

# Formale Syntax: HPSG

## 07. Konstituentenreihenfolge und Verbbewegung

Roland Schäfer

Institut für Germanistische Sprachwissenschaft  
Friedrich-Schiller-Universität Jena

Stets aktuelle Fassungen: <https://github.com/rsling/VL-HPSG>

Basiert teilweise auf Folien von Stefan Müller: <https://hpsg.hu-berlin.de/~stefan/Lehre/S2021/hpsg.html>

Grundlage ist Stefans HPSG-Buch: <https://hpsg.hu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html.de>

Stefan trägt natürlich keinerlei Verantwortung für meine Fehler und Missverständnisse!

## Übersicht

- 1 Phrasenstruktur und Phrasenstrukturgrammatiken
- 2 Merkmalstrukturen und Merkmalbeschreibungen
- 3 Komplementation und Grammatikregeln
- 4 Verbsemantik und Linking (Semantik 1)
- 5 Adjunktion und Spezifikation
- 6 Lexikon und Lexikonregeln
- 7 Konstituentenreihenfolge und Verbbewegung
- 8 Nicht-lokale Abhängigkeiten und Vorfelddbesetzung
- 9 Quantorenspeicher (Semantik 2)
- 10 Unterspezifikationssemantik (Semantik 3)

<https://rolandschaefer.net/archives/2805>

<https://github.com/rsling/VL-HPSG/tree/main/output>

<https://hpsg.hu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html>

# Einleitung

# Meditieren Sie fünf Minuten!

# Meditieren Sie fünf Minuten!



M. C. Escher, *Wasserfall*, Lithografie, 1961.

[https://en.wikipedia.org/wiki/File:Escher\\_Waterfall.jpg](https://en.wikipedia.org/wiki/File:Escher_Waterfall.jpg)

# Das war die Vorbereitung

Sie sind jetzt bereit für den schönsten Lexikoneintrag überhaupt!



Sie sind jetzt bereit für den schönsten Lexikoneintrag überhaupt!

$$\left[ \begin{array}{ll} \text{PHON} & \langle \rangle \\ \text{LOC} & \boxed{1} \left[ \text{CAT} | \text{HEAD} | \text{DSL} \quad \boxed{1} \right] \end{array} \right]$$

# Satzsyntax, und zwar auch noch deutsche!

Über Konstituentenstellung müssen wir sowieso noch reden!

Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?

Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?
- Wie stellt man sicher, dass Köpfe entweder initial oder final in ihrer Phrase stehen?

Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?
- Wie stellt man sicher, dass Köpfe entweder initial oder final in ihrer Phrase stehen?
- Wie kommt das Verb in die „linke Satzklammer“?

Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?
- Wie stellt man sicher, dass Köpfe entweder initial oder final in ihrer Phrase stehen?
- Wie kommt das Verb in die „linke Satzklammer“?
- Im Gegensatz zu Stefan finde ich seine Analyse für Verbbewegung total gut zu verstehen.

Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?
- Wie stellt man sicher, dass Köpfe entweder initial oder final in ihrer Phrase stehen?
- Wie kommt das Verb in die „linke Satzklammer“?
- Im Gegensatz zu Stefan finde ich seine Analyse für Verbbewegung total gut zu verstehen.
- Ich lasse allerdings auch einiges von seiner Argumentation weg.



Über Konstituentenstellung müssen wir sowieso noch reden!

- Wie lizenziert man die freie Konstituentenstellung im Mittelfeld?
- Wie stellt man sicher, dass Köpfe entweder initial oder final in ihrer Phrase stehen?
- Wie kommt das Verb in die „linke Satzklammer“?
- Im Gegensatz zu Stefan finde ich seine Analyse für Verbbewegung total gut zu verstehen.
- Ich lasse allerdings auch einiges von seiner Argumentation weg.

Müller (2013: Kapitel 9)



# Scrambling

Das Verb steht immer **rechts**. Der Rest macht, was er darf.

# Scrambling

Das Verb steht immer **rechts**. Der Rest macht, was er darf.

- (1) während Otje [den Artikel] **liest**
- (2) während [den Artikel] Otje **liest**
- (3) während Otje [den Artikel] schnell **liest**
- (4) während Otje schnell [den Artikel] **liest**
- (5) während [den Artikel] Otje schnell **liest**
- (6) während [den Artikel] schnell Otje **liest**
- (7) während schnell Otje [den Artikel] **liest**
- (8) während schnell [den Artikel] Otje **liest**

# Scrambling

Das Verb steht immer **rechts**. Der Rest macht, was er darf.

- (1) während Otje [den Artikel] **liest**
- (2) während [den Artikel] Otje **liest**
- (3) während Otje [den Artikel] schnell **liest**
- (4) während Otje schnell [den Artikel] **liest**
- (5) während [den Artikel] Otje schnell **liest**
- (6) während [den Artikel] schnell Otje **liest**
- (7) während schnell Otje [den Artikel] **liest**
- (8) während schnell [den Artikel] Otje **liest**
- (9) \* während Otje **liest** [den Artikel]
- (10) \* während [den Artikel] **liest** Otje
- (11) \* während **liest** Otje [den Artikel]
- (12) \* während **liest** [den Artikel] Otje



Sie kennen es aus CP/IP-Ansätzen. Darüber ist man sich auch einig.

Sie kennen es aus CP/IP-Ansätzen. Darüber ist man sich auch einig.

- (13)  $\text{Liest}_1$  [Otje den Artikel  $t_1$ ]?
- (14)  $\text{Otje}_2$   $\text{liest}_1$  [ $t_2$  den Artikel  $t_1$ ].
- (15)  $\text{Den Artikel}_2$   $\text{liest}_1$  [Otje  $t_2$   $t_1$ ].
- (16)  $\text{Was}_2$   $\text{glaubst}_1$  [du [dass Otje  $t_2$  gelesen hat]  $t_1$ ]?





Das für mich wichtigste Argument für binäre Verzweigung in der VP.

Das für mich wichtigste Argument für binäre Verzweigung in der VP.

(17) während [Otje [morgen [den Artikel [schnell [lesen [müssen wird]]]]]]]

(18) Wird [Otje [morgen [den Artikel [schnell [lesen [müssen  $t_1$ ]]]]]]?

(19) [Lesen [müssen  $t_1$ ]] wird [Otje [morgen [den Artikel [schnell  $t_2$ ]]]].

(20) [Schnell [lesen [müssen  $t_1$ ]]] wird [Otje [morgen [den Artikel  $t_2$ ]]].

(21) [Den Artikel [schnell [lesen [müssen  $t_1$ ]]]] wird [Otje [morgen  $t_2$ ]].

(22) [Morgen [den Artikel [schnell [lesen [müssen  $t_1$ ]]]]] wird [Otje  $t_2$ ].

Das für mich wichtigste Argument für binäre Verzweigung in der VP.

- (17) während [Otje [morgen [den Artikel [schnell [lesen [müssen wird]]]]]]]
- (18) Wird [Otje [morgen [den Artikel [schnell [lesen [müssen  $t_1$ ]]]]]]?
- (19) [Lesen [müssen  $t_1$ ]] wird [Otje [morgen [den Artikel [schnell  $t_2$ ]]]].
- (20) [Schnell [lesen [müssen  $t_1$ ]]] wird [Otje [morgen [den Artikel  $t_2$ ]]].
- (21) [Den Artikel [schnell [lesen [müssen  $t_1$ ]]]] wird [Otje [morgen  $t_2$ ]].
- (22) [Morgen [den Artikel [schnell [lesen [müssen  $t_1$ ]]]]] wird [Otje  $t_2$ ].

So haben wir jeweils eine **Konstituente**, die ins Vorfeld bewegt werden kann.

## Konstituententellung in HPSG

# Freie Reihenfolge bei der Komplementation

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\textit{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \\ \text{HD-DTR|LOC|CAT|SUBCAT} \\ \text{NHD-DTR} \end{array} \begin{array}{l} \boxed{1} \\ \boxed{1} \oplus \langle \boxed{2} \rangle \\ \boxed{2} \end{array} \right]$$



# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\textit{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{HD-DTR|LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \oplus \boxed{3} \\ \text{NHD-DTR} \quad \boxed{2} \end{array} \right]$$

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\textit{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \boxed{2} \\ \text{HD-DTR|LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \\ \text{NHD-DTR} \quad \boxed{2} \end{array} \right]$$

Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\textit{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{HD-DTR|LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \oplus \boxed{3} \\ \text{NHD-DTR} \quad \boxed{2} \end{array} \right]$$

Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

- Sowohl  $\boxed{1}$  als auch  $\boxed{3}$  können leer sein.

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\text{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \\ \text{HD-DTR|LOC|CAT|SUBCAT} \\ \text{NHD-DTR} \end{array} \quad \begin{array}{l} \boxed{1} \oplus \boxed{2} \\ \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \\ \boxed{2} \end{array} \right]$$

Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

- Sowohl  $\boxed{1}$  als auch  $\boxed{3}$  können leer sein.
- Wenn  $\boxed{1}$  leer: erstes Argument abbinden

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\text{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{HD-DTR|LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \oplus \boxed{3} \\ \text{NHD-DTR} \quad \boxed{2} \end{array} \right]$$

Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

- Sowohl  $\boxed{1}$  als auch  $\boxed{3}$  können leer sein.
- Wenn  $\boxed{1}$  leer: erstes Argument abbinden
- Wenn  $\boxed{3}$  leer: letztes Argument abbinden (= alte Version)

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\textit{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{HD-DTR|LOC|CAT|SUBCAT} \quad \boxed{1} \oplus \langle \boxed{2} \rangle \oplus \boxed{3} \\ \text{NHD-DTR} \quad \boxed{2} \end{array} \right]$$

Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

- Sowohl  $\boxed{1}$  als auch  $\boxed{3}$  können leer sein.
- Wenn  $\boxed{1}$  leer: erstes Argument abbinden
- Wenn  $\boxed{3}$  leer: letztes Argument abbinden (= alte Version)
- Wenn  $\boxed{1}$  und  $\boxed{3}$  leer: intransitives Verb

# Freie Reihenfolge bei der Komplementation

Argumente können in beliebiger Reihenfolge saturiert werden.

$$\text{head-argument-phrase} \Rightarrow \left[ \begin{array}{l} \text{LOC|CAT|SUBCAT} \\ \text{HD-DTR|LOC|CAT|SUBCAT} \\ \text{NHD-DTR} \end{array} \quad \begin{array}{l} \boxed{1} \oplus \langle \boxed{2} \rangle \\ \boxed{1} \oplus \langle \boxed{2} \rangle \oplus \boxed{3} \\ \boxed{2} \end{array} \right]$$

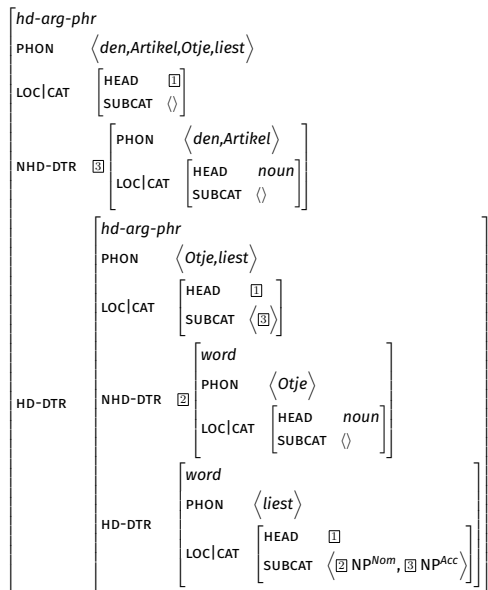
Statt des letzten Arguments irgendeins ( $\boxed{2}$ ) abbinden:

- Sowohl  $\boxed{1}$  als auch  $\boxed{3}$  können leer sein.
- Wenn  $\boxed{1}$  leer: erstes Argument abbinden
- Wenn  $\boxed{3}$  leer: letztes Argument abbinden (= alte Version)
- Wenn  $\boxed{1}$  und  $\boxed{3}$  leer: intransitives Verb
- Der neue Pfad über LOC wird unten motiviert.





# Mögliche Struktur | Subjekt zuerst abgebunden





Reihenfolge der Argumentabbildung: nur **Dominanz**

Reihenfolge der Argumentabbildung: nur **Dominanz**

Abfolge der Konstituenten in Struktur: **Präzedenz**

Links- und Rechtsköpfigkeit sind nicht modellierbar durch Dominanz!

Reihenfolge der Argumentabbildung: nur **Dominanz**

Abfolge der Konstituenten in Struktur: **Präzedenz**

Links- und Rechtsköpfigkeit sind nicht modellierbar durch Dominanz!

So sollte es sein:

Reihenfolge der Argumentabbildung: nur **Dominanz**

Abfolge der Konstituenten in Struktur: **Präzedenz**

Links- und Rechtsköpfigkeit sind nicht modellierbar durch Dominanz!

So sollte es sein:

$$\left[ \begin{array}{ll} \text{head-arg-phr} & \\ \text{PHON} & \boxed{1} \oplus \boxed{2} \\ \text{HD-DTR|PHON} & \boxed{1} \langle in \rangle \\ \text{NHD-DTR|PHON} & \boxed{2} \langle dem, Buch \rangle \end{array} \right]$$

Reihenfolge der Argumentabbildung: nur **Dominanz**

Abfolge der Konstituenten in Struktur: **Präzedenz**

Links- und Rechtsköpfigkeit sind nicht modellierbar durch Dominanz!

So sollte es sein:

|  |   |
|--|---|
| $\left[ \begin{array}{ll} \textit{head-arg-phr} & \\ \text{PHON} & \boxed{1} \oplus \boxed{2} \\ \text{HD-DTR PHON} & \boxed{1} \langle \textit{in} \rangle \\ \text{NHD-DTR PHON} & \boxed{2} \langle \textit{dem, Buch} \rangle \end{array} \right]$ | $\left[ \begin{array}{ll} \textit{head-arg-phr} & \\ \text{PHON} & \boxed{2} \oplus \boxed{1} \\ \text{HD-DTR PHON} & \boxed{1} \langle \textit{liest} \rangle \\ \text{NHD-DTR PHON} & \boxed{2} \langle \textit{das, Buch} \rangle \end{array} \right]$ |
|--|---|





Achtung | Wir erlauben bisher **zu viele Strukturen**, nicht zu wenige!

Achtung | Wir erlauben bisher **zu viele Strukturen**, nicht zu wenige!

|  |  |
|--|--|
| $\left[ \begin{array}{l} \text{head-arg-phr} \\ \text{PHON} \quad \langle in, dem, Buch \rangle \\ \text{HD-DTR}   \text{PHON} \quad \boxed{1} \langle in \rangle \\ \text{NHD-DTR}   \text{PHON} \quad \boxed{2} \langle dem, Buch \rangle \end{array} \right]$ |  |
|--|--|

Achtung | Wir erlauben bisher **zu viele Strukturen**, nicht zu wenige!

$$\left[ \begin{array}{l} \text{head-arg-phr} \\ \text{PHON} \quad \langle \text{in}, \text{dem}, \text{Buch} \rangle \\ \text{HD-DTR} | \text{PHON} \quad \boxed{1} \langle \text{in} \rangle \\ \text{NHD-DTR} | \text{PHON} \quad \boxed{2} \langle \text{dem}, \text{Buch} \rangle \end{array} \right]$$
$$\left[ \begin{array}{l} \text{head-arg-phr} \\ \text{PHON} \quad \langle \text{dem}, \text{Buch}, \text{in} \rangle \\ \text{HD-DTR} | \text{PHON} \quad \boxed{1} \langle \text{in} \rangle \\ \text{NHD-DTR} | \text{PHON} \quad \boxed{2} \langle \text{dem}, \text{Buch} \rangle \end{array} \right]$$

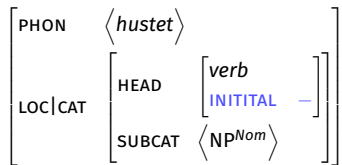
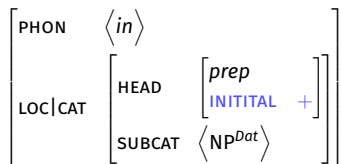


Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.

Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.

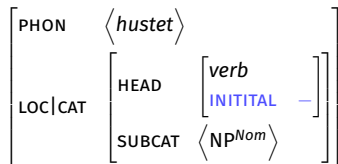
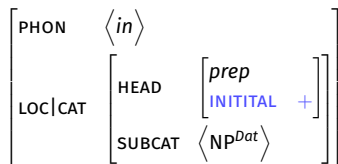
$$\left[ \begin{array}{cc} \text{PHON} & \langle in \rangle \\ \text{LOC|CAT} & \left[ \begin{array}{cc} \text{HEAD} & \left[ \begin{array}{c} prep \\ \text{INITIAL} \end{array} \right] + \\ \text{SUBCAT} & \langle NP^{Dat} \rangle \end{array} \right] \end{array} \right]$$

Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



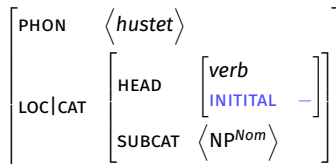
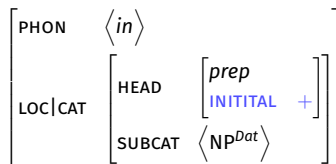
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.

Linearisierungsregeln legen fest,  
wie PHON konkateniert wird.





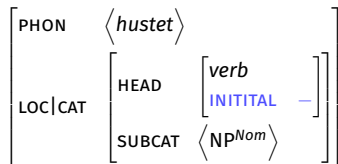
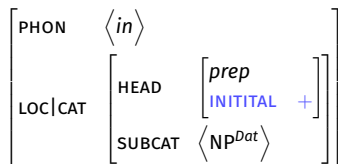
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.

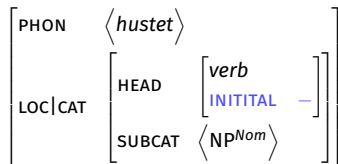
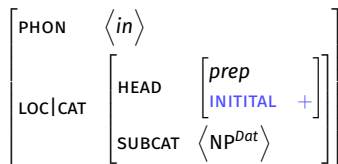
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.
- $\text{Head}[\text{INIT } +] < \text{Argument}$

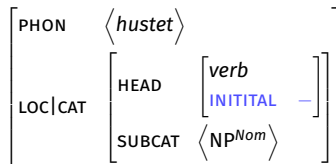
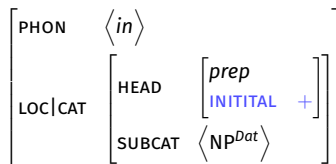
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.
- $\text{Head}[\text{INIT} +] < \text{Argument}$
- $\text{Argument} < \text{Head}[\text{INIT} -]$

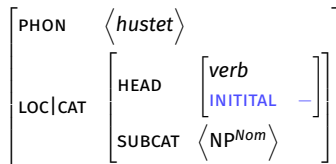
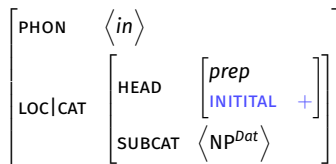
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.
- $\text{Head}[\text{INIT} +] < \text{Argument}$
- $\text{Argument} < \text{Head}[\text{INIT} -]$
- $\text{Adjunct}[\text{PRE-MOD} +] < \text{Head}$

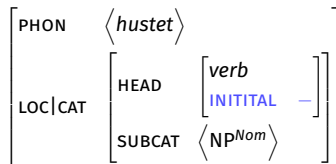
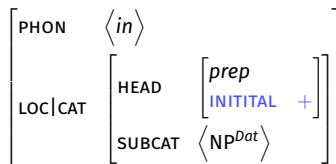
Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.
- Head[INIT +] < Argument
- Argument < Head[INIT -]
- Adjunct[PRE-MOD +] < Head
- Head < Adjunct[PRE-MOD -]

Köpfe wissen selbst, ob sie **initial** oder **final** stehen müssen.



Linearisierungsregeln legen fest, wie PHON konkateniert wird.

- Mit  $<$  wird festgelegt, welches PHON zuerst kommt.
- Head[INIT +] < Argument
- Argument < Head[INIT -]
- Adjunct[PRE-MOD +] < Head
- Head < Adjunct[PRE-MOD -]
- Specifier < Head

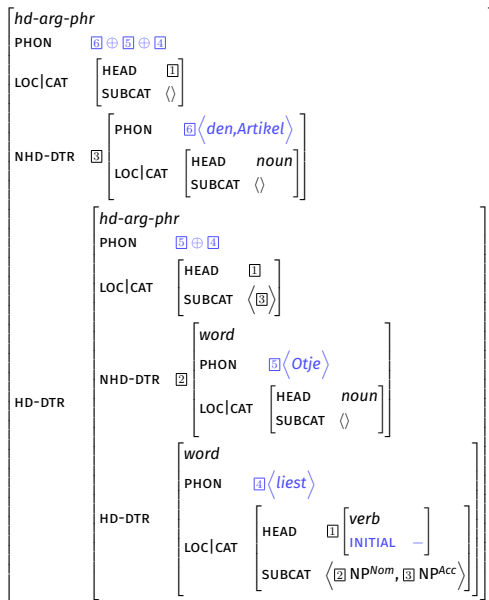
# Verben stehen also unabhängig von der Dominanzfolge VP-final

# Verben stehen also unabhängig von der Dominanzfolge VP-final



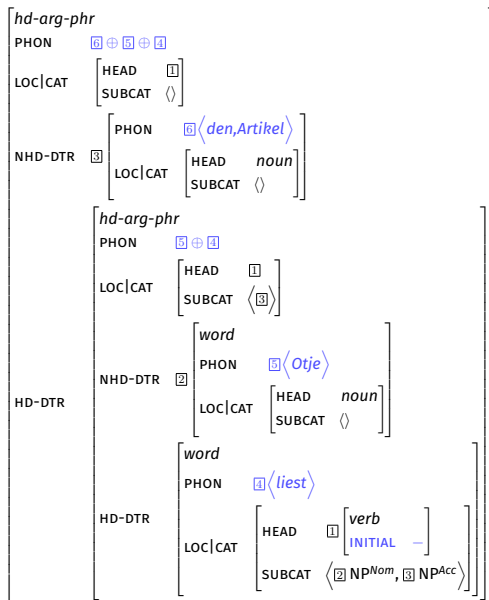


# Verben stehen also unabhängig von der Dominanzfolge VP-final



Ohne Linearisierungsregeln möglich:

# Verben stehen also unabhängig von der Dominanzfolge VP-final



Ohne Linearisierungsregeln möglich:

- (23) \* während liest Otje den Artikel
- (24) \* während liest den Artikel Otje
- (25) \* während Otje liest den Artikel
- (26) \* während den Artikel liest Otje

Verbbewegung



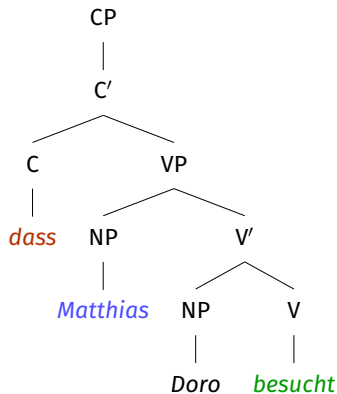
# Wiederholung | Bewegungstransformationen

**Bewegung** | Erklärt **Abhängigkeiten** zwischen Positionen in Strukturen.

*Transformationen* sagt man seit der GB-Theorie nicht mehr. Technisch gesehen sind es Transformationen.

**Bewegung** | Erklärt **Abhängigkeiten** zwischen Positionen in Strukturen.

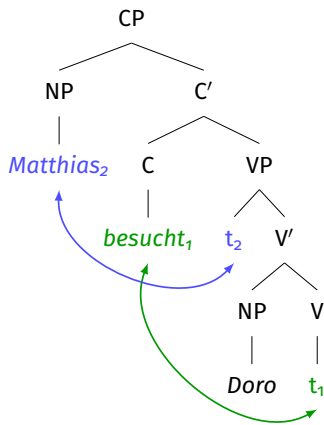
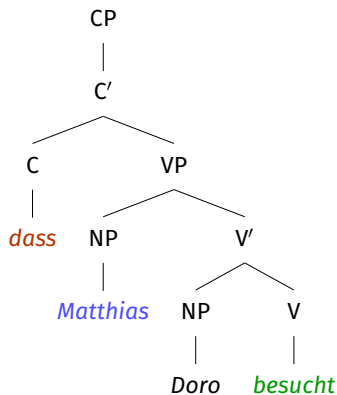
*Transformationen* sagt man seit der GB-Theorie nicht mehr. Technisch gesehen sind es Transformationen.



# Wiederholung | Bewegungstransformationen

**Bewegung** | Erklärt **Abhängigkeiten** zwischen Positionen in Strukturen.

*Transformationen* sagt man seit der GB-Theorie nicht mehr. Technisch gesehen sind es Transformationen.





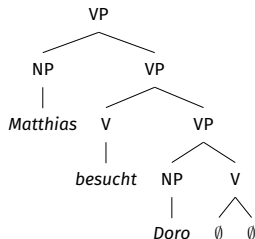


HPSG | Die gleichen Abhängigkeiten ohne Bewegung, dafür mit Strukturteilung

Aber nicht unbedingt ohne leere Elemente.

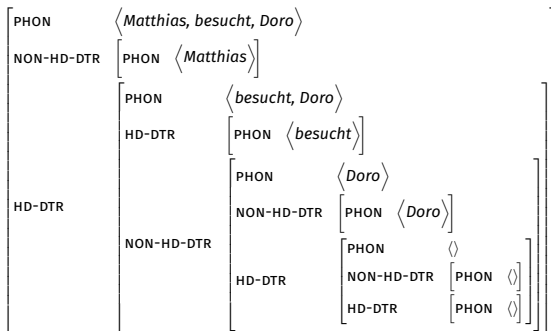
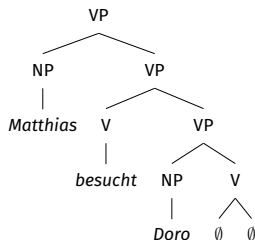
## HPSG | Die gleichen Abhängigkeiten ohne Bewegung, dafür mit Strukturteilung

Aber nicht unbedingt ohne leere Elemente.



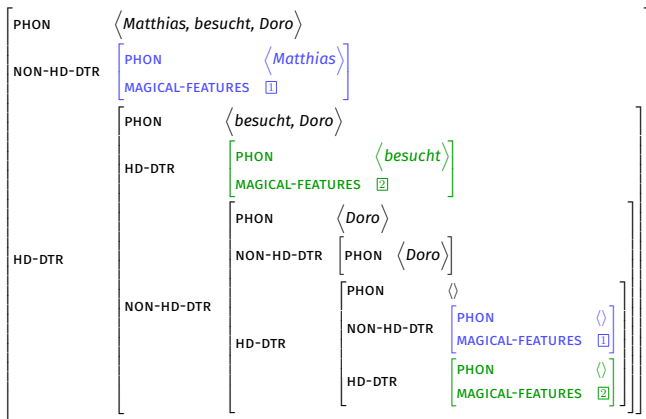
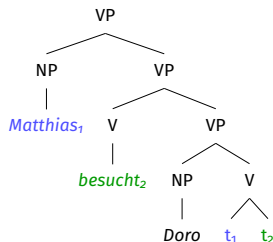
## HPSG | Die gleichen Abhängigkeiten ohne Bewegung, dafür mit Strukturteilung

Aber nicht unbedingt ohne leere Elemente.



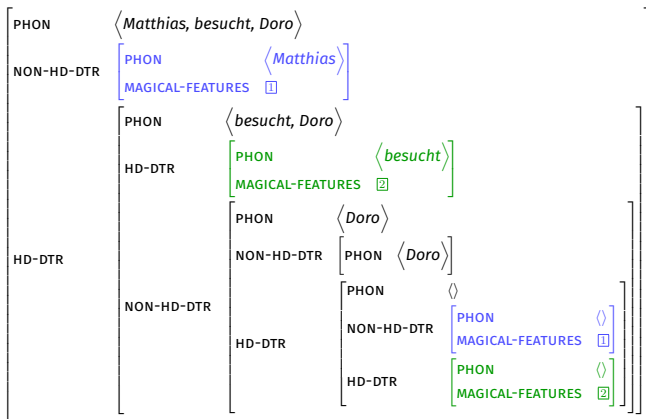
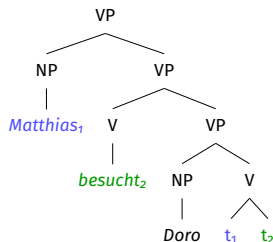
## HPSG | Die gleichen Abhängigkeiten ohne Bewegung, dafür mit Strukturteilung

Aber nicht unbedingt ohne leere Elemente.



## HPSG | Die gleichen Abhängigkeiten ohne Bewegung, dafür mit Strukturteilung

Aber nicht unbedingt ohne leere Elemente.



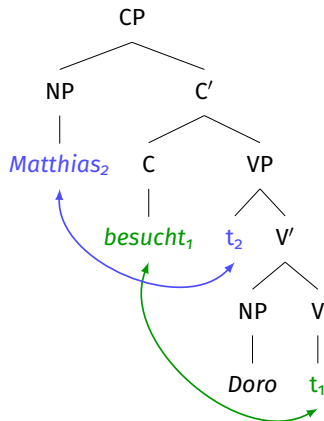
Heute klären wir für das bewegte Verb, wie die Merkmalsmagie funktioniert.



Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

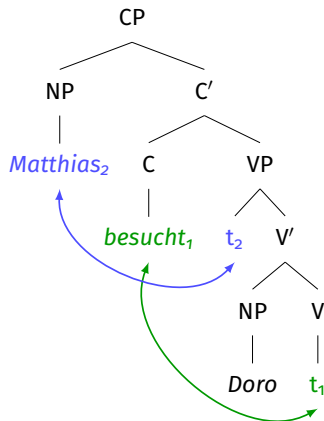




# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

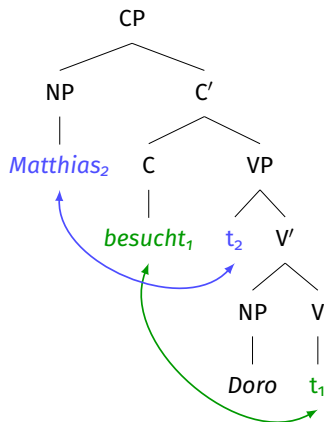
Probleme



# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

## Probleme

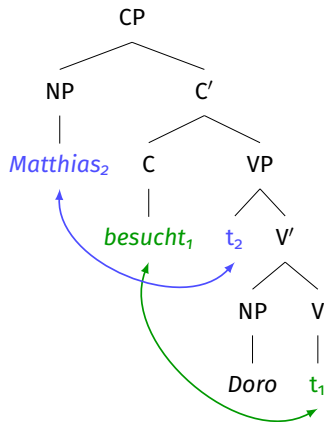


- Koindizierte bewegte Elemente und Spuren müssen eine Kette (*chain*) bilden.

# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

## Probleme

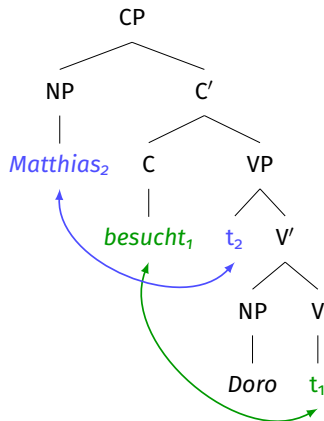


- Koindizierte bewegte Elemente und Spuren müssen eine Kette (*chain*) bilden.
- Dazu muss der Formalismus sie einander zuordnen.

# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

## Probleme

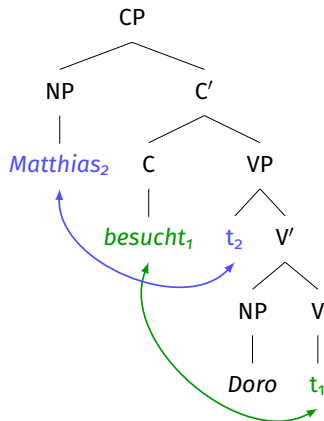


- Koindizierte bewegte Elemente und Spuren müssen eine Kette (*chain*) bilden.
- Dazu muss der Formalismus sie einander zuordnen.
- Aus Sicht des bewegten Elements muss **die gesamte c-Kommando-Domäne durchsucht werden**.

# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

## Probleme

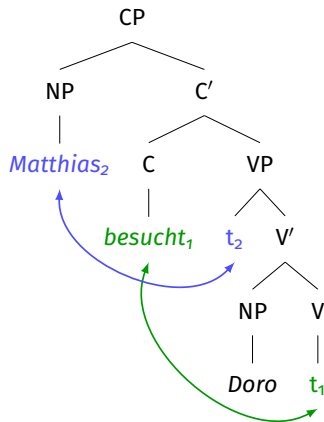


- Koindizierte bewegte Elemente und Spuren müssen eine Kette (*chain*) bilden.
- Dazu muss der Formalismus sie einander zuordnen.
- Aus Sicht des bewegten Elements muss **die gesamte c-Kommando-Domäne durchsucht werden**.
- Der Baum kann beliebig komplex sein, es gibt **kein einfaches Rezept für die Suche** (in der Art von: *aufwärts, dann abwärts: rechts, rechts, links*).

# Nicht streng lokale Theorien

Auch bei *Bewegung* geht es letztlich darum, wie die magischen Merkmale in der Struktur einander zugeordnet werden können.

## Probleme



- Koindizierte bewegte Elemente und Spuren müssen eine Kette (*chain*) bilden.
- Dazu muss der Formalismus sie einander zuordnen.
- Aus Sicht des bewegten Elements muss **die gesamte c-Kommando-Domäne durchsucht werden**.
- Der Baum kann beliebig komplex sein, es gibt **kein einfaches Rezept für die Suche** (in der Art von: *aufwärts, dann abwärts: rechts, rechts, links*).
- **Bäume** und **Baumdurchsuchungen** machen solche Theorien unnötig komplex und komputational nachteilig.



# Streng lokale Theorien

In einer lokalen Theorie müssen die relevanten Informationen  
am jeweiligen „Knoten“ verfügbar sein. Man durchsucht keine Bäume!



# Streng lokale Theorien

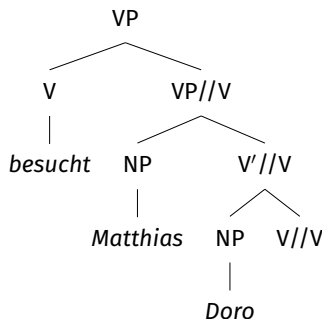
In einer lokalen Theorie müssen die relevanten Informationen  
am jeweiligen „Knoten“ verfügbar sein. Man durchsucht keine Bäume!

Die Information, dass etwas fehlt, wird an geeigneter Stelle von Knoten zu Knoten weitergegeben. Hier steht hinter dem Doubleslash jeweils, was fehlt.

# Streng lokale Theorien

In einer lokalen Theorie müssen die relevanten Informationen  
am jeweiligen „Knoten“ verfügbar sein. Man durchsucht keine Bäume!

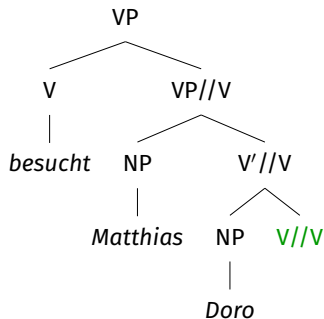
Die Information, dass etwas fehlt, wird an geeigneter Stelle von Knoten zu Knoten  
weitergegeben. Hier steht hinter dem Doubleslash jeweils, was fehlt.



# Streng lokale Theorien

In einer lokalen Theorie müssen die relevanten Informationen  
am jeweiligen „Knoten“ verfügbar sein. Man durchsucht keine Bäume!

Die Information, dass etwas fehlt, wird an geeigneter Stelle von Knoten zu Knoten  
weitergegeben. Hier steht hinter dem Doubleslash jeweils, was fehlt.



[ PHON <>  
LOC 1 [CAT|HEAD|DSL 1] ]

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.

# Ausblick auf die Modellierung

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL** ist strukturgeteilt mit ihrem **LOC**.



Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.

# Ausblick auf die Modellierung

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL** ist strukturgeteilt mit ihrem **LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).
  - ▶ Er ist INITIAL +, weil er links von der VP steht.

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).
  - ▶ Er ist INITIAL +, weil er links von der VP steht.
  - ▶ Auf seiner SUBCAT steht eine VP, deren LOC|CAT|HEAD|DSL ...

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).
  - ▶ Er ist INITIAL +, weil er links von der VP steht.
  - ▶ Auf seiner SUBCAT steht eine VP, deren LOC|CAT|HEAD|DSL ...
  - ▶ ... mit dem LOC seiner LEX-DTR token-identisch ist.



Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).
  - ▶ Er ist INITIAL +, weil er links von der VP steht.
  - ▶ Auf seiner SUBCAT steht eine VP, deren LOC|CAT|HEAD|DSL ...
  - ▶ ... mit dem LOC seiner LEX-DTR token-identisch ist.
  - ▶ **Dadurch wird die gesamte Syntax und Semantik des lexikalischen Verbs durch die Knoten, deren Kopf die Verbspur ist, in die Verbspur gepumpt.**

Um **Verbbewegung** zu modellieren, brauchen wir keine neuen Regeln, sondern:

- 1 Eine Zusammenfassung von CAT und CONT zu **LOCAL** bzw. **LOC**.
- 2 Eine **lexikalische Verbspur** für alle Verben
  - ▶ Ihr PHON ist eine leere Liste.
  - ▶ Ihr Kopfmerkmal DOUBLESASH bzw. **DSL ist strukturgeteilt mit ihrem LOC**.
  - ▶ Mit DSL kodiert sie, was fehlt (also das lexikalische Verb selbst).
  - ▶ Sie ist INITIAL – wie normale Verben.
  - ▶ Als Kopfmerkmal wird LOC|CAT|HEAD|DSL in Kopf-Strukturen (VP) weitergegeben.
- 3 Einen Lexikoneintrag per **Lexikonregel für das bewegte Verb**
  - ▶ Sein PHON entspricht dem seiner LEX-DTR (normales Verb).
  - ▶ Er ist INITIAL +, weil er links von der VP steht.
  - ▶ Auf seiner SUBCAT steht eine VP, deren LOC|CAT|HEAD|DSL ...
  - ▶ ... mit dem LOC seiner LEX-DTR token-identisch ist.
  - ▶ **Dadurch wird die gesamte Syntax und Semantik des lexikalischen Verbs durch die Knoten, deren Kopf die Verbspur ist, in die Verbspur gepumpt.**
  - ▶ **Die Verbspur muss daher nicht verbspezifisch sein!**

# Analyse (ausnahmsweise als Baum)

# Analyse (ausnahmsweise als Baum)

V1-Satz | *Hustet Matthias?*

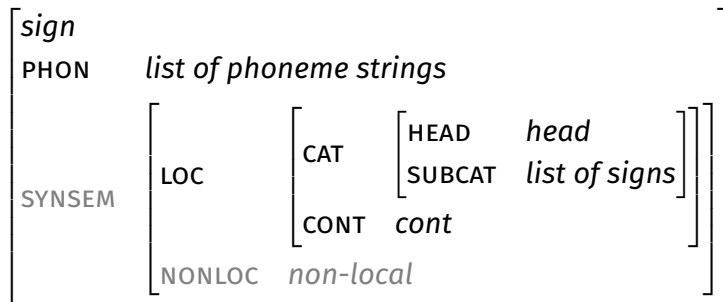
## V1-Satz | Hustet Matthias?



## Details der Analyse

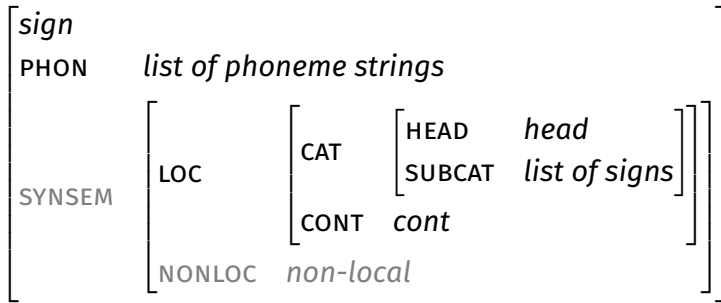
# Neue Merkmalgeometrie für Zeichen

# Neue Merkmalgeometrie für Zeichen



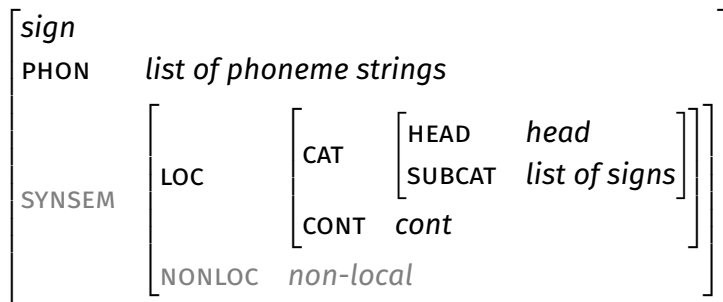


# Neue Merkmalgeometrie für Zeichen



NONLOC brauchen wir nächste Woche.

# Neue Merkmalgeometrie für Zeichen



**NONLOC** brauchen wir nächste Woche.

**SYNSEM** brauchen wir bei der Lektüre von Pollard & Sag (1994).

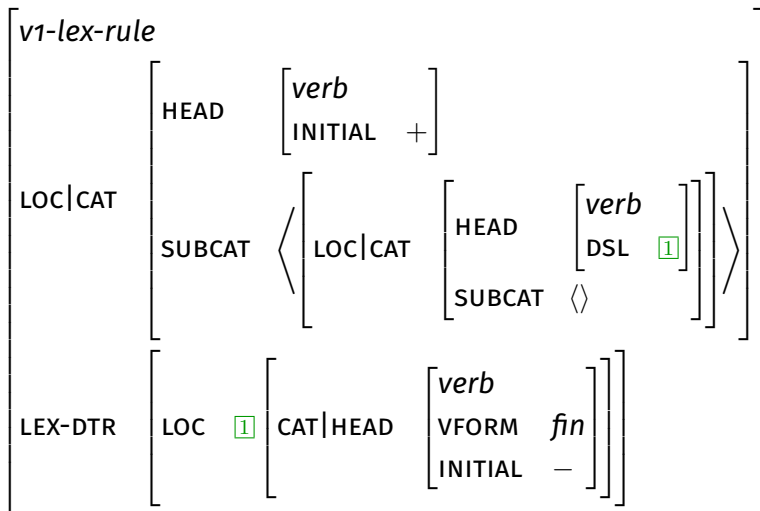
# Lexikonregel für Verben in Nicht-Letzt-Stellung

# Lexikonregel für Verben in Nicht-Letzt-Stellung

Die Regel bildet eine Verb wie *hustet*, das eine VP verlangt, in der es selbst „fehlt“ (DSL).

# Lexikonregel für Verben in Nicht-Letzt-Stellung

Die Regel bildet eine Verb wie *hustet*, das eine VP verlangt, in der es selbst „fehlt“ (DSL).





Verhindert ansonsten mögliche Verbdopplung | *\*Hustet Matthias hustet?*

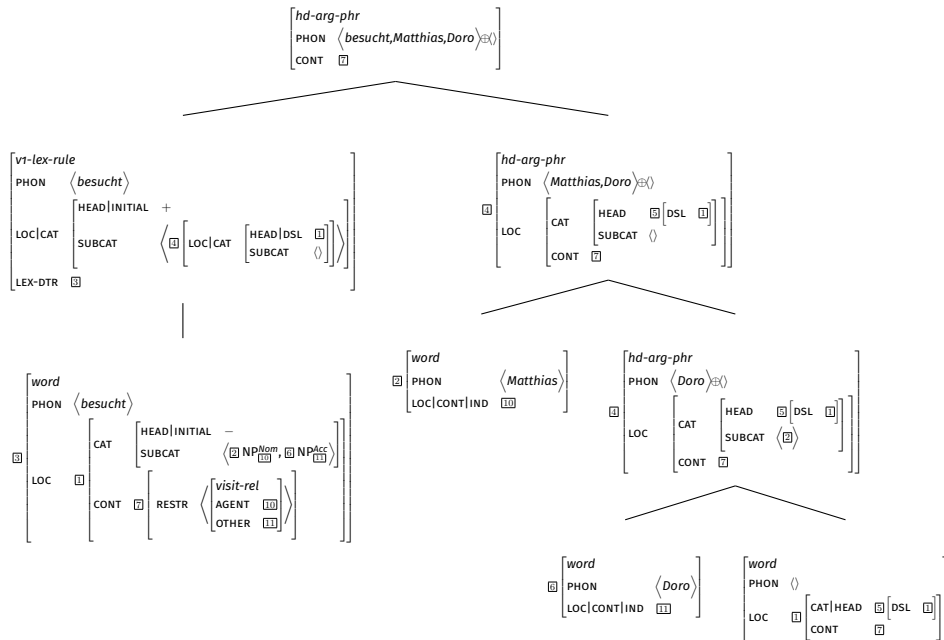
Verhindert ansonsten mögliche Verbdopplung | *\*Hustet Matthias hustet?*

$$\left[ \text{HD-DTR} \left[ \begin{array}{l} \text{word} \\ \text{PHON} \end{array} \text{ non-empty-list} \right] \right] \Rightarrow \left[ \text{LOC} | \text{CAT} | \text{HEAD} | \text{DSL} \quad \text{none} \right]$$



# Die Semantik gibt es umsonst!

# Die Semantik gibt es umsonst!



Nächste Woche



Nächste Woche reden wir über Fernabhängigkeiten wie Vorfeldbesetzung.

Nächste Woche reden wir über Fernabhängigkeiten wie Vorfeldbesetzung.

Sie sollten dringend vorher aus dem HPSG-Buch  
von Kapitel 10 die Seiten 163–171 lesen!

Nächste Woche reden wir über Fernabhängigkeiten wie Vorfeldbesetzung.

Sie sollten dringend vorher aus dem HPSG-Buch  
von Kapitel 10 die Seiten 163–171 lesen!

Das sind 9 Seiten.

- Müller, Stefan. 2013. *Head-Driven Phrase Structure Grammar: Eine Einführung*. 3. Aufl. (Stauffenburg Einführungen 17). Tübingen: Stauffenburg Verlag.
- Pollard, Carl & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. (Studies in Contemporary Linguistics 4).



## Kontakt

Prof. Dr. Roland Schäfer  
Institut für Germanistische Sprachwissenschaft  
Friedrich-Schiller-Universität Jena  
Fürstengraben 30  
07743 Jena

<https://rolandschaefer.net>  
[roland.schaefer@uni-jena.de](mailto:roland.schaefer@uni-jena.de)

## Creative Commons BY-SA-3.0-DE

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ *Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland* zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie

<http://creativecommons.org/licenses/by-sa/3.0/de/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.