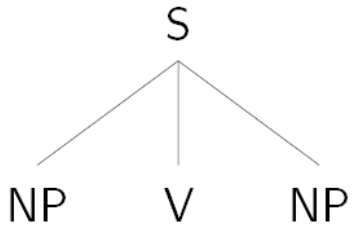
- interpreted as node admissibility conditions, i.e., trees must meet PSR descriptions
- allow for *regular expressions* (Kleene star, disjunction, optionality, etc.) on the right-hand side.

Metacategories

A **metacategory** represents several different sets of categories

(19) a. XP $\equiv$ {NP | PP | VP | AP | AdvP}
b. VP $\equiv$ V NP

Note that using the metacategory VP given in (19b) in the rule S $\rightarrow$ NP VP results in the following tree:



ID/LP Rules

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

(20) No LP rules:

a. VP $\rightarrow$ V, NP

b. VP $\rightarrow$ {V NP | NP V}

(21) One LP rule:

a. VP $\rightarrow$ V, NP V $<$ NP

b. VP $\rightarrow$ V NP

(22) Interacting LP rules:

a. VP $\rightarrow$ V, NP, PP V $<$ NP, V $<$ PP

b. VP $\rightarrow$ {V NP PP | V PP NP}

How a string is licensed

- A context-free c-structure grammar licenses the c-structure of a string.
- The grammar is augmented with functional descriptions, which map the c-structure to an f-structure representation.

Relation of c-structure to f-structure

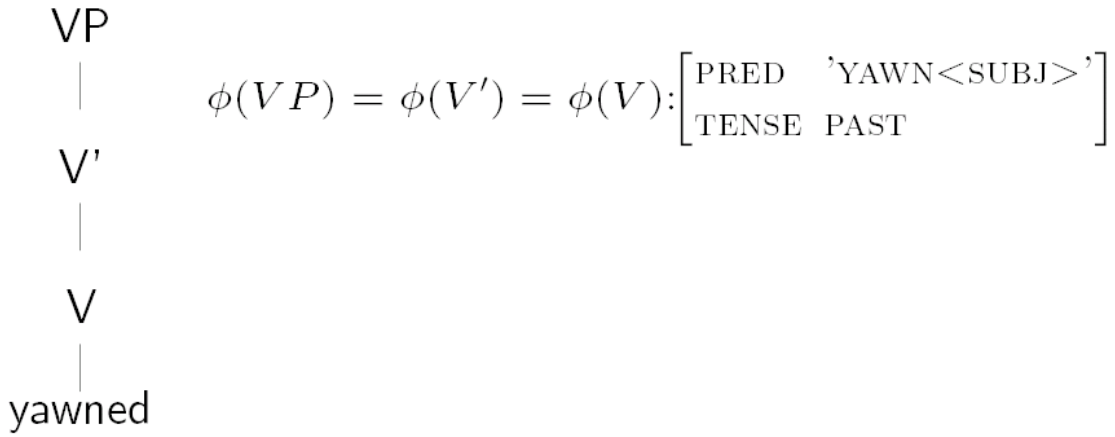
A function ϕ maps the c-structure to the f-structure of a sentence.

- Function: each c-structure related to only one f-structure (but not necessarily vice versa)

V
|
yawned $\quad \phi(V): \begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{TENSE} & \text{PAST} \end{bmatrix}$

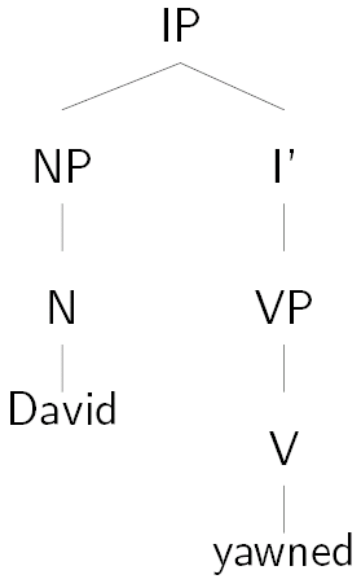
The Head Convention

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



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)



$$\phi(IP): \begin{bmatrix} \text{PRED} & \text{'YAWN<SUBJ>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'DAVID'} \end{bmatrix} \end{bmatrix}$$

$$\phi(NP): \begin{bmatrix} \text{PRED} & \text{'DAVID'} \end{bmatrix}$$

Alternate Notation

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

(23) IP → XP (↑SUBJ) = ↓ I' ↑ = ↓

- 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

(24) V' → V NP
$\uparrow = \downarrow$ $(\uparrow \text{OBJ}) = \downarrow$

(25) VP → V NP NP
$\uparrow = \downarrow$ $(\uparrow \text{OBJ}) = \downarrow$ $(\uparrow \text{SECOBJ}) = \downarrow$

(26) VP → V NP PP
$\uparrow = \downarrow$ $(\uparrow \text{OBJ}) = \downarrow$ $(\uparrow \text{OBJ2}) = \downarrow$
$(\uparrow \text{PFORM}) = \text{TO}$

Lexical Entries

Can use the same notation to express lexical entries

(27) a. *yawned* V $(\uparrow \text{PRED}) = \text{'YAWN<SUBJ>'}$
$(\uparrow \text{TENSE}) = \text{PAST}$

b. *David* N $(\uparrow \text{PRED}) = \text{'DAVID'}$

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

Example of *David yawned*

Work out how we get the final f-structure for *David yawned*.

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

(based on Kaplan and Bresnan 1995)

(28) a. $S \rightarrow \underset{(\uparrow \text{SUBJ}) = \downarrow}{NP} \quad \underset{\uparrow = \downarrow}{VP}$

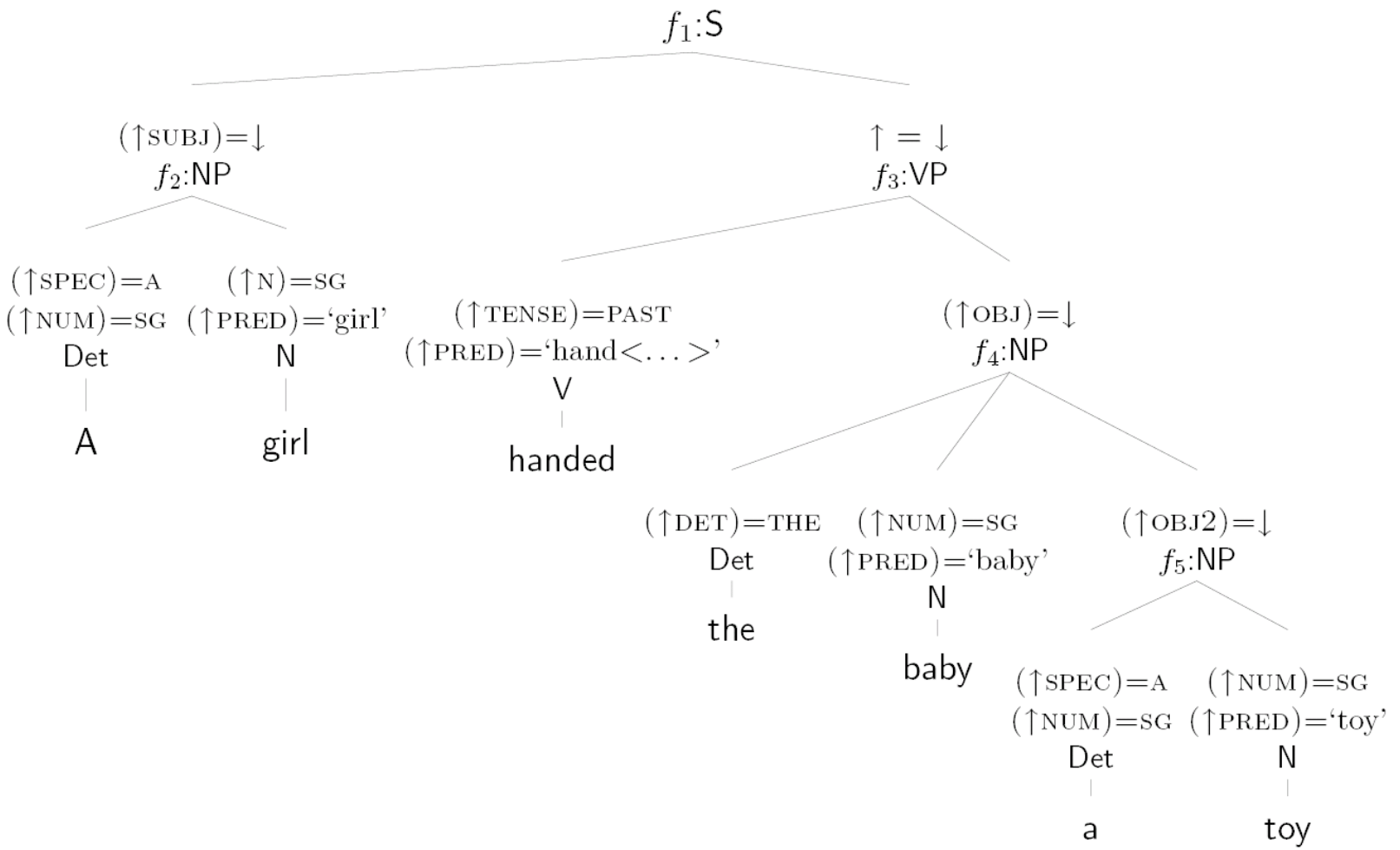
b. $NP \rightarrow \underset{\uparrow = \downarrow}{Det} \quad \underset{\uparrow = \downarrow}{N}$

c. $VP \rightarrow V \quad \underset{(\uparrow \text{OBJ}) = \downarrow}{NP} \quad \underset{(\uparrow \text{OBJ2}) = \downarrow}{NP}$

An example grammar II: The lexicon

(29) a.	*a*	Det	$(\uparrow \text{SPEC}) = \text{A}$ $(\uparrow \text{NUM}) = \text{SG}$
b.	*girl*	N	$(\uparrow \text{NUM}) = \text{SG}$ $(\uparrow \text{PRED}) =$ 'girl'
c.	*handed*	V	$(\uparrow \text{TENSE}) = \text{PAST}$ $(\uparrow \text{PRED}) =$ 'hand$<(\uparrow \text{SUBJ}), (\uparrow \text{OBJ}), (\uparrow \text{OBJ2})>$'
d.	*the*	Det	$(\uparrow \text{SPEC}) = \text{THE}$
e.	*baby*	N	$(\uparrow \text{NUM}) = \text{SG}$ $(\uparrow \text{PRED}) =$ 'baby'
f.	*toy*	N	$(\uparrow \text{NUM}) = \text{SG}$ $(\uparrow \text{PRED}) =$ 'toy'

A sentence licensed by the example grammar



The resulting f-structure for the example sentence

$$
f_1, f_3: \begin{bmatrix}
\text{SUBJ} & f_2: \begin{bmatrix} \text{SPEC} & \text{A} \\ \text{NUM} & \text{SG} \\ \text{PRED} & \text{'girl'} \end{bmatrix} \\
\text{TENSE} & \text{PAST} \\
\text{PRED} & \text{'hand} <(\uparrow\text{SUBJ}), (\uparrow\text{OBJ}), (\uparrow\text{OBJ2})>\text{'} \\
\text{OBJ} & f_4: \begin{bmatrix} \text{SPEC} & \text{THE} \\ \text{NUM} & \text{SG} \\ \text{PRED} & \text{'baby'} \end{bmatrix} \\
\text{OBJ2} & f_5: \begin{bmatrix} \text{SPEC} & \text{A} \\ \text{NUM} & \text{SG} \\ \text{PRED} & \text{'toy'} \end{bmatrix}
\end{bmatrix}
$$

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.:

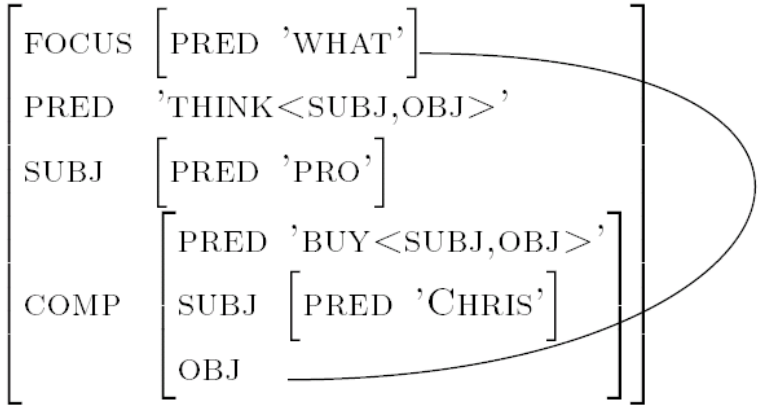
(30) $\text{CP} \rightarrow$ XP ($(\uparrow \text{FOCUS}) = \downarrow$, $(\uparrow \text{FOCUS}) = (\uparrow \text{COMP}^* \text{ GF})$) C' ($\uparrow = \downarrow$)

This says that the FOCUS element is equated with some GF after a path of COMP values

Extraction example

(31) What do you think Chris bought?

(32)

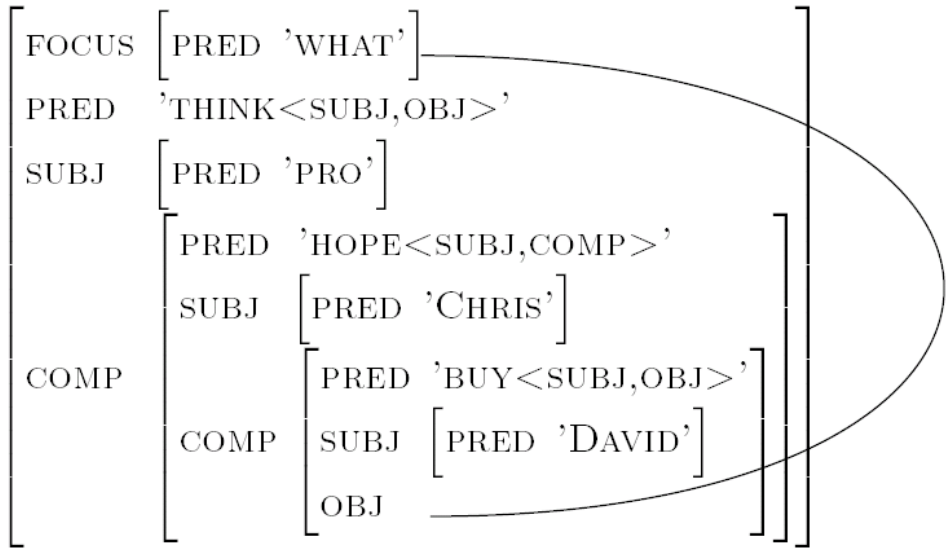


The principle of completeness ensures that *bought* has a realized object, and the functional equation fills it in.

Extraction example 2

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

(34)



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

$\Rightarrow$ This f/c-structure split is similar to the different between derivation and derived trees in TAG.

References

Austin, Peter K. (2001). Lexical Functional Grammar. In N.J. Smelser and P. Baltes (eds.), *International Encyclopedia of the Social and Behavioural Sciences*, Elsevier, pp. 8748–8754.

Dalrymple, Mary (2001). *Lexical Functional Grammar*, vol. 34 of *Syntax and Semantics*. New York: Academic Press.

Kaplan, Ronald M. and Joan Bresnan (1995). Lexical-Functional Grammar: A Formal System for Grammatical Representations. In Mary Dalrymple, Ronald M. Kaplan, III Maxwell, John t. and Annie Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, Stanford, CA: CSLI Publications, pp. 29–130.