

Formale Syntax

Formale Syntax und Grammatikarchitektur der Lexikalisch-Funktionalen Grammatik (II)

Prof. Dr. Anette Frank

Institut für Computerlinguistik
Universität Heidelberg

08.05.2018

Projektionsarchitektur der LFG: von C- zu F-Struktur

C-Struktur

- Konstituenz und Wortordnung unterliegen Regularitäten, aber auch großer Variation über die Sprachen hinweg.

F-Struktur

- Grammatische Funktionen und morpho-syntaktische Merkmale sind primitive Kategorien, die den meisten Sprachen gemeinsam sind.

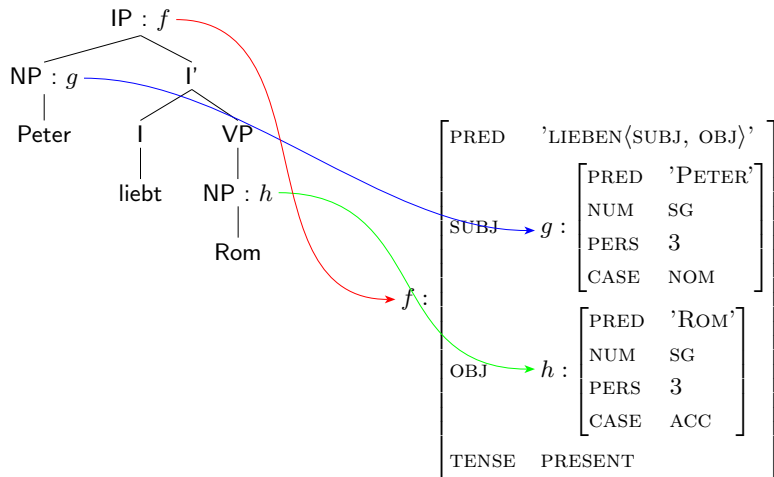
Funktionale Abbildung zwischen C- und F-Struktur

- Durch eine *funktionale Abbildung* von C- zu F-Strukturen können unterschiedliche Konstituentenstrukturen auf eine nach gemeinsamen Prinzipien strukturierte F-Struktur abgebildet werden.

Eine solche Abbildung muss sehr flexibel sein !!

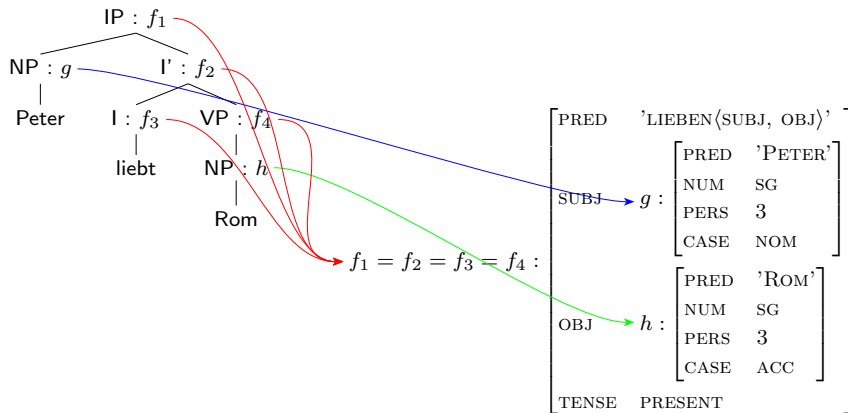
- **Heute: Resolutionsalgorithmus** zur Projektion von F-Strukturen

Projektionsarchitektur: first glimpse



Peter liebt Rom.

Projektionsarchitektur: first glimpse



Resolutionsalgorithmus für F-Strukturen

- Funktionale Abbildung von Teil-Konstituenten der C-Struktur auf Teile der zugehörigen F-Struktur
- Definition von Korrespondenzen auf Basis von Phrasenstrukturregeln und Lexikon
 - Spezifikation von F-Strukturen:
'defining' vs. 'constraining equations'
 - Funktionale Abbildung ϕ und Metavariablen
- Algorithmus zur Lösung des funktionalen Gleichungssystems zur Berechnung der F-Struktur

Funktionale Beschreibungen: (partielle) Spezifikation von F-Strukturen

Functional Descriptions (f-descriptions)

Wir unterscheiden zwischen **F-Strukturen** und **Beschreibungen (Spezifikationen) von F-Strukturen**:

- $(f \text{ TENSE})$ referiert auf den Wert des Merkmals TENSE in F-Struktur f .
- $(f_1 \text{ TENSE}) = \text{PRES}$; $(f_2 \text{ TENSE}) = \text{PAST}$

$$f_1 : \left[\begin{array}{ll} \text{PRED} & \text{'SINGEN}\langle \text{SUBJ} \rangle\text{' } \\ \text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

$$f_2 : \left[\begin{array}{ll} \text{PRED} & \text{'LACHEN}\langle \text{SUBJ} \rangle\text{' } \\ \text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PAST} \end{array} \right]$$

Funktionale Beschreibungen: (partielle) Spezifikation von F-Strukturen

Defining equations

- $(f \text{ TENSE}) = \text{PAST}$ spezifiziert den Wert des Merkmals TENSE in F-Struktur f als PAST.
- $(f \text{ SUBJ}) = g$ definiert den Wert des Merkmals SUBJ in F-Struktur f durch die F-Struktur g .
- $h \in (f \text{ ADJ})$ spezifiziert, dass die F-Struktur h ein Element der Menge ist, die in f als Wert des Merkmals ADJ definiert ist.

'Defining vs. constraining equations'

Defining equations

- Eine F-Struktur kann durch eine Menge atomarer funktionaler Beschreibungen spezifiziert (oder beschrieben) werden.
- Beschreibung der F-Struktur für *Peter singt.* :
 $(f \text{ PRED}) = \text{'SINGEN'}$, $(f \text{ TENSE}) = \text{PRES}$,
 $(f \text{ SUBJ}) = g$, $(g \text{ PRED}) = \text{'PETER'}$

'Defining vs. constraining equations'

Defining equations

- Eine F-Struktur kann durch eine Menge atomarer funktionaler Beschreibungen spezifiziert (oder beschrieben) werden.
- Beschreibung der F-Struktur für *Peter singt.* :
(f PRED)= 'SINGEN<SUBJ>', (f TENSE)= PRES,
(f SUBJ)= g , (g PRED)= 'PETER'
- Es gibt mind. zwei F-Strukturen, die diese Bedingungen erfüllen:

$$f_1 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right] \quad f_2 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{ADJ} & \left\{ \left[\begin{array}{cc} \text{PRED} & \text{'OFT'} \end{array} \right] \right\} \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

'Defining vs. constraining equations'

Defining equations

- Eine F-Struktur kann durch eine Menge atomarer funktionaler Beschreibungen spezifiziert (oder beschrieben) werden.
- Beschreibung der F-Struktur für *Peter singt.* :
(f PRED) = 'SINGEN(SUBJ)', (f TENSE) = PRES,
(f SUBJ) = g , (g PRED) = 'PETER'
- Es gibt mind. zwei F-Strukturen, die diese Bedingungen erfüllen:

$$f_1 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN(SUBJ)'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right] \quad f_2 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN(SUBJ)'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{ADJ} & \left\{ \left[\begin{array}{cc} \text{PRED} & \text{'OFT'} \end{array} \right] \right\} \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

- I.a. suchen wir die *minimale F-Struktur*, die alle Bedingungen der funktionalen Beschreibung erfüllt!

'Defining vs. constraining equations'

Constraining equations

- Neben definierenden funktionalen Gleichungen gibt es sog. *constraining equations*.
- Mit ihnen können **Bedingungen** über die Form der minimalen F-Struktur definiert werden – im Gegensatz zur (partiellen) **Definition** einer F-Struktur.
- Beispiel: $(f \text{ SUBJ NUM}) =_c \text{SG}$

Welche der beiden F-Strukturen erfüllt diese Bedingung?

$$f_1 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \\ \text{NUM} & \text{SG} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

$$f_2 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

'Defining vs. constraining equations'

Constraining equations

- Neben definierenden funktionalen Gleichungen gibt es sog. *constraining equations*.
- Mit ihnen können **Bedingungen** über die Form der minimalen F-Struktur definiert werden – im Gegensatz zur (partiellen) **Definition** einer F-Struktur.
- Beispiel: $(f \text{ SUBJ NUM}) =_c \text{SG}$

f_1 erfüllt die Gleichung, f_2 erfüllt die Gleichung nicht.

$$f_1 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \\ \text{NUM} & \text{SG} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

$$f_2 : \left[\begin{array}{cc} \text{PRED} & \text{'SINGEN<SUBJ>'} \\ \text{SUBJ} & \left[\begin{array}{cc} \text{PRED} & \text{'PETER'} \end{array} \right] \\ \text{TENSE} & \text{PRES} \end{array} \right]$$

'Defining vs. constraining equations'

Arten von constraining equations

- Positive constraining equation: $(f \text{ TENSE}) =_c \text{past}$
- Negative constraining equation: $(f \text{ TENSE}) \neq \text{present}$
- Existential constraint: $(f \text{ TENSE})$
- Negative existential constraint: $\neg (f \text{ TENSE})$

'Defining vs. constraining equations'

Lexikoneintrag für *sneeze*:

- Implizite Konjunktion der einzelnen f-descriptions

sneezes: (*f* PRED)= 'sneeze⟨SUBJ⟩'
(*f* TENSE)= present
(*f* SUBJ PERS)= 3
(*f* SUBJ NUM) = sg.

'Defining vs. constraining equations'

Lexikoneintrag für *sneeze*:

- Implizite Konjunktion der einzelnen f-descriptions

sneezes: (*f* PRED)= 'sneeze⟨SUBJ⟩'
(*f* TENSE)= present
(*f* SUBJ PERS)= 3
(*f* SUBJ NUM) = sg.

- Daneben: Disjunktion und Negation

sneeze: (*f* PRED)= 'sneeze⟨SUBJ⟩'
 { (*f* VFORM)= base
 | (*f* TENSE)= present
 ¬ { (*f* SUBJ PERS)= 3
 (*f* SUBJ NUM) = sg } }.

'Defining vs. constraining equations'

Lexikoneintrag für *sneeze*:

sneeze: (f PRED)= 'sneeze<SUBJ>
 { (f VFORM)= base
 | (f TENSE)= present
 \neg { (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg } }.

sneezes: (f PRED)= 'sneeze<SUBJ>
 (f TENSE)= present
 (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg.

Lexikoneintrag für *isst*:

Lexikoneintrag für *schenkte*:

'Defining vs. constraining equations'

Lexikoneintrag für *sneeze*:

sneeze: (f PRED)= 'sneeze<SUBJ>
 { (f VFORM)= base
 | (f TENSE)= present
 \neg { (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg } }.

sneezes: (f PRED)= 'sneeze<SUBJ>
 (f TENSE)= present
 (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg.

Lexikoneintrag für *isst*:

isst: (f PRED)= 'essen<SUBJ,OBJ>
 (f TENSE)= present
 (f SUBJ NUM) = sg
 {(f SUBJ PERS)=2
 | (f SUBJ PERS)= 3 }.

Lexikoneintrag für *schenkte*:

'Defining vs. constraining equations'

Lexikoneintrag für *sneeze*:

sneeze: (f PRED)= 'sneeze<SUBJ>'
 { (f VFORM)= base
 | (f TENSE)= present
 \neg { (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg } }.

sneezes: (f PRED)= 'sneeze<SUBJ>'
 (f TENSE)= present
 (f SUBJ PERS)= 3
 (f SUBJ NUM) = sg.

Lexikoneintrag für *isst*:

isst: (f PRED)= 'essen<SUBJ,OBJ>'
 (f TENSE)= present
 (f SUBJ NUM) = sg
 { (f SUBJ PERS)=2
 | (f SUBJ PERS)= 3 }.

Lexikoneintrag für *schenke*:

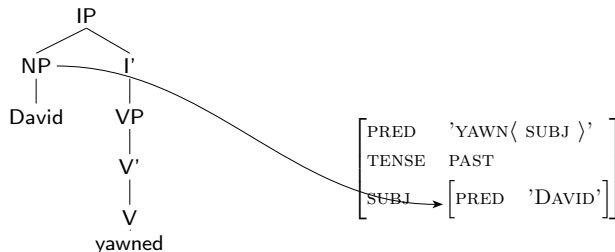
schenke: (f PRED)= 'schenken<SUBJ,OBJ,OBJ _{θ} >'
 (f TENSE)= past
 (f SUBJ NUM) = sg
 { (f SUBJ PERS)= 1
 | (f SUBJ PERS)= 3 }.

Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Im Englischen ist der Spezifikator der IP (satzinitiale NP) SUBJekt.

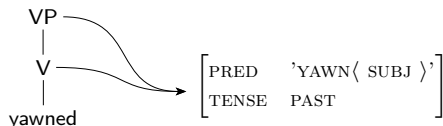


Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Die F-Struktur (FS) des Verbs ist gleich der FS des V'/VP-Knotens.



Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

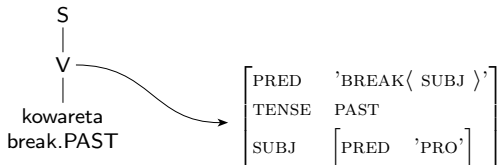
Beispiele:

- In manchen Sprachen ist in finiten Sätzen ein pronominales SUBJekt fakultativ. Das finite Verb definiert ein pronominales SUBJekt: (f SUBJ PRED) = 'PRO'.

koware-ta

break.PAST

'[IT/Something] broke.'

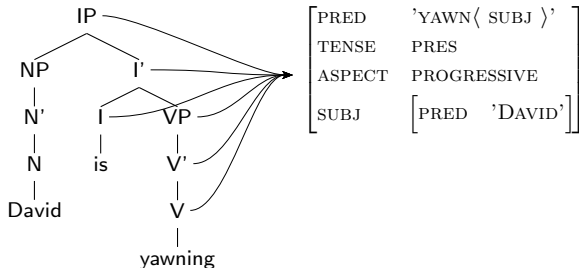


Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Alle lexikalischen und funktionalen Projektionen des Verbs korrespondieren mit ein und derselben F-Struktur

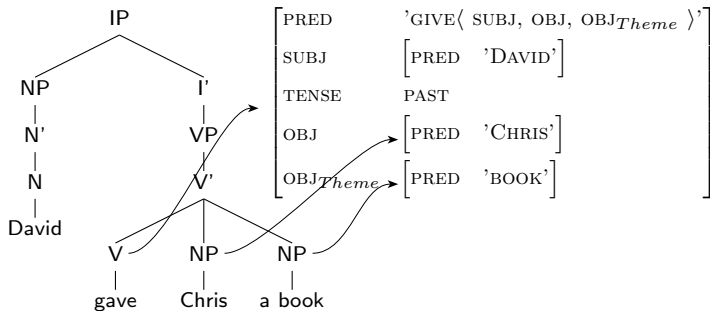


Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Im Englischen ist die Reihenfolge der Objekte relativ strikt festgelegt. In anderen Sprachen werden direktes und indirektes Objekt durch Kasusmerkmale identifiziert.

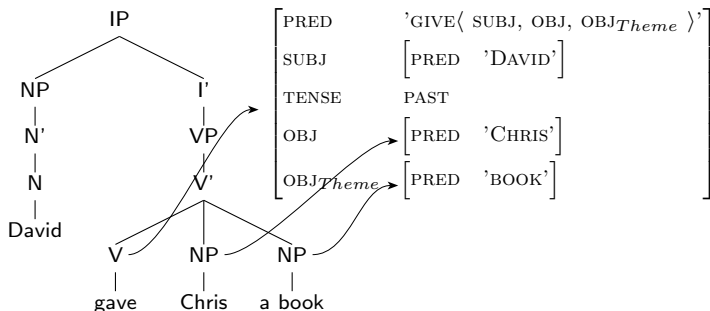


Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Im Englischen ist die Reihenfolge der Objekte relativ strikt festgelegt. In anderen Sprachen werden direktes und indirektes Objekt durch Kasusmerkmale identifiziert.



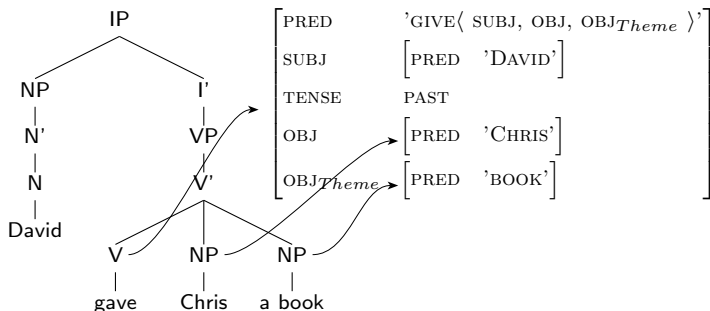
(Der Mann)_{Nom} gab (dem Pferd)_{Dat} (einen Kuss)_{Akk}

Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Im Englischen ist die Reihenfolge der Objekte relativ strikt festgelegt. In anderen Sprachen werden direktes und indirektes Objekt durch Kasusmerkmale identifiziert.



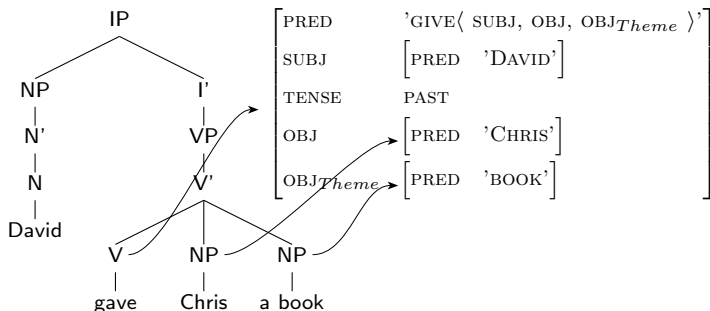
(Dem Pferd)_{Dat} gab (der Mann)_{Nom} (einen Kuss)_{Akk}

Korrespondenzen zwischen C- und F-Struktur

In vielen Sprachen und über Sprachen hinweg bestehen systematische Korrespondenzen zwischen C- und F-Strukturen.

Beispiele:

- Im Englischen ist die Reihenfolge der Objekte relativ strikt festgelegt. In anderen Sprachen werden direktes und indirektes Objekt durch Kasusmerkmale identifiziert.



(Einen Kuss)_{Akk} gab (der Mann)_{Nom} (dem Pferd)_{Dat}

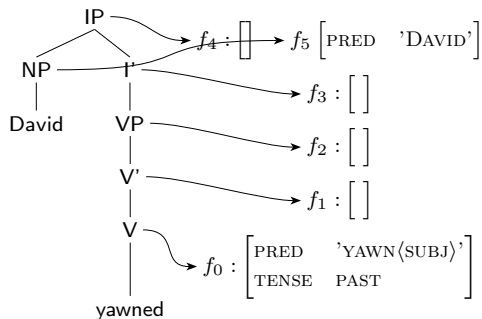
Funktionale Abbildung ϕ zwischen C- und F-Struktur

- Jedem C-Struktur-Knoten wird eine partielle F-Struktur zugeordnet
- durch eine Funktion ϕ , die jeden C-Struktur-Knoten auf eine F-Struktur (Variable) abbildet
- Lexikon und Phrasenstrukturregeln definieren zusätzliche Bedingungen durch *funktionale Beschreibungen*

Lexikon:

yawned: (f_0 PRED)= 'yawn< SUBJ >'
(f_0 TENSE)= past.

David: (f_5 PRED)= 'David'.



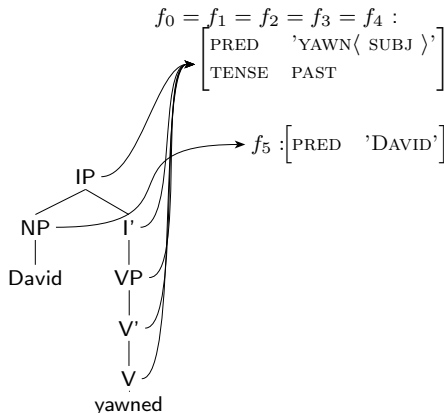
Funktionale Abbildung ϕ zwischen C- und F-Struktur

- Jedem C-Struktur-Knoten wird eine partielle F-Struktur zugeordnet
- durch eine Funktion ϕ , die jedem C-Struktur-Knoten eine F-Struktur (Variable) zuordnet.
- Lexikon und Phrasenstrukturregeln definieren zusätzliche Bedingungen durch *funktionale Beschreibungen*.

Grammatik:

Alle Knoten der Verb-Projektion werden auf dieselbe F-Struktur abgebildet:

V'	\rightarrow	V	:	$f_1 = f_0$
VP	\rightarrow	V'	:	$f_2 = f_1$
I'	\rightarrow	VP	:	$f_3 = f_2$
IP	\rightarrow	$NP\ I'$:	$f_4 = f_3$



Funktionale Abbildung ϕ zwischen C- und F-Struktur

- Jedem C-Struktur-Knoten wird eine partielle F-Struktur zugeordnet
- durch eine Funktion ϕ , die jedem C-Struktur-Knoten eine F-Struktur (Variable) zuordnet.
- Lexikon und Phrasenstrukturregeln definieren zusätzliche Bedingungen durch *funktionale Beschreibungen*.

Grammatik:

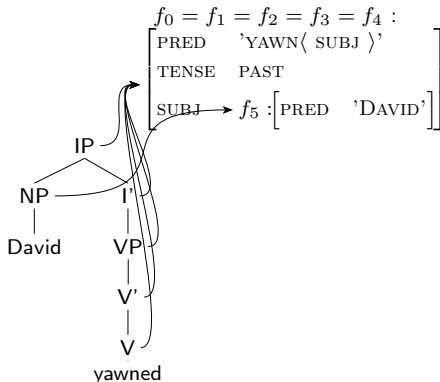
Die FS der satzinitialen NP (f_5) ist Subjekt der FS des Satzes (f_4):

$IP \rightarrow NP I'$

$\phi(\text{node}(\text{NP})) = f_5$

$\phi(\text{node}(\text{IP})) = f_0 = \dots = f_4$

Constraint: $(f_4 \text{ SUBJ}) = f_5$



Funktionale Gleichungen in PS-Regeln und Lexikon

- Funktionale Gleichungen werden definiert für PS-Regeln und Lexikon
- Aber: Lexikoneinträge und PS-Regeln können mehrfach auftreten !

Funktionale Gleichungen in PS-Regeln und Lexikon

- Verwendung sog. *Metavariablen* \uparrow und \downarrow für FS-Variablen (f_0, \dots)
 - \uparrow referiert auf FS-Variable der Mutter des lokalen Knotens n
 $\uparrow := \phi(\text{mother}(n))$
 - \downarrow referiert auf FS-Variable des lokalen Knotens n : $\downarrow := \phi(n)$

Funktionale Gleichungen in PS-Regeln und Lexikon

- Verwendung sog. *Metavariablen* \uparrow und \downarrow für FS-Variablen (f_0, \dots)
 - \uparrow referiert auf FS-Variable der Mutter des lokalen Knotens n
 $\uparrow := \phi(\text{mother}(n))$
 - \downarrow referiert auf FS-Variable des lokalen Knotens n : $\downarrow := \phi(n)$

“Die NP-Tochter von IP entspricht dem SUBJekt des Verbs”

$$\begin{array}{ccc} \text{IP} & \rightarrow & \text{NP} \quad \text{I}' \\ & & (\uparrow \text{SUBJ}) = \downarrow \quad \uparrow = \downarrow \end{array}$$

$(\uparrow \text{SUBJ}) = \downarrow$: Die FS des lokalen Knotens (NP) ist gleich dem Wert des Merkmals SUBJ in der FS der Mutter von NP (= IP)

Funktionale Gleichungen in PS-Regeln und Lexikon

- Verwendung sog. *Metavariablen* \uparrow und \downarrow für FS-Variablen (f_0, \dots)
 - \uparrow referiert auf FS-Variable der Mutter des lokalen Knotens n
 $\uparrow := \phi(\text{mother}(n))$
 - \downarrow referiert auf FS-Variable des lokalen Knotens n : $\downarrow := \phi(n)$

“Die NP-Tochter von IP entspricht dem SUBJekt des Verbs”

$$\begin{array}{ccc} \text{IP} \rightarrow & \text{NP} & \text{I}' \\ & (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$

$(\uparrow \text{SUBJ}) = \downarrow$: Die FS des lokalen Knotens (NP) ist gleich dem Wert des Merkmals SUBJ in der FS der Mutter von NP (= IP)

- Lexikoneintrag für lieben: lieben V $(\uparrow \text{PRED}) = \text{'lieben}\langle \text{SUBJ}, \text{OBJ} \rangle'$

1 Annotation:

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

1 Annotation:

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

2 Instantiierung:

- Dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation wird eine (jeweils neue) FS-Variable zugewiesen.
D.h. jeder Knoten, der auf eine partielle F-Struktur abzubilden ist, erhält eine eindeutige F-Struktur-Variable.
- Jedes \uparrow bzw. \downarrow wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.

1 Annotation:

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

2 Instantiierung:

- Dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation wird eine (jeweils neue) FS-Variable zugewiesen.
D.h. jeder Knoten, der auf eine partielle F-Struktur abzubilden ist, erhält eine eindeutige F-Struktur-Variable.
- Jedes \uparrow bzw. \downarrow wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.

3 Konstruktion einer minimalen F-Struktur:

- Die resultierenden funktionalen Gleichungen müssen *konsistent* sein und die *Wohlgeformtheitsbedingungen* für FS erfüllen.
- Die F-Struktur muss *minimal* sein. Es ist die *kleinste F-Struktur*, die alle Gleichungen erfüllt und keine zusätzlichen Attribut-Wert-Paare enthält.

1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .

I' \rightarrow I VP.

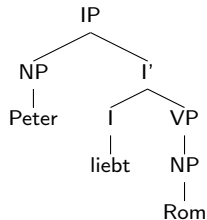
VP \rightarrow NP .

NP \rightarrow N .

liebt V (\uparrow PRED)= 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM)= SG

...
Rom N (\uparrow PRED)= 'Rom'
 (\uparrow NUM)= SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
 (\uparrow SUBJ)= \downarrow

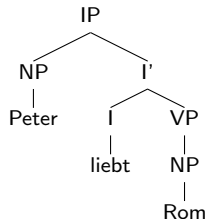
I' \rightarrow I VP.

VP \rightarrow NP .

NP \rightarrow N .

liebt V (\uparrow PRED)= 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM)= SG

...
Rom N (\uparrow PRED)= 'Rom'
 (\uparrow NUM)= SG
...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
 (\uparrow SUBJ)= \downarrow $\uparrow=\downarrow$

I' \rightarrow I VP.

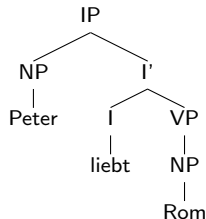
VP \rightarrow NP .

NP \rightarrow N .

liebt V (\uparrow PRED)= 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM)= SG

...
Rom N (\uparrow PRED)= 'Rom'
 (\uparrow NUM)= SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
 (\uparrow SUBJ) = \downarrow $\uparrow = \downarrow$

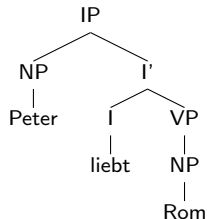
I' \rightarrow I VP.
 $\uparrow = \downarrow$

VP \rightarrow NP .

NP \rightarrow N .

liebt V (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM) = SG

...
Rom N (\uparrow PRED) = 'Rom'
 (\uparrow NUM) = SG
...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
 (\uparrow SUBJ)= \downarrow $\uparrow=\downarrow$

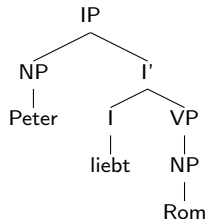
I' \rightarrow I VP.
 $\uparrow=\downarrow$ $\uparrow=\downarrow$

VP \rightarrow NP .

NP \rightarrow N .

liebt V (\uparrow PRED)= 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM)= SG

...
Rom N (\uparrow PRED)= 'Rom'
 (\uparrow NUM)= SG
...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
 (\uparrow SUBJ) = \downarrow $\uparrow = \downarrow$

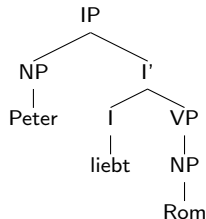
I' \rightarrow I VP.
 $\uparrow = \downarrow$ $\uparrow = \downarrow$

VP \rightarrow NP .
 (\uparrow OBJ) = \downarrow

NP \rightarrow N .

liebt V (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
 (\uparrow SUBJ NUM) = SG

...
Rom N (\uparrow PRED) = 'Rom'
 (\uparrow NUM) = SG
...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

$$\text{IP} \rightarrow \begin{array}{cc} \text{NP} & \text{I}' \\ (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$
$$\begin{array}{ccc} I' \rightarrow & I & VP. \\ & \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

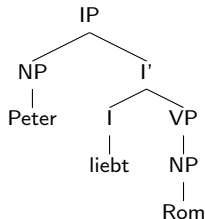
VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

$$\text{NP} \rightarrow \text{N} \cdot$$

$$\uparrow = \downarrow$$

liebt V (↑ PRED)= 'lieben<SUBJ,OBJ>'
 (↑ SUBJ NUM)= SG

Rom N (↑ PRED)= 'Rom'
 (↑ NUM)= SG



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
(\uparrow SUBJ NUM) = SG

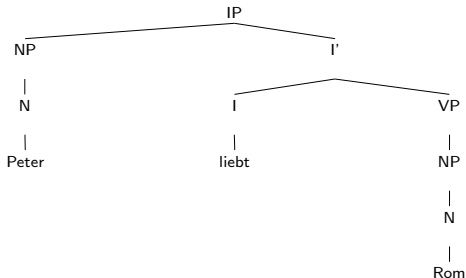
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ, OBJ)'
(\uparrow SUBJ NUM) = SG

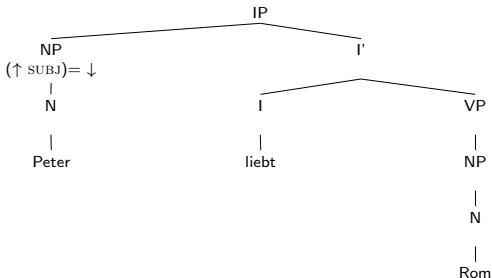
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ, OBJ)'
(\uparrow SUBJ NUM) = SG

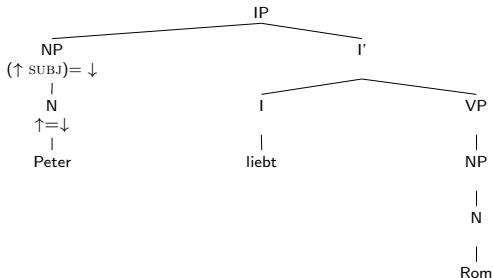
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
(\uparrow SUBJ NUM) = SG

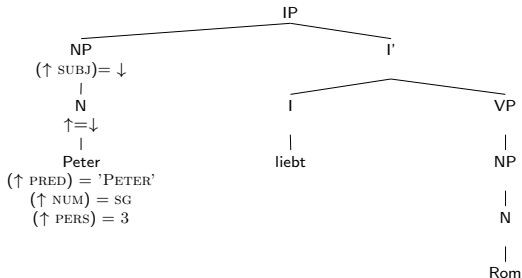
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
(\uparrow SUBJ NUM) = SG

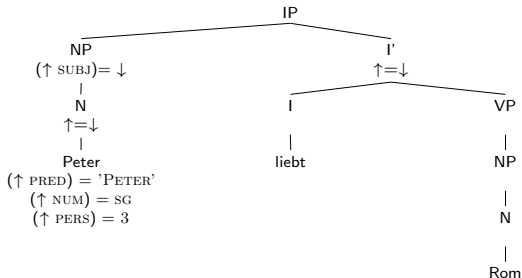
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben(SUBJ,OBJ)'
(\uparrow SUBJ NUM) = SG

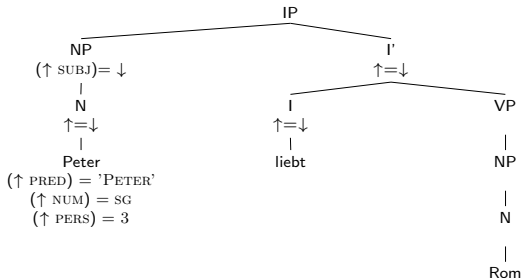
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP → NP I' .
(↑ SUBJ)= ↓ ↑=↓

I' → I VP.
↑=↓ ↑=↓

VP → NP .
(↑ OBJ)= ↓

NP → N .
↑ = ↓

liebt I (↑ PRED)= 'lieben<SUBJ,OBJ>'
(↑ SUBJ NUM)= SG

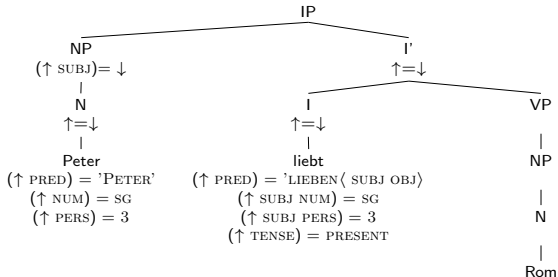
...

Peter N (↑ PRED)= 'Peter'
(↑ NUM)= SG

...

Rom N (↑ PRED)= 'Rom'
(↑ NUM)= SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben' (SUBJ, OBJ)'
(\uparrow SUBJ NUM) = SG

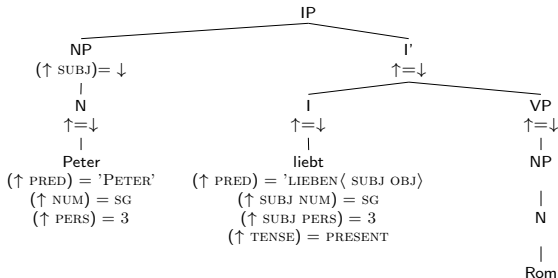
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow \uparrow = \downarrow

I' \rightarrow I VP.
 \uparrow = \downarrow \uparrow = \downarrow

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 \uparrow = \downarrow

liebt I (\uparrow PRED) = 'lieben' (SUBJ, OBJ)'
(\uparrow SUBJ NUM) = SG

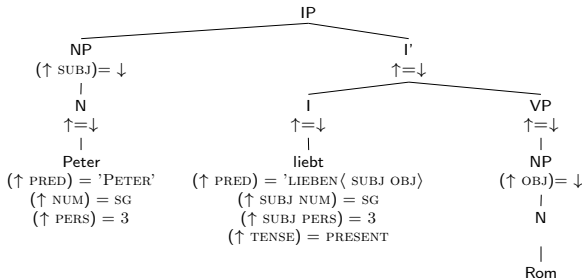
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP \rightarrow NP I' .
(\uparrow SUBJ) = \downarrow $\uparrow = \downarrow$

I' \rightarrow I VP.
 $\uparrow = \downarrow$ $\uparrow = \downarrow$

VP \rightarrow NP .
(\uparrow OBJ) = \downarrow

NP \rightarrow N .
 $\uparrow = \downarrow$

liebt I (\uparrow PRED) = 'lieben' (SUBJ, OBJ)'
(\uparrow SUBJ NUM) = SG

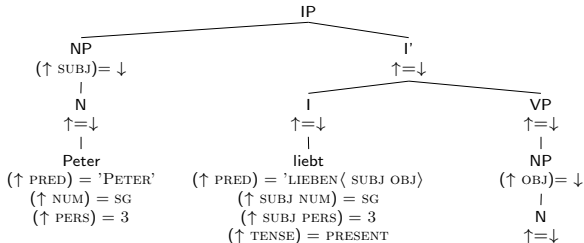
...

Peter N (\uparrow PRED) = 'Peter'
(\uparrow NUM) = SG

...

Rom N (\uparrow PRED) = 'Rom'
(\uparrow NUM) = SG

...



1. Annotation

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

IP → NP I' .
(↑ SUBJ) = ↓ ↑ = ↓

I' → I VP.
↑ = ↓ ↑ = ↓

VP → NP .
(↑ OBJ) = ↓

NP → N .
↑ = ↓

liebt I (↑ PRED) = 'lieben' (SUBJ, OBJ)'
(↑ SUBJ NUM) = SG

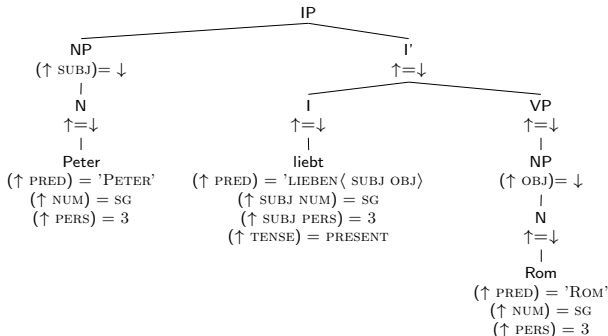
...

Peter N (↑ PRED) = 'Peter'
(↑ NUM) = SG

...

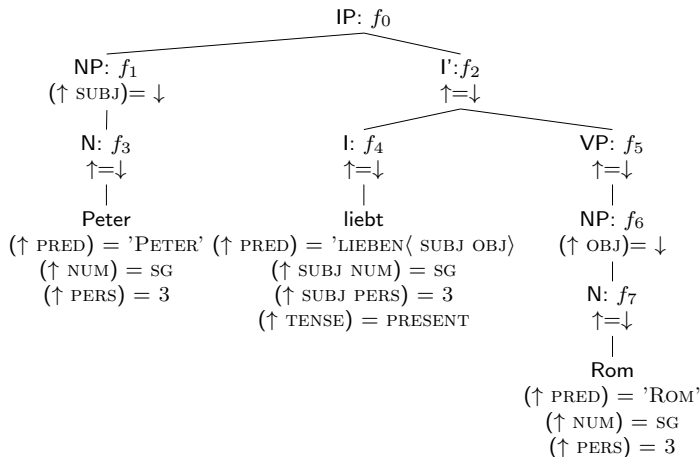
Rom N (↑ PRED) = 'Rom'
(↑ NUM) = SG

...



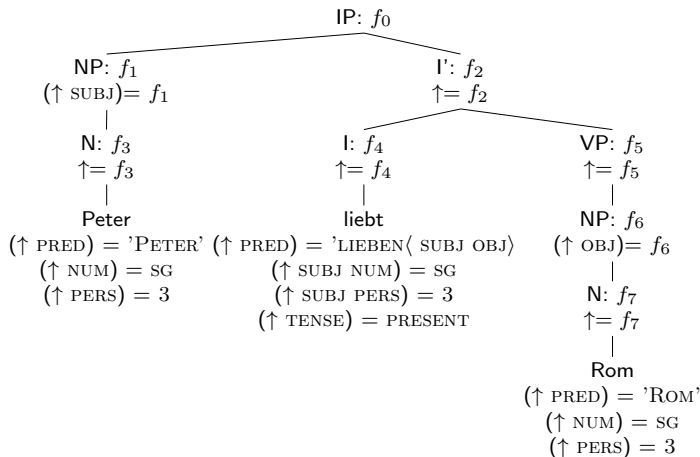
2.a Instantiierung

- a. Weise dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation eine FS-Variable zu.
- b. Jedes \uparrow bzw. \downarrow in funktionalen Gleichungen wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.



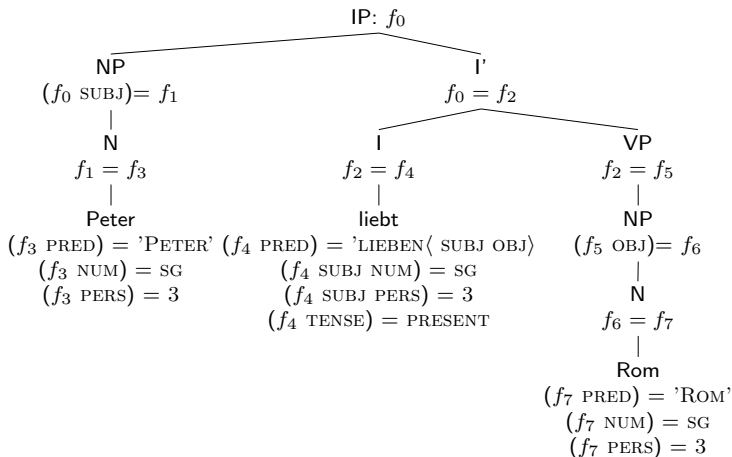
2.b Instantiierung

- a. Weise dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation eine FS-Variable zu.
- b. Jedes \uparrow bzw. \downarrow wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.



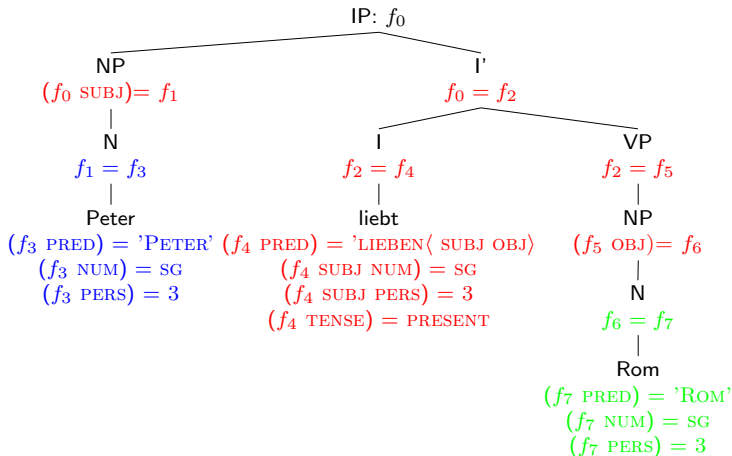
2.b Instantiierung

- a. Weise dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation eine FS-Variable zu.
- b. Jedes \uparrow bzw. \downarrow wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.



3. Resolution der funktionalen Gleichungen

Aufsummieren der funktionalen Gleichungen



3. Resolution der funktionalen Gleichungen

Die funktionalen Gleichungen müssen konsistent sein und die Wohlgeformtheitsbedingungen für F-Strukturen erfüllen.

Die F-Struktur muss minimal sein.

$$f_0 = f_2$$

$$f_2 = f_4$$

$$f_2 = f_5$$

$$(f_4 \text{ PRED}) = \text{'LIEBEN' } \langle \text{SUBJ OBJ} \rangle$$

$$(f_4 \text{ SUBJ NUM}) = \text{SG}$$

$$(f_4 \text{ SUBJ PERS}) = 3$$

$$(f_4 \text{ TENSE}) = \text{PRESENT}$$

$$(f_0 \text{ SUBJ}) = f_1$$

$$f_1 = f_3$$

$$(f_3 \text{ PRED}) = \text{'PETER'}$$

$$(f_3 \text{ NUM}) = \text{SG}$$

$$(f_3 \text{ PERS}) = 3$$

$$(f_5 \text{ OBJ}) = f_6$$

$$f_6 = f_7$$

$$(f_7 \text{ PRED}) = \text{'ROM'}$$

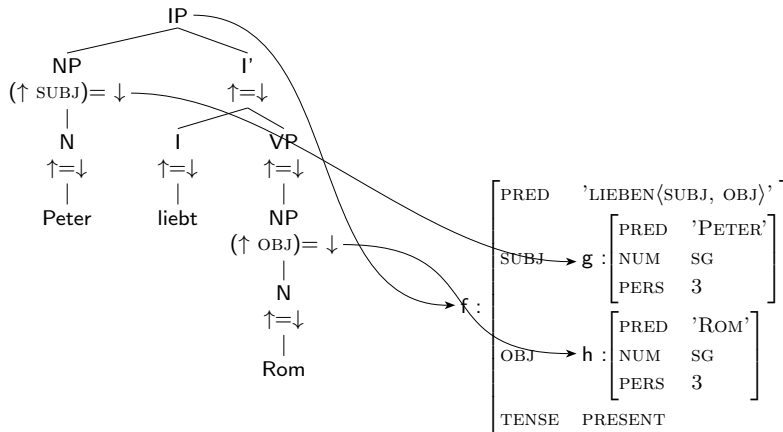
$$(f_7 \text{ NUM}) = \text{SG}$$

$$(f_7 \text{ PERS}) = 3$$

f_0, f_2, f_4, f_5

PRED	'LIEBEN⟨SUBJ, OBJ⟩'			
SUBJ	f_1, f_3	PRED	'PETER'	
		NUM	SG	
		PERS	3	
OBJ	f_6, f_7	PRED	'ROM'	
		NUM	SG	
		PERS	3	
TENSE	PRESENT			

Funktionale Abbildung von C- zu F-Struktur



PRED	'LIEBEN(SUBJ, OBJ)'								
SUBJ	g :	<table> <tr> <td>PRED</td> <td>'PETER'</td> </tr> <tr> <td>NUM</td> <td>SG</td> </tr> <tr> <td>PERS</td> <td>3</td> </tr> </table>		PRED	'PETER'	NUM	SG	PERS	3
PRED	'PETER'								
NUM	SG								
PERS	3								
OBJ	h :	<table> <tr> <td>PRED</td> <td>'ROM'</td> </tr> <tr> <td>NUM</td> <td>SG</td> </tr> <tr> <td>PERS</td> <td>3</td> </tr> </table>		PRED	'ROM'	NUM	SG	PERS	3
PRED	'ROM'								
NUM	SG								
PERS	3								
TENSE	PRESENT								

1 Annotation:

Annotiere den C-Struktur-Baum mit den funktionalen Gleichungen der verwendeten Phrasenstrukturregeln und Lexikoneinträge.

2 Instantiierung:

- Weise dem Wurzelknoten und jedem Knoten mit einer \downarrow -Annotation eine (jeweils neue) FS-Variable zu.
D.h. jeder Knoten, der auf eine partielle F-Struktur abzubilden ist, erhält eine eindeutige F-Struktur-Variable.
- Jedes \uparrow bzw. \downarrow wird ersetzt durch den Index (die FS-Variable) des Mutterknotens bzw. des lokalen Knotens.

3 Konstruktion einer minimalen F-Struktur:

- Die funktionalen Gleichungen (Constraints) müssen *konsistent* sein und die *Wohlgeformtheitsbedingungen* für FS erfüllen.
- Die F-Struktur muss *minimal* sein: Es ist die *kleinste F-Struktur*, die alle Gleichungen erfüllt und keine zusätzlichen Attribut-Wert-Paare enthält.

Kursziele: Sie werden lernen ...

I. ... die Struktur von (deutschen) Sätzen zu analysieren
durch

- Zerlegen in Konstituenten
- Zuweisung ihrer entsprechenden grammatischen Funktionen
- sowie die Beziehung von Syntax zu Morphologie und Semantik

II. ... wie Sie dies alles in einer formalen Grammatik definieren

- durch Grammatikregeln und Lexikoneinheiten in einem Grammatikformalismus
- unser Formalismus: LFG (Lexical-Functional Grammar)

III. ... zu erklären, wie ein Algorithmus für diese Theorie für gegebene Sätze eine syntaktische Analyse erzeugt

IV. . . in einem lauffähigen System (XLE) ein eigenes (kleines) LFG Grammatikfragment zu definieren und zur Analyse einzusetzen

und dabei

- den Formalismus und bestimmte syntaktische Konstruktionen besser verstehen lernen,
- verstehen, wie Ambiguitäten entstehen
- und wie man sie (manchmal) filtern kann.

A glimpse at real grammar engineering: XLE-Web

LFG Grammar Engineering and LFG-based Parsing and Generation

- ParGram / ParSem Tools and Demos page:

<https://pargram.b.uib.no/tools/>

- **XLE-Web:** <http://clarino.uib.no/iness/xle-web>

Parse sentences in many (common and exotic) languages and view their c- and f-structures with correspondences.

Theorie der LFG und Grammatikarchitektur

- Bresnan 2001: *Lexical-Functional Syntax*, Ch. 4.1 – 4.7.
- Dalrymple 2001: *Lexical-Functional Grammar*, Ch.5, Section 3.1, 3.2
- Falk, 2001: *Lexical-Functional Grammar*, Ch. 1