LFG

Syntactic Theory
Winter Semester 2009/2010

Antske Fokkens

Department of Computational Linguistics
Saarland University

17 November 2009

Outline

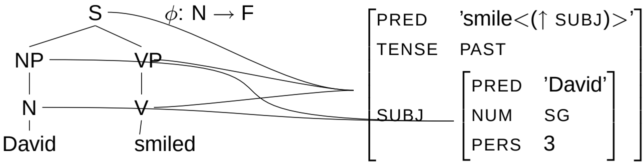
Outline

Function ϕ

- ϕ maps nodes to their associated f-structure, i.e. ϕ: N $\rightarrow$ F
- ϕ(*n*) leads to the f-structure associated with *n*
- ϕ(M(*n*)) leads to the f-structure associated with the mother node of *n*
- $\downarrow \equiv \phi(n)$
- $\uparrow \equiv \phi(M(n))$

Mapping from c- to f-structure: The head convention

- Consider the following example:



- The **head convention** states that a phrase inherits its functional properties and requirements from its head: a constituent structure phrase and its head map to the same f-structure
- S, VP and V thus map to the same f-structure

Annotating PS-rules: heads

- Consider the following rule to expand VP to V

  VP $\rightarrow$ V

- We express the fact that VP and V have the same f-structure by annotating the V-node:

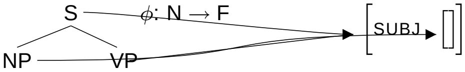
  $$\text{VP} \rightarrow \underset{\phi(\text{M}(n)) = \phi(n)}{\text{V}}$$

- This equation indicates that the f-structure of the mothernode of V (ϕ(M(n))) is equal to the node of V (ϕ(n))
  - An alternative notation:

    $$\text{VP} \rightarrow \underset{\uparrow = \downarrow}{\text{V}}$$

Annotating PS-rules: grammatical functions

- Consider the following example:



- Here the NP bears the SUBJ function
- The following phrase structure rule carries the additional information to derive the correct f-structure:

$$\begin{array}{lcc} S \rightarrow & NP & VP \\ & (\phi(M(n))\ \textsc{subj}) = \phi(n) & \phi(M(n)) = \phi(n) \end{array}$$

- An alternative notation:

$$\begin{array}{lcc} S \rightarrow & NP & VP \\ & (\uparrow \textsc{subj}) = \downarrow & \uparrow = \downarrow \end{array}$$

Lexical Entries

In lexical entries, information about the item's f-structure is represented in the same way as in c-structures:

smiled V $(\uparrow \textsc{pred})$ = 'smile$<(\uparrow \textsc{subj})>$'
$(\uparrow \textsc{tense}) = \textsc{past}$

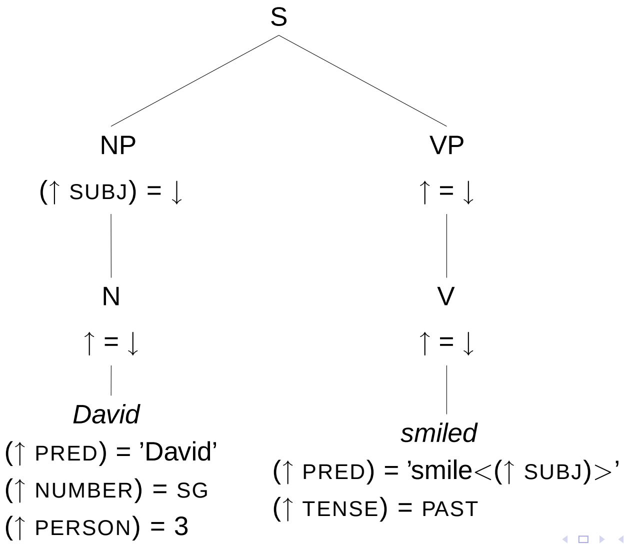
- The equivalent phrase structure rule:

V $\rightarrow$ *smiled*
$(\uparrow \textsc{pred})$ = 'smile$<(\uparrow \textsc{subj})>$'
$(\uparrow \textsc{tense}) = \textsc{past}$

An example analysis: *David smiled*

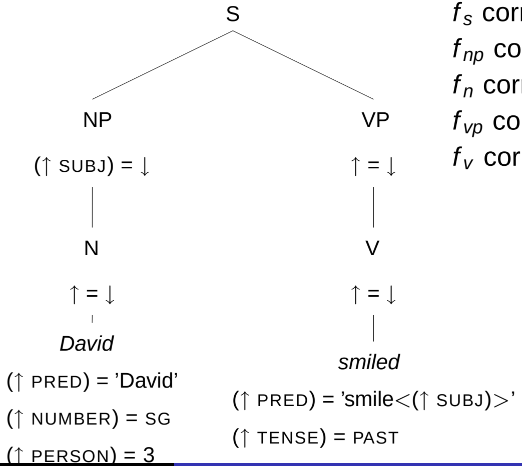
- We assume the following annotated PS-rules:
  - S $\rightarrow$ NP VP
    (↑ SUBJ) = ↓ ↑ = ↓
    VP $\rightarrow$ V
    ↑ = ↓
    NP $\rightarrow$ N
    ↑ = ↓
- and the following lexical entries
  - *smiled* V (↑ PRED) = 'smile<(↑ SUBJ)>'
    (↑ TENSE) = PAST
    David N (↑ PRED) 'David'
    (↑ NUMBER) = SG
    (↑ PERSON) = 3

Analysis of *David smiled*



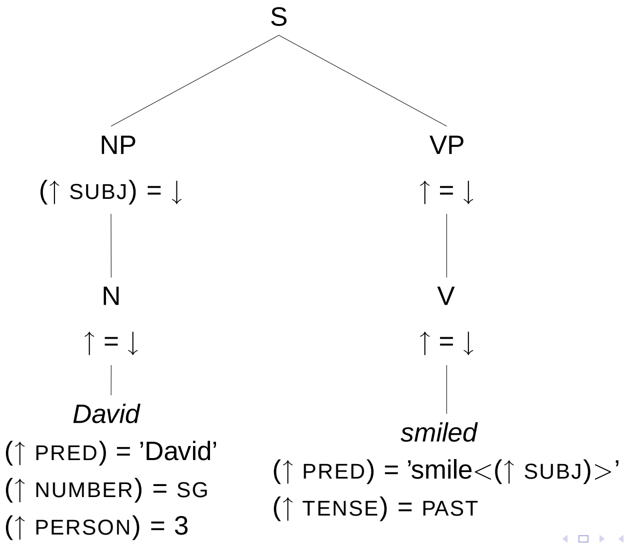
Instantiating the f-description of the sentence

- In order to get the functional description of the sentence, we associate each node with an f-structure:

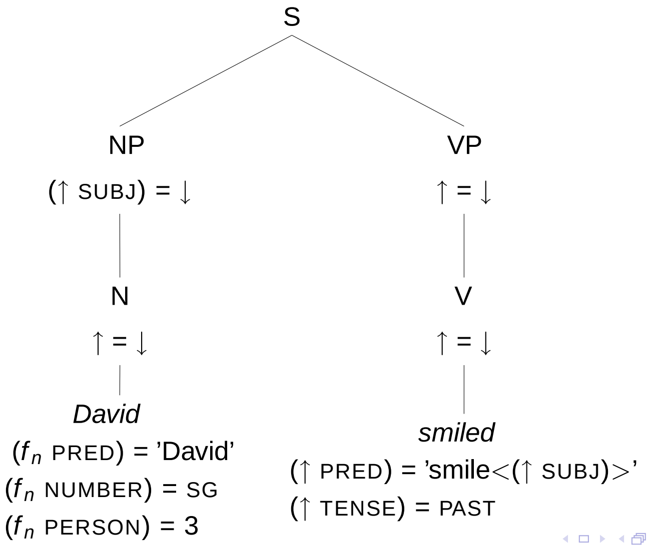


f_s corresponds to node S
f_{np} corresponds to node NP
f_n corresponds to node N
f_{vp} corresponds to node VP
f_v corresponds to node V

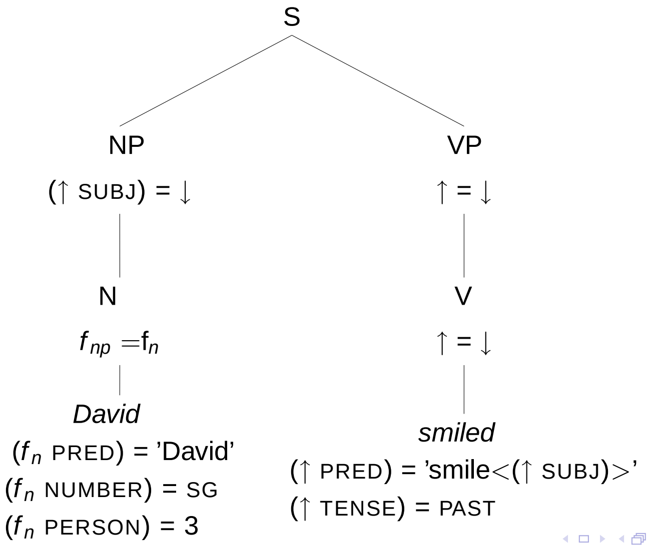
References of ↑ and ↓



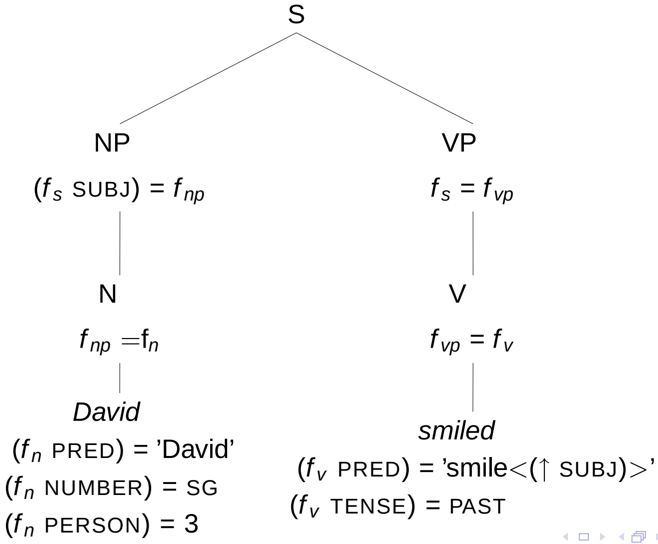
References of ↑ and ↓



References of ↑ and ↓



References of ↑ and ↓



The functional description

- The tree on the previous slide provides the following functional description:

$(f_s \text{ SUBJ}) = f_{np}$
$f_{np} = f_n$
$(f_n \text{ PRED}) = \text{'David'}$
$(f_n \text{ NUMBER}) = \text{SG}$
$(f_n \text{ PERSON}) = 3$
$f_s = f_{vp}$
$f_{vp} = f_v$
$(f_v \text{ PRED}) = \text{'smile}<(\uparrow\text{SUBJ})>\text{'}$
$(f_v \text{ TENSE}) = \text{PAST}$

The functional description

- The tree on the previous slide provides the following functional description:

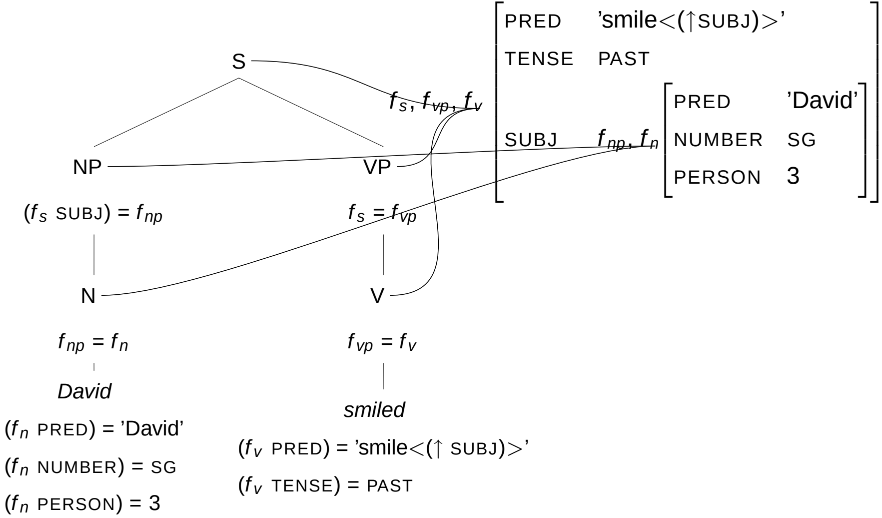
$(f_s \text{ SUBJ}) = f_{np}$
$f_{np} = f_n$
$(f_n \text{ PRED}) = \text{'David'}$
$(f_n \text{ NUMBER}) = \text{SG}$
$(f_n \text{ PERSON}) = 3$
$f_s = f_{vp}$
$f_{vp} = f_v$
$(f_v \text{ PRED}) = \text{'smile}<(\uparrow\text{SUBJ})>\text{'}$
$(f_v \text{ TENSE}) = \text{PAST}$

$$f_s, f_{vp}, f_v \begin{bmatrix} \text{PRED} & \text{'smile}<(\uparrow\text{SUBJ})>\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & f_{np}, f_n \begin{bmatrix} \text{PRED} & \text{'David'} \\ \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix} \end{bmatrix}$$

F-structure of an utterance

- The minimal solution of a functional description is the f-structure that satisfies the functional description and contains no additional attribute-value pairs
- The f-structure of an utterance is the *minimal solution* satisfying the constraints introduced by the words and the phrase structure of the utterance (Dalrymple 2001, p: 101)
- In other words: the f-structure of an utterance must contain all relevant information provided by the string, and nothing more

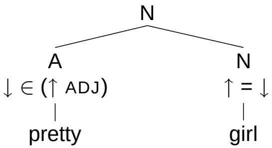
David smiled: f- and annotated c-structure



Adjuncts

- The attribute ADJ takes a set as its value
- The c-structure/f-structure correspondance rule expresses membership to a set as follows:

$$\begin{array}{lcc} N \rightarrow & \text{AdjP} & N \\ & \downarrow \in (\uparrow \text{ADJ}) & \uparrow = \downarrow \end{array}$$

- 

$$\begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \\ \text{ADJ} & \left\{ \begin{bmatrix} \text{PRED} & \text{'pretty'} \end{bmatrix} \right\} \end{bmatrix}$$

Outline

Grammatical functions in LFG

Recall: LFG has a universal inventory of arguments, which can be cross-classified in several ways:

- 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

We have seen governable functions and modifiers, in this lecture we'll look at other divisions and grammatical functions

Terms and non-terms

- Among governable functions, we distinguish *terms* 'direct functions' and *nonterms* 'oblique functions'
  - Terms: SUBJ, OBJ, OBJ_θ
  - NON-TERMS: OBL_θ, XCOMP, COMP
- The phenomena may distinguish terms from nonterms:
  - Agreement: in some language all and only term nominals trigger verb agreement (Ojibwa) (Southern Tiwa)
  - Anaphoric binding patterns: in some languages terms behave differently with respect to anaphoric binding:
    - Albanian: terms may be antecedent of any governable grammatical function, obliques may only be antecedent of obliques
  - Word order requirements:
    - In English, terms precede nonterms

Based on Dalrymple (2001)

Semantically restricted and unrestricted functions

- Subjects and objects are semantically unrestricted. In other words, the can be associated with any thematic role (Fillmore 1968), subject examples:

AGENT	*he hit the ball*
EXPERIENCER	*he felt cold*
THEME	*he lives in Saarbrücken*
PATIENT	*the window broke*
INSTRUMENT	*the stone broke the window*

- OBJ_θ and OBL_θ are bound to a specific thematic role, e.g. OBJ_{THEME} must always be a theme
  - I gave her *a book*
  - I asked him *a question*

Based on Dalrymple (2001)

Subject I

- The subject is the highest argument in the Keenan-Comrie hierarchy
- If a phenomenon is only applicable to one grammatical function, this is often the subject
- There are many tests to identify the subject, which tests apply differs from language to language (as for all functions)
  - Agreement: the subject is often the argument that agrees with the verb

    Moravcsik's universal: there is no language in which the verbs agrees with an element distinct from the intransitive subject, which does not also include sentences where the verb agrees with the intransitive subject
  - Honorification: in Japanese honorific verb forms are used to honor the subject (Matsumoto (1996))

Subject II

(1) sensei wa hon o-yomi ni
teacher TOPIC book-ACC honorific-READ COPULA
narimashi-ta
become.POLITE-PAST
'the teacher read a book'

(2) * Jon wa sensei ni o-tasuke-rare
John TOPIC teacher by HONORIFIC-help-PASSIVE
ni nat-ta
COPULA become-PAST
'John was saved by the teacher'

- Subject noncoreference: in Hindi the antecedent of a pronoun cannot be a subject of the same clause (Mohanan (1994))

Subject III

(3) Vijay ne Ravii ko uskii saikil par bithaayaa
Vijay ERG Ravi ACC his bicycle LOC sit.CAUSATIVE.PERFECT
"Vijay$_i$ seated Ravi$_j$ on his$_{*i,j}$ bike"

- The subject condition:
  - The subject condition states that:
    Every verbal predicate must have a subject
    → no consensus to whether this is universal, or only holds for most languages

Based on Dalrymple (2001)

Objects

- In some languages, there is a clear distinction between subjects and objects on the one hand, and other functions on the other hand
  - Languages may reveal subject and object agreement on the verb (e.g. Palauan, Abkhaz, Jingulu, Malayam)
  - Languages may allow only subjects and objects to be relativized (e.g. Kinyarwanda)
- Case marking can also indicate whether an element is an object, but note that this is seldom a one-to-one mapping

Based on Dalrymple (2001)

Multiple objects

- In many languages, there may be more than one phrase bearing the object function

  e.g. He gave her a book

- Originally, these second objects where called 'indirect objects' IOBJ or OBJ2 (after traditional grammar approaches)
- It has been observed though, that languages only have one unrestricted object, the secondary object is usually *thematically restricted*

  e.g. English: $\text{OBJ}_{\textit{THEME}}$
  He made her a cake
  * He made a cake her

Oblique

- Oblique arguments are
  - associated with a particular semantic role
  - marked to indicate their function overtly
- English marks oblique arguments with prepositions, in other languages, cases may be used
- Oblique arguments may
  1. bear a mark that reflects their semantic role ('semantic case'),

     e.g. OBL$_{GOAL}$ in *He gave the book to Chris*
  2. bear an idiosyncratic marker ('quirky case') (Butt and King (1999))

     e.g. *David relied on/*to/*about Chris*

Clausal functions

- COMP, XCOMP and XADJ are clausal functions
- the X in XCOMP and XADJ indicates that these functions are open functions: they have an external subject
- COMP is a closed function: its subject is internal
- XADJ differs from COMP and XCOMP in that it is a modifier

Clausal functions, examples I

- COMP clauses containing an overt subject internal to their phrase

  (4) David complained that Chris smiled

  (5) David wondered who smiled

  (6) David couldn't believe how big the house was

- XCOMP clauses that do not contain an internal subject, whose subject must be realized externally

  (7) David seemed to smile

  (8) Chris expected David to smile

- XADJ a modifier that has a subject that must be specified externally

Clausal functions, examples II

(9) Stretching his arms, David smiled

(10) David announced the news dancing

Outline

Some more examples of PS-rules

$$\mathrm{S} \longrightarrow \begin{matrix} \mathrm{NP} \\ (\uparrow \textsc{subj}) = \downarrow \end{matrix} \quad \begin{matrix} \mathrm{VP} \\ \uparrow = \downarrow \end{matrix}$$

$$\mathrm{VP} \longrightarrow \begin{matrix} \mathrm{VP} \\ \uparrow = \downarrow \end{matrix} \quad \begin{pmatrix} \mathrm{PP}^{+} \\ \downarrow \in (\uparrow \textsc{adj}) \end{pmatrix}$$

$$\mathrm{VP} \longrightarrow \begin{matrix} \mathrm{V} \\ \uparrow = \downarrow \end{matrix} \quad \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{obj}) = \downarrow \end{pmatrix} \quad \left\{ \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{obj}_{\theta}) = \downarrow \end{pmatrix} \middle| \begin{pmatrix} \mathrm{PP} \\ (\uparrow \textsc{obj}_{\theta}) = \downarrow \end{pmatrix} \right\}$$

$$\mathrm{NP} \longrightarrow \begin{matrix} \mathrm{Det} \\ \uparrow = \downarrow \end{matrix} \quad \begin{pmatrix} \mathrm{AP}^{+} \\ \downarrow \in (\uparrow \textsc{adj}) \end{pmatrix} \quad \begin{matrix} \mathrm{N} \\ \uparrow = \downarrow \end{matrix}$$

LFG Equations

- Assigning equations (attribute X has value Y)
- Constraining equations (attribute X must have value Y)
- Existential constraints (attribute X must be present and have some value)
- Negative equations (attribute X may not have value Y)

Assigning equations

- Assigning equations assign a specific value to an attribute

  e.g. (↑ DEF) = +
  (↑ TENSE) = PAST

- The expression may involve multiple attribute names:

  e.g. (↑ SUBJ NUM) = SG
  (↑ SUBJ PERSON) = 3

  - The example above can be used to ensure person-number agreement between subject and verb in English

Constraining equations (1/2)

- Constraining equation require an attribute to have a specific value

  e.g. $(\uparrow$ VFORM$) =_c$ PARTICIPLE

  - The constraint above can be used to ensure that complements of *have* (for present/past participles) and *be* (for passives) have the right form.

    (11) David has smiled/*smile/*smiles

    (12) The cat was chased/*chase/*chases

Constraining equations (2/2)

- They can be introduced in the lexicon or as part of the phrase structure rules:

  e.g. $S \rightarrow$ CP VP
  $(\uparrow \text{SUBJ COMPFORM}) =_c \text{THAT}$

  - This constraint can be used to account for the following data:

    (13) I believe (that) David smiled.

    (14) That David smiled surprised me.

    (15) * David smiled surprised me.

Existential constraints (1/2)

- Existential constraints require the presence of a feature

  e.g. $(\uparrow \text{DEF})$
  - This constraint can be used to account for obligatory presence of determiners for count nouns in singular:

    girl CN $\{ (\uparrow \text{NUM}) = \text{SG}$
    $(\uparrow \text{DEF}) \mid$
    $(\uparrow \text{NUM}) = \text{PL} \}$
  - This entry captures: *girl/a girl/the girl/girls

Existential constraints (2/2)

- Existential constraints are also found in the argument lists of semantic forms
- Like assigning and constraining equations, they can be introduced in the phrase structure grammar or in the lexicon:

  e.g. $S \rightarrow$ NP VP
  ($\uparrow$ TENSE)

- This constraint ensures that a sentence will be headed by a finite verb form. In the examples below *smile* is not marked for tense:

(16) David seemed to smile.

(17) * David smile.

Negative equations

- Negative equations can be used to *negate* f-descriptions, i.e. to prohibit them

e.g. Quirk et al. (1995) provide the following data:

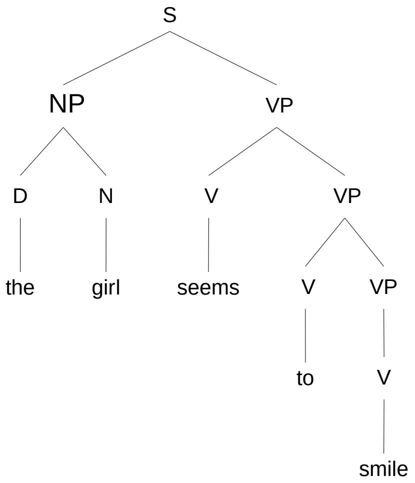
(i) I know whether/if David smiled
(ii) You have to justify whether/*if your journey is really necessary

- The following equation prevents *if* from occurring in sentence (ii)

justify $(\uparrow \text{COMP COMPFORM}) \neq \text{IF}$

Example analysis: *the girl seems to walk*

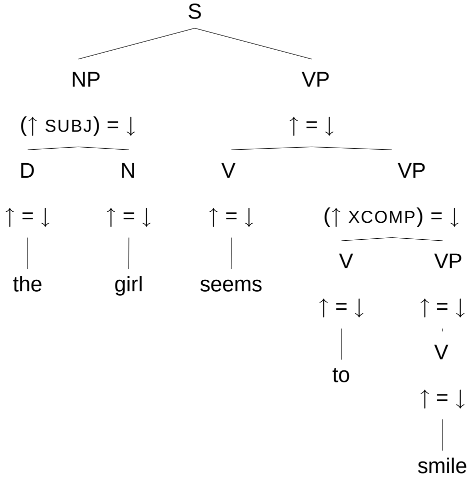
C-structure:



Example analysis: *the girl seems to walk*

C-structure:

+ annotations (words are presented in the next slides)



Adding lexicon and morphology *the girl*

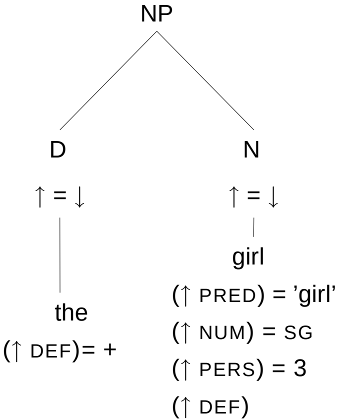
The (fully inflected) words *the* and *girl* provide the following information

- *the* D (↑ DEF) = +

  girl N (↑ PRED) = 'girl'
  (↑ NUMBER) = SG
  (↑ PERS) = 3
  (↑ DEF)
- Recall that (↑ DEF) as part of the word *girl* is an existential constraint: definiteness must be specified

The phrase *the girl*

- The NP and its associated f-structure looks like this:



$$\begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{NUM} & \text{SG} \\ \text{PERS} & 3 \\ \text{DEF} & + \end{bmatrix}$$

Lexical entry and morphology: *seems*

- *seems* V 'seem<(↑ XCOMP)>(↑ SUBJ)'
  (↑ SUBJ) = (↑ XCOMP SUBJ)
  (↑ TENSE) = PRESENT
  (↑ SUBJ PERS) = 3
  (↑ SUBJ NUM) = SG
  (↑ XCOMP VFORM) $=_c$ TO-INF
- Some remarks about these equations:
  - 'seem<(↑ XCOMP)>(↑ SUBJ)': SUBJ is outside of the square brackets because it is not a semantic argument of *seem*
  - (↑ SUBJ) = (↑ XCOMP SUBJ) states that the subject of XCOMP and the subject of *seem* are identical
  - (↑ SUBJ PERS) = 3 and (↑ SUBJ NUM) = SG ensure subject verb agreement
  - (↑ XCOMP TO) $=_c$ + constraints the verbform of XCOMP to be an infinitive

Lexical entry and morphology: *to smile*

- to V $(\uparrow \textsc{to}) = +$
  $(\uparrow \textsc{vform}) =_c \textsc{inf}$

  smile V $(\uparrow \textsc{pred}) = \text{'smile}\langle(\uparrow \textsc{subj})\rangle\text{'}$
  $(\uparrow \textsc{vform}) = \textsc{inf}$
- Remarks:
  - The presence of *to* is provided by the verb selecting XCOMP
  - the verb combined with *to* has to be an infinitive
- How can we prevent an infinitive form such as *smile* to become the head of a sentence (e.g. **the girl smile*)?

LFG – Recap I

- LFG (minimally) distinguishes two kinds of representation:
  - the f-structure encoding functional relation using feature structures
  - the c-structure encoding constituency using phrase structure trees
- The function ϕ allows to map nodes from the c-structure to f-structures (N.B. there is also a relation ϕ^{-1} that maps from f-structures to c-structure nodes, which we have not seen in these lectures)
- Words and structure provide information about grammatical functions in an utterance
- An f-structure of an utterance is the minimal solution satisfying the constraints provided by the words and structure of the utterance

LFG – Recap II

- Equations can be used to constrain what well-formed f-structures may look like. They can:
  - Assign a specific value to an attribute
  - Restrict attribute to have a specific value
  - Require a feature to be present (with some value)
  - Prevent an attribute from having a specific value

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.