



Formale Morphosyntax: HPSG

Einleitung

Roland Schäfer

Professur für Grammatik und Lexikon

Institut für Germanistische Sprachwissenschaft

Friedrich-Schiller-Universität Jena

roland.schaefer@uni-jena.de

Dieser Foliensatz wurde von Stefan Müller geklaut!

<https://hpsg.hu-berlin.de/~stefan/Lehre/S2021/hpsg.html>

4. November 2022



Ziele

- Vermittlung grundlegender Vorstellungen über deutsche Syntax
- Gefühl für die Daten, Zusammenhänge und Komplexität
- Einführung in Grundannahmen in der HPSG
- Befähigung zum Schreiben formaler Grammatiken



Alte Weisheit

[Grammatik ist] das Tor zur Freiheit, die Medizin für die Krankheiten der Sprache, der Reiniger aller Wissenschaften; sie verbreitet ihr Licht über ihnen; ... sie ist die erste Sprosse auf der Leiter, die zur Realisierung übernatürlicher Kräfte führt und der gerade, königliche Weg für diejenigen, die die Freiheit suchen. (Bhartrhari, Spruchdichter, gest. vor 650 n. Chr., aus *Vakyapadiya*, gefunden von Gabriele Knoll)



Gliederung

- Wozu Syntax? / Phrasenstrukturgrammatiken
- Formalismus
- Valenz und Grammatikregeln
- Komplementation
- Semantik
- Adjunktion und Spezifikation
- Das Lexikon: Typen und Lexikonregeln
- Topologie des deutschen Satzes
- Konstituentenreihenfolge
- Nichtlokale Abhängigkeiten
- Relativsätze
- Lokalität



Literaturhinweise

- Literatur: Müller (2013b: Kapitel 1) bzw. Müller (2013a: Kapitel 1)
- Englische Version des Grammatiktheoriebuches: Müller (2020: Kapitel 1)



Wozu Syntax?

- Literatur: Müller (2013b: Kapitel 1) bzw. Müller (2013a: Kapitel 1)

- Zeichen: Form-Bedeutungs-Paare Saussure (1916)

- Wörter, Wortgruppen, Sätze

- Sprache $\stackrel{?}{=}$ endliche Aufzählung von Wortfolgen
Sprache ist endlich, wenn man maximale Satzlänge annimmt

- (1) a. Dieser Satz geht weiter und weiter und weiter und weiter ...
b. [Ein Satz ist ein Satz] ist ein Satz.

extrem viele Sätze, Beschränkung der Wiederholung willkürlich

- Unterscheidung zwischen **Kompetenz** (das Wissen darüber, was geht) und **Performanz** (der Benutzung des Wissens)



Die Kinder von Bullerbü

Und wir beeilten uns, den Jungen zu erzählen, wir hätten von Anfang an gewußt, daß es nur eine Erfindung von Lasse gewesen sei. Und da sagte Lasse, die Jungen hätten gewußt, daß wir gewußt hätten, es sei nur eine Erfindung von ihm. Das war natürlich gelogen, aber vorsichtshalber sagten wir, wir hätten gewußt, die Jungen hätten gewußt, daß wir gewußt hätten, es sei nur eine Erfindung von Lasse. Und da sagten die Jungen – ja – jetzt schaffe ich es nicht mehr aufzuzählen, aber es waren so viele „gewußt“, daß man ganz verwirrt davon werden konnte, wenn man es hörte. (S. 248)

Wir sind prinzipiell in der Lage, komplexere Sätze zu bilden (Kompetenz), aber irgendwann werden wir verwirrt, weil unsere Gehirne nicht mehr mitmachen (Performanz).



Kreativität

- Wir können Sätze bilden, die wir noch nie gehört haben →
muss Strukturierung, Muster geben



Direkte Evidenz für syntaktische Strukturen?

- Wir können feststellen, dass wir Regeln verwenden, indem wir Kinder beobachten.
Kinder wenden Regeln mitunter falsch an (bzw. eben ihre eigenen Regeln).
- Beispiel aus der Morphologie:

(2) a. * die Baggers
 b. * die Ritters



Wozu Syntax? Bedeutung aus Bestandteilen ermitteln

- Bedeutung einer Äußerung aus den Bedeutungen ihrer Teile bestimmen

(3) Der Mann kennt diese Frau.

- Syntax: Art und Weise der Kombination, Strukturierung

(4) a. Die Frau kennt die Mädchen.

b. Die Frau kennen die Mädchen.

c. Die Frau schläft.

d. Die Mädchen schlafen.

Subjekt-Verb-Kongruenz → Bedeutung von (4a,b) ist eindeutig



Warum formal?

Precisely constructed models for linguistic structure can play an important role, both negative and positive, in the process of discovery itself. By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the linguistic data. More positively, a formalized theory may automatically provide solutions for many problems other than those for which it was explicitly designed. Obscure and intuition-bound notions can neither lead to absurd conclusions nor provide new and correct ones, and hence they fail to be useful in two important respects. I think that some of those linguists who have questioned the value of precise and technical development of linguistic theory have failed to recognize the productive potential in the method of rigorously stating a proposed theory and applying it strictly to linguistic material with no attempt to avoid unacceptable conclusions by ad hoc adjustments or loose formulation. (Chomsky 1957: S. 5)

As is frequently pointed out but cannot be overemphasized, an important goal of formalization in linguistics is to enable subsequent researchers to see the defects of an analysis as clearly as its merits; only then can progress be made efficiently. (Dowty 1979: S. 322)

- Was bedeutet eine Analyse genau?
- Welche Vorhersagen macht sie?
- Ausschluß anderer Analysen



Einteilung in Einheiten

- Sätze können Sätze enthalten, die Sätze enthalten, die ...:
(5) dass Max glaubt, [dass Julius weiß, [dass Otto behauptet, [dass Karl vermutet, [dass Richard bestätigt, [dass Friederike lacht]]]]]]

Das funktioniert wie eine Matrjoschka bzw. wie eine Zwiebel.

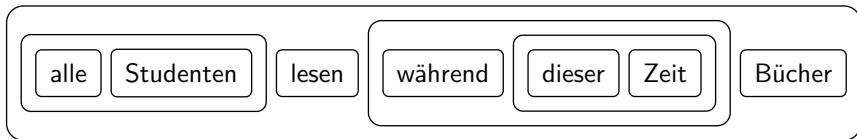
- Genauso kann man in (6) Wörter zu Einheiten zusammenfassen:

(6) Alle Studenten lesen während dieser Zeit Bücher.

Welche?



Schachteln



Wir tun alle Wörter, die zusammengehören, in eine Schachtel.

Diese Schachteln können wieder in andere Schachteln getan werden.

Im Beispiel ist intuitiv klar, was zusammengehört, aber gibt es Tests?



Konstituenz

Begriffe:

Wortfolge Eine beliebige linear zusammenhängende Folge von Wörtern, die nicht unbedingt syntaktisch oder semantisch zusammengehörig sein müssen.

Wortgruppe, Konstituente, Phrase Ein Wort oder mehrere Wörter, die eine strukturelle Einheit bilden.



Konstituententests

Welche kennen Sie?

- Substituierbarkeit/Pronominalisierungstest/Fragetest
- Weglaßtest
- Verschiebetest (Umstelltest)
- Koordinationstest



Konstituententests (I)

Substituierbarkeit Kann man eine Wortfolge einer bestimmten Kategorie in einem Satz gegen eine andere Wortfolge so austauschen, dass wieder ein akzeptabler Satz entsteht, so ist das ein Indiz dafür, dass die beiden Wortfolgen Konstituenten bilden.

- (7) a. Er kennt den Mann.
b. Er kennt eine Frau.

Pronominalisierungstest Alles, worauf man sich mit einem Pronomen beziehen kann, ist eine Konstituente.

- (8) a. Der Mann schläft.
b. Er schläft.



Konstituententests (II)

Fragetest Was sich erfragen läßt, ist eine Konstituente.

- (9) a. Der Mann arbeitet.
b. Wer arbeitet?

Verschiebetest Wortfolgen, die man ohne Beeinträchtigung der Korrektheit des Satzes verschieben bzw. umstellen kann, bilden eine Konstituente.

- (10) a. weil keiner diese Frau kennt.
b. weil diese Frau keiner kennt.

Koordinationstest Was sich koordinieren läßt, ist eine Konstituente.

- (11) Der Mann und die Frau arbeiten.



Warnung

Achtung: Diese Tests liefern leider nur Indizien für den Konstituentenstatus.
Zu den Details siehe Müller (2013a: Abschnitt 1.3.2).



Köpfe

Kopf bestimmt die wichtigsten Eigenschaften einer Phrase

- (12)
- a. **Träumt** er?
 - b. **Erwartet** er einen dreiprozentigen Anstieg?
 - c. **in** diesem Haus
 - d. ein **Mann**

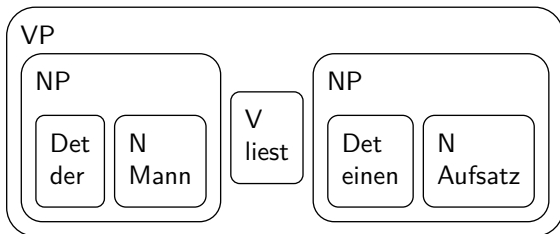
Kombination eines Kopfes mit anderem Material wird **Projektion des Kopfes** genannt.

Eine vollständige Projektion ist eine **Maximalprojektion**.

Ein Satz ist die Maximalprojektion eines finiten Verbs.



Beschriftete Schachteln



Wer schon einmal umgezogen ist, weiß, dass es sinnvoll ist, Schachteln zu beschriften.

Im obigen Bild steht auf jeder Schachtel etwas über das wichtigste Element in der Schachtel.



Schachteln sind austauschbar

- Der genaue Inhalt einer Schachtel ist egal:

- (13)
- a. er
 - b. der Mann
 - c. der Mann aus Stuttgart
 - d. der Mann aus Stuttgart, den wir kennen

Wichtig ist: Die Wörter bzw. Wortfolgen in (13) sind alle nominal und vollständig: NP.
Man kann sie innerhalb größerer Schachtel gegeneinander vertauschen.

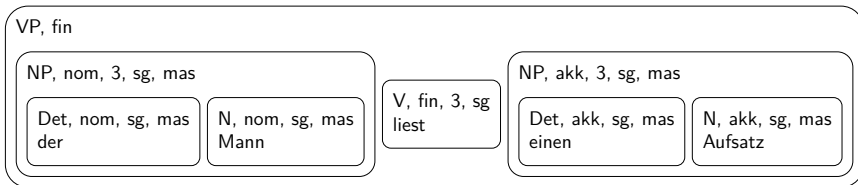
- Das geht aber nicht mit allen NPen:

- (14)
- a. Der Mann liest einen Aufsatz.
 - b. * Die Männer liest einen Aufsatz.
 - c. * Des Mannes liest einen Aufsatz.

- Es gibt Eigenschaften, die für die Verteilung (Distribution) von Phrasen wichtig sind.



Ausführlich beschriftete Schachteln



Alle Merkmale, die für die Distribution der gesamten Phrase wichtig sind, werden projiziert.

Diese Merkmale werden auch **Kopfmerkmale** genannt.



Argumente

- Konstituenten stehen in verschiedenartigen Beziehungen zu ihrem Kopf.
- Man unterscheidet zwischen **Argumenten** und **Adjunkten**.
- Bestimmte Mitspieler (Aktanten) gehören zur Bedeutung eines Verbs.
Z. B. gibt es in Situationen, die durch *lieben* beschrieben werden,
immer einen *Liebenden* und einen *Geliebten* / etwas *Geliebtes*.

- (15) a. Conny liebt Aicke.
b. *lieben'*(*Conny'*, *Aicke'*)

(15b) ist eine logische Repräsentation für (15a).

Conny' und *Aicke'* sind **logische Argumente** von *lieben'*.

- Syntaktische Argumente entsprechen meistens den logischen (später mehr).
- Solche Beziehungen zwischen Kopf und Argumenten werden mit dem Begriff **Selektion** bzw. **Valenz** erfasst.
- Tesnière 1959 überträgt Valenzbegriff aus der Chemie auf die Linguistik.



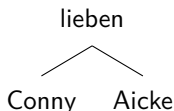
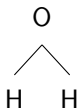
Valenz in der Chemie

- Atome können sich mit anderen Atomen zu mehr oder weniger stabilen Molekülen verbinden.
- Wichtig für die Stabilität ist, wie Elektronenschalen besetzt sind.
- Eine Verbindung mit anderen Atomen kann dazu führen, dass eine Elektronenschale voll besetzt ist, was dann zu einer stabilen Verbindung führt.
- Die Valenz sagt etwas über die Anzahl der Wasserstoffatome aus, die mit einem Atom eines Elements verbunden werden können.
- Sauerstoff hat die Valenz 2 und kann sich zu H_2O verbinden.
- Man kann nun die Elemente in Valenzklassen einteilen. Elemente mit einer bestimmten Valenz werden im Periodensystem von Mendeleev in einer Spalte repräsentiert.



Valenz in der Linguistik

- Ein Kopf braucht bestimmte Argumente, um eine stabile Verbindung einzugehen.
- Wörter mit der gleichen Valenz (mit gleicher Anzahl und Art von Argumenten) werden in Valenzklassen eingeordnet, da sie sich in bezug auf die Verbindungen, die sie eingehen, gleich verhalten.



Verbindung von Sauerstoff mit Wasserstoff und Verbindung eines Verbs mit seinen Argumenten



Optionale Argumente

- Argumente müssen nicht immer realisiert werden:

- (16)
- a. Er wartet auf den Installateur.
 - b. Er wartet.

Das Präpositionalobjekt von *warten* ist ein **fakultatives Argument**.

- In nominalen Umgebungen sind Argumente immer optional!

- (17)
- a. Jemand liest diese Bücher.
 - b. das Lesen dieser Bücher
 - c. das Lesen



Syntaktische Argumente, die keine logischen sind

- In unserem bisherigen Beispiel entsprechen die syntaktischen den logischen Argumenten:
 - (18) a. Conny liebt Aicke.
b. *lieben'*(*Conny'*, *Aicke'*)
- Allerdings gibt es auch Argumente, die keinen semantischen Beitrag leisten:
 - (19) a. Es regnet.
b. Conny erholt sich.

es und *sich* sind **syntaktische Argumente**,
aber keine **logischen Argumente**.



Argumente und Adjunkte

- Adjunkte füllen keine semantische Rolle
- Adjunkte sind optional
- Adjunkte sind iterierbar



Adjunkte füllen keine semantische Rolle

- In einer *lieben*-Situation gibt es einen Liebenden und etwas Geliebtes.
seit der Schulzeit in (20) ist von anderer Art:

(20) Conny liebt Aicke seit der Schulzeit.

Es sagt zusätzlich etwas über die Dauer der Relation aus,
in der Conny und Aicke zueinander stehen.



Adjunkte sind optional

- Adjunkte sind optional:

- (21)
- a. Conny liebt Aicke.
 - b. Conny liebt Aicke seit der Schulzeit.
 - c. Conny liebt Aicke aufrichtig.

- Vorsicht! Das ist auch bei Argumenten mitunter der Fall:

- (22)
- a. Conny gibt den Armen Geld.
 - b. Conny gibt den Armen.
 - c. Conny gibt Geld.
 - d. Conny gibt gerne.
 - e. Du gibst. (beim Skat)
 - f. Gib!



Adjunkte sind iterierbar

- Argumente können nur einmal mit dem Kopf kombiniert werden:

(23) * Das Kind das Kind lacht.

Die entsprechende Andockstelle des Kopfes (*lacht*) ist besetzt.

- Bei Adjunkten ist das anders:

(24) A: Alle grauen Eichhörnchen sind groß.

B: Nein, ich habe ein kleines graues Eichhörnchen gesehen.

A: Aber alle kleinen grauen Eichhörnchen sind krank.

B: Nein, ich habe ein gesundes kleines graues Eichhörnchen gesehen.

...



Weiter Beispiele für Adjunkte

Adverbial gebrauchtes Adjektiv (nicht alle Adjektive):

(25) Conny lacht *laut*.

Relativsätze (nicht alle):

(26) a. das Kind, *dem Aicke hilft*

b. das Kind, *das Aicke hilft*

Präpositionalphrasen (nicht alle):

(27) a. Die Frau arbeitet *in Berlin*.

b. die Frau *aus Berlin*



Andere Bezeichnungen

- Argument: Ergänzung
- Adjunkt: (freie) Angabe
- Argumente werden mitunter in Subjekt und Komplemente aufgeteilt.
- auch Aktant für Subjekte und Objekte
(aber nicht Prädikative und Adverbialien)
- Zirkumstant für Adverbialien
 - Adverbiale des Raumes (Lage, Richtung/Ziel, Herkunft, Weg)
 - Adverbiale der Zeit (Zeitpunkt, Anfang, Ende, Dauer)
 - Adverbiale des Grundes.
Hierher werden traditionellerweise auch Adverbialien gestellt,
die einen Gegengrund oder eine Bedingung ausdrücken.
 - Adverbiale der Art und Weise.



Verschiedene Grammatikmodelle (I)

- **Dependenzgrammatik (DG)**
(Tesnière 1980, 2015, Kunze 1975, Weber 1997, Heringer 1996, Eroms 2000)
- **Kategorialgrammatik (CG)**
(Ajdukiewicz 1935, Steedman 2000)
- **Phrasenstrukturgrammatik (PSG)**
- **Transformationsgrammatik und deren Nachfolger**
 - **Transformationsgrammatik**
(Chomsky 1957, Bierwisch 1963)
 - **Government & Binding**
(Chomsky 1981, von Stechow & Sternefeld 1988, Grewendorf 1988)
 - **Minimalismus**
(Chomsky 1995, Grewendorf 2002)

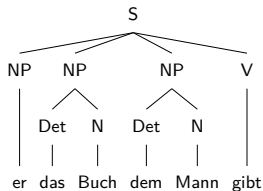


Verschiedene Grammatikmodelle (II)

- Tree Adjoning Grammar
(Joshi, Levy & Takahashi 1975, Joshi 1987, Kroch & Joshi 1985)
- Generalisierte Phrasenstrukturgrammatik (GPSG)
(Gazdar, Klein, Pullum & Sag 1985, Uszkoreit 1987)
- Lexikalisch Funktionale Grammatik (LFG)
(Bresnan 1982, 2001, Berman & Frank 1996, Berman 2003)
- Head-Driven Phrase Structure Grammar (HPSG)
(Pollard & Sag 1987, 1994, Müller 1999, 2002, 2013b, Müller et al. 2021)
- Construction Grammar (CxG)
(Fillmore, Kay & O'Connor 1988, Goldberg 1995, 2006, Fischer & Stefanowitsch 2006)
- Zu einem Überblick siehe Müller (2013a) bzw. Müller (2020).

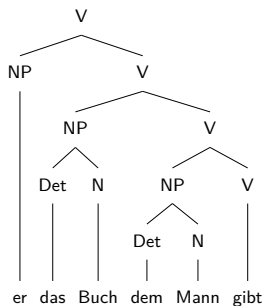


Phrasenstrukturen



$NP \rightarrow Det, N$

$S \rightarrow NP, NP, NP, V$



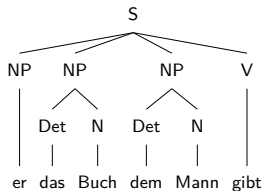
$NP \rightarrow Det, N$

$V \rightarrow NP, V$

Das Eigentliche sind die Ersetzungsregeln! Die Bäume sind nur die Visualisierung.

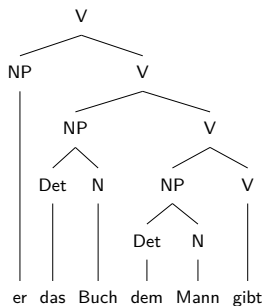


Phrasenstrukturen



NP → Det, N

S → NP, NP, NP, V



NP → Det, N

V → NP, V

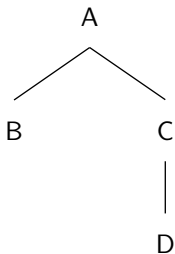
Das Eigentliche sind die Ersetzungsregeln! Die Bäume sind nur die Visualisierung.

Aus Platzgründen auch Kammerschreibweise:

[S [NP er] [NP [Det das] [N Buch]] [NP [Det dem] [N Mann]] [V gibt]]

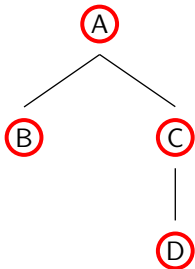


Knoten (*node*)



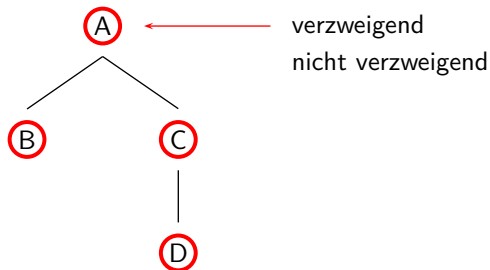


Knoten (*node*)



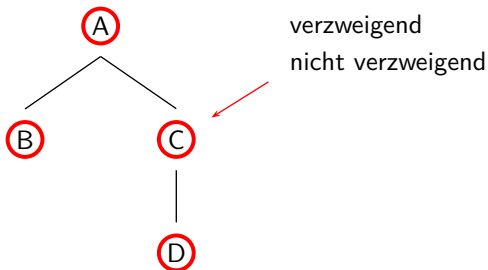


Knoten (*node*)



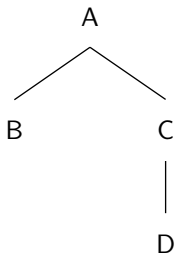


Knoten (*node*)





Mutter, Tochter und Schwester



A ist die Mutter von B und C

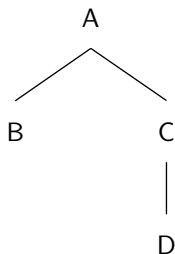
C ist die Mutter von D

B ist die Schwester von C

Verhältnisse wie in Stammbäumen



Dominanz (*dominance*)

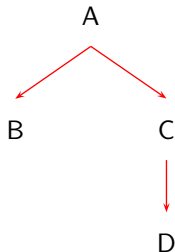


A dominiert

A dominiert B genau dann, wenn A höher im Baum steht und wenn es eine ausschließlich abwärts führende Linie von A nach B gibt.



Dominanz (*dominance*)

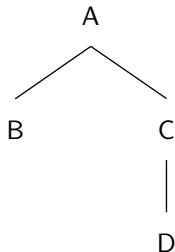


A dominiert B, C und D

A dominiert B genau dann, wenn A höher im Baum steht und wenn es eine ausschließlich abwärts führende Linie von A nach B gibt.



Dominanz (*dominance*)

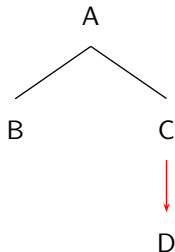


A dominiert B, C und D
C dominiert

A dominiert B genau dann, wenn A höher im Baum steht und wenn es eine ausschließlich abwärts führende Linie von A nach B gibt.



Dominanz (*dominance*)



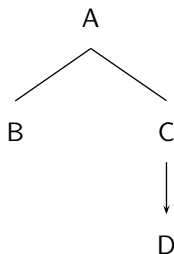
A dominiert B, C und D

C dominiert D

A dominiert B genau dann, wenn A höher im Baum steht und wenn es eine ausschließlich abwärts führende Linie von A nach B gibt.



Unmittelbare Dominanz (*immediate dominance*)

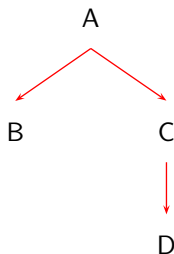


A dominiert unmittelbar

A dominiert unmittelbar B genau dann, wenn
A B dominiert und es keinen Knoten C zwischen A und B gibt.



Unmittelbare Dominanz (*immediate dominance*)

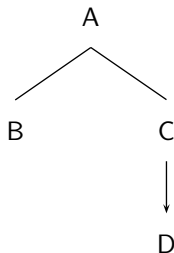


A dominiert unmittelbar B und C

A dominiert unmittelbar B genau dann, wenn
A B dominiert und es keinen Knoten C zwischen A und B gibt.



Unmittelbare Dominanz (*immediate dominance*)

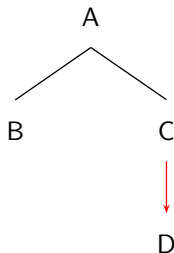


A dominiert unmittelbar B und C
C dominiert unmittelbar D

A dominiert unmittelbar B genau dann, wenn
A B dominiert und es keinen Knoten C zwischen A und B gibt.



Unmittelbare Dominanz (*immediate dominance*)



A dominiert unmittelbar B und C
C dominiert unmittelbar D

A dominiert unmittelbar B genau dann, wenn
A B dominiert und es keinen Knoten C zwischen A und B gibt.



Präzedenz

Präzedenz (*precedence*)

A geht B voran, wenn A in einer Baumgrafik vor B steht und keiner der beiden Knoten den anderen dominiert.



Präzedenz

Präzedenz (*precedence*)

A geht B voran, wenn A in einer Baumgrafik vor B steht und keiner der beiden Knoten den anderen dominiert.

Unmittelbare Präzedenz (*immediate precedence*)

Kein Element C zwischen A und B.



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

er das Buch dem Mann gibt

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

er das Buch dem Mann gibt

NP das Buch dem Mann gibt

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

er das Buch dem Mann gibt

NP das Buch dem Mann gibt

NP Det Buch dem Mann gibt

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er das Buch dem Mann gibt

NP das Buch dem Mann gibt

NP Det Buch dem Mann gibt

NP Det N dem Mann gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er	das	Buch	dem	Mann	gibt
NP	das	Buch	dem	Mann	gibt
NP	Det	Buch	dem	Mann	gibt
NP	Det	N	dem	Mann	gibt
NP		NP	dem	Mann	gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er das Buch dem Mann gibt

NP das Buch dem Mann gibt

NP Det Buch dem Mann gibt

NP Det N dem Mann gibt

NP NP dem Mann gibt

NP NP Det Mann gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er	das	Buch	dem	Mann	gibt
NP	das	Buch	dem	Mann	gibt
NP	Det	Buch	dem	Mann	gibt
NP	Det	N	dem	Mann	gibt
NP		NP	dem	Mann	gibt
NP		NP	Det	Mann	gibt
NP		NP	Det	N	gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er	das	Buch	dem	Mann	gibt
NP	das	Buch	dem	Mann	gibt
NP	Det	Buch	dem	Mann	gibt
NP	Det	N	dem	Mann	gibt
NP		NP	dem	Mann	gibt
NP		NP	Det	Mann	gibt
NP		NP	Det	N	gibt
NP		NP		NP	gibt



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er	das	Buch	dem	Mann	gibt
NP	das	Buch	dem	Mann	gibt
NP	Det	Buch	dem	Mann	gibt
NP	Det	N	dem	Mann	gibt
NP		NP	dem	Mann	gibt
NP		NP	Det	Mann	gibt
NP		NP	Det	N	gibt
NP		NP		NP	gibt
NP		NP		NP	V



Beispielableitung bei Annahme flacher Strukturen

NP → Det N

S → NP NP NP V

NP → er

Det → das

Det → dem

N → Buch

N → Mann

V → gibt

er	das	Buch	dem	Mann	gibt
NP	das	Buch	dem	Mann	gibt
NP	Det	Buch	dem	Mann	gibt
NP	Det	N	dem	Mann	gibt
NP		NP	dem	Mann	gibt
NP		NP	Det	Mann	gibt
NP		NP	Det	N	gibt
NP		NP		NP	gibt
NP		NP		NP	V

S



Do try this at home!

Sie können solche Grammatiken selbst ausprobieren.

- Gehen Sie auf <https://swish.swi-prolog.org/>.
- Klicken Sie „Program“.
- Geben Sie folgendes ein:
s --> np, v, np, np.
np --> det, n.
np --> [er].
det --> [das].
det --> [dem].
n --> [buch].
n --> [kind].
v --> [gibt].
- Geben Sie in die untere rechte Box folgendes ein:
s([er,gibt,das,buch,dem,kind], []).
- Wenn in der Box darüber „true“ erscheint, feiern Sie!



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`
- `s([X],[])` fordert Prolog auf, ein `X` zu finden, das ein „s“ ist.



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`
- `s([X],[])` fordert Prolog auf, ein `X` zu finden, das ein „s“ ist.
- `print(X),nl` gibt das `X` und eine newline aus und



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`
- `s([X],[])` fordert Prolog auf, ein `X` zu finden, das ein „s“ ist.
- `print(X),nl` gibt das `X` und eine newline aus und
- `fail` teilt Prolog mit, dass wir nicht zufrieden sind und dass es noch eine weitere Lösung suchen soll.



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`
- `s([X],[])` fordert Prolog auf, ein `X` zu finden, das ein „s“ ist.
- `print(X),nl` gibt das `X` und eine newline aus und
- `fail` teilt Prolog mit, dass wir nicht zufrieden sind und dass es noch eine weitere Lösung suchen soll.
- Es versucht weiter, bis es keine weiteren Lösungen mehr gibt und `failt` dann.



Eine Generative Grammatik

- Die Grammatik, die Sie eingegeben haben, kann Sätze erzeugen:
- Man kann testen, welche Sätze die Grammatik generiert, indem man folgendes eingibt: `s([X],[]),print(X),nl,fail.`
- `s([X],[])` fordert Prolog auf, ein `X` zu finden, das ein „s“ ist.
- `print(X),nl` gibt das `X` und eine newline aus und
- `fail` teilt Prolog mit, dass wir nicht zufrieden sind und dass es noch eine weitere Lösung suchen soll.
- Es versucht weiter, bis es keine weiteren Lösungen mehr gibt und `fail` dann.
- Einige Grammatiken generieren unendlich viele `X`. Dieser Prozess würde also nie terminieren (es sei denn, der Computer hat nicht genug Speicher ...).



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

- (28)
- a. er das Buch dem Mann gibt
 - b. * ich das Buch dem Mann gibt



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

(28) a. er das Buch dem Mann gibt

b. * ich das Buch dem Mann gibt
(Subjekt-Verb-Kongruenz *ich, gibt*)



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

(28) a. er das Buch dem Mann gibt

b. * ich das Buch dem Mann gibt
(Subjekt-Verb-Kongruenz *ich*, *gibt*)

c. * er das Buch den Mann gibt



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

- (28)
- a. er das Buch dem Mann gibt
 - b. * ich das Buch dem Mann gibt
(Subjekt-Verb-Kongruenz *ich, gibt*)
 - c. * er das Buch den Mann gibt
(Kasusanforderungen des Verbs *gibt* verlangt Dativ)



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

- (28)
- a. er das Buch dem Mann gibt
 - b. * ich das Buch dem Mann gibt
(Subjekt-Verb-Kongruenz *ich, gibt*)
 - c. * er das Buch den Mann gibt
(Kasusanforderungen des Verbs *gibt* verlangt Dativ)
 - d. * er den Buch dem Mann gibt



Von der Grammatik beschriebene Sätze

- die Grammatik ist zu ungenau:

NP → Det N

S → NP NP NP V

- (28)
- a. er das Buch dem Mann gibt
 - b. * ich das Buch dem Mann gibt
(Subjekt-Verb-Kongruenz *ich, gibt*)
 - c. * er das Buch den Mann gibt
(Kasusanforderungen des Verbs *gibt* verlangt Dativ)
 - d. * er den Buch dem Mann gibt
(Determinator-Nomen-Kongruenz *den, Buch*)



Subjekt-Verb-Kongruenz (I)

- Übereinstimmung in Person (1, 2, 3) und Numerus (sg, pl)

- (29)
- a. Ich schlafe. (1, sg)
 - b. Du schläfst. (2, sg)
 - c. Er schläft. (3, sg)
 - d. Wir schlafen. (1, pl)
 - e. Ihr schlaft. (2, pl)
 - f. Sie schlafen. (3,pl)

- Wie drückt man das in Regeln aus?



Subjekt-Verb-Kongruenz (II)

- Verfeinerung der verwendeten Symbole
aus $S \rightarrow NP\ NP\ NP\ V$ wird

$S \rightarrow NP_1_sg\ NP\ NP\ V_1_sg$

$S \rightarrow NP_2_sg\ NP\ NP\ V_2_sg$

$S \rightarrow NP_3_sg\ NP\ NP\ V_3_sg$

$S \rightarrow NP_1_pl\ NP\ NP\ V_1_pl$

$S \rightarrow NP_2_pl\ NP\ NP\ V_2_pl$

$S \rightarrow NP_3_pl\ NP\ NP\ V_3_pl$

- sechs Symbole für Nominalphrasen, sechs für Verben
- sechs Regeln statt einer



Kasuszuweisung durch das Verb

- Kasus muß repräsentiert sein:
S \rightarrow NP_1_sg_nom NP_dat NP_acc V_1_sg_ditransitiv
S \rightarrow NP_2_sg_nom NP_dat NP_acc V_2_sg_ditransitiv
S \rightarrow NP_3_sg_nom NP_dat NP_acc V_3_sg_ditransitiv
S \rightarrow NP_1_pl_nom NP_dat NP_acc V_1_pl_ditransitiv
S \rightarrow NP_2_pl_nom NP_dat NP_acc V_2_pl_ditransitiv
S \rightarrow NP_3_pl_nom NP_dat NP_acc V_3_pl_ditransitiv
- insgesamt $3 * 2 * 4 = 24$ neue Kategorien für NP
- $3 * 2 * x$ Kategorien für V (x = Anzahl der Valenzmuster)



Determinator-Nomen-Kongruenz

- Übereinstimmung in Genus (fem, mas, neu), Numerus (sg, pl) und Kasus (nom, gen, dat, acc)

- (30)
- a. der Mann, die Frau, das Buch (Genus)
 - b. das Buch, die Bücher (Numerus)
 - c. des Buches, dem Buch (Kasus)



Determinator-Nomen-Kongruenz

- Übereinstimmung in Genus (fem, mas, neu), Numerus (sg, pl) und Kasus (nom, gen, dat, acc)

- (30)
- a. der Mann, die Frau, das Buch (Genus)
 - b. das Buch, die Bücher (Numerus)
 - c. des Buches, dem Buch (Kasus)

- aus NP → Det N wird

NP_3_sg_nom → Det_fem_sg_nom N_fem_sg_nom
 NP_3_sg_nom → Det_mas_sg_nom N_mas_sg_nom
 NP_3_sg_nom → Det_neu_sg_nom N_neu_sg_nom
 NP_3_pl_nom → Det_fem_pl_nom N_fem_pl_nom
 NP_3_pl_nom → Det_mas_pl_nom N_mas_pl_nom
 NP_3_pl_nom → Det_neu_pl_nom N_neu_pl_nom

...

Dativ

NP_gen → Det_fem_sg_gen N_fem_sg_gen
 NP_gen → Det_mas_sg_gen N_mas_sg_gen
 NP_gen → Det_neu_sg_gen N_neu_sg_gen
 NP_gen → Det_fem_pl_gen N_fem_pl_gen
 NP_gen → Det_mas_pl_gen N_mas_pl_gen
 NP_gen → Det_neu_pl_gen N_neu_pl_gen

...

Akkusativ

- 24 Symbole für Determinatoren, 24 Symbole für Nomen
- 24 Regeln statt einer



Probleme dieses Ansatzes

- Generalisierungen werden nicht erfaßt.
- weder in Regeln noch in Categoriesymbolen
 - Wo kann eine NP oder NP_nom stehen?
Nicht wo kann eine NP_3_sg_nom stehen?
 - Gemeinsamkeiten der Regeln sind nicht offensichtlich.



Probleme dieses Ansatzes

- Gernalisierungen werden nicht erfaßt.
- weder in Regeln noch in Categoriesymbolen
 - Wo kann eine NP oder NP_nom stehen?
Nicht wo kann eine NP_3_sg_nom stehen?
 - Gemeinsamkeiten der Regeln sind nicht offensichtlich.
- Lösung: Merkmale mit Werten und Identität von Werten
Categoriesymbol: NP Merkmal: Per, Num, Kas, ...
Wir erhalten z. B. die Regeln:
NP(3,sg,nom) → Det(fem,sg,nom) N(fem,sg,nom)
NP(3,sg,nom) → Det(mas,sg,nom) N(mas,sg,nom)



Merkmale und Regelschemata (I)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:
$$\text{NP}(3, \text{Num}, \text{Kas}) \rightarrow \text{Det}(\text{Gen}, \text{Num}, \text{Kas}) \text{ N}(\text{Gen}, \text{Num}, \text{Kas})$$



Merkmale und Regelschemata (I)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:
$$\text{NP}(3, \text{Num}, \text{Kas}) \rightarrow \text{Det}(\text{Gen}, \text{Num}, \text{Kas}) \text{ N}(\text{Gen}, \text{Num}, \text{Kas})$$
- Gen-, Num- und Kas-Werte sind egal,
Hauptsache sie stimmen überein (identische Werte)



Merkmale und Regelschemata (I)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:
 $NP(3, \text{Num}, \text{Kas}) \rightarrow \text{Det}(\text{Gen}, \text{Num}, \text{Kas}) N(\text{Gen}, \text{Num}, \text{Kas})$
- Gen-, Num- und Kas-Werte sind egal,
Hauptsache sie stimmen überein (identische Werte)
- Der Wert des Personenmerkmals (erste Stelle in $NP(3, \text{Num}, \text{Kas})$)
ist durch die Regel festgelegt: 3.



Merkmale und Regelschemata (II)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:

NP(3,Num,Kas) → Det(Gen,Num,Kas) N(Gen,Num,Kas)

S → NP(Per1,Num1,nom)

NP(Per2,Num2,dat)

NP(Per3,Num3,akk)

V(Per1,Num1)

- Per1 und Num1 sind beim Verb und Subjekt gleich.



Merkmale und Regelschemata (II)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:

NP(3,Num,Kas) → Det(Gen,Num,Kas) N(Gen,Num,Kas)

S → NP(Per1,Num1,nom)

NP(Per2,Num2,dat)

NP(Per3,Num3,akk)

V(Per1,Num1)

- Per1 und Num1 sind beim Verb und Subjekt gleich.
- Bei anderen NPen sind die Werte egal.
(Schreibweise für irrelevante Werte: ' _ ')



Merkmale und Regelschemata (II)

- Regeln mit speziellen Werten zu Regelschemata verallgemeinern:

NP(3,Num,Kas) → Det(Gen,Num,Kas) N(Gen,Num,Kas)

S → NP(Per1,Num1,nom)

NP(Per2,Num2,dat)

NP(Per3,Num3,akk)

V(Per1,Num1)

- Per1 und Num1 sind beim Verb und Subjekt gleich.
- Bei anderen NPen sind die Werte egal.
(Schreibweise für irrelevante Werte: ' _ ')
- Die Kasus der NPen sind in der zweiten Regel festgelegt.



Abstraktion über Regeln

\bar{X} -Theorie (Jackendoff 1977):

\bar{X} -Regel

$\bar{X} \rightarrow \overline{\text{Spezifikator}} \bar{X}$

$\bar{X} \rightarrow \bar{X} \overline{\text{Adjunkt}}$

$\bar{X} \rightarrow \overline{\text{Adjunkt}} \bar{X}$

$\bar{X} \rightarrow \bar{X} \overline{\text{Komplement}^*}$

mit Kategorien

$\bar{N} \rightarrow \overline{\text{DET}} \bar{N}$

$\bar{N} \rightarrow \bar{N} \overline{\text{REL_SATZ}}$

$\bar{N} \rightarrow \overline{\text{ADJ}} \bar{N}$

$\bar{N} \rightarrow \bar{N} \bar{P}$

Beispiel

das [Bild von Maria]

[Bild von Maria] [das alle kennen]

schöne [Bild von Maria]

Bild [von Maria]

X steht für beliebige Kategorie, '*' für beliebig viele Wiederholungen



\bar{X} -Theorie

\bar{X} -Theorie wird in vielen verschiedenen Frameworks angenommen:

- Government & Binding (GB): Chomsky (1981)
- Lexical Functional Grammar (LFG): Bresnan (1982, 2001)
- Generalized Phrase Structure Grammar (GPSG):
Gazdar, Klein, Pullum & Sag (1985)



Hausaufgabe

1. Schreiben Sie eine Phrasenstrukturgrammatik, mit der man u. a. die Sätze in (31) analysieren kann, die die Wortfolgen in (32) aber nicht zulässt.

- (31) a. Der Mann hilft der Frau.
b. Er gibt ihr das Buch.
c. Er wartet auf ein Wunder.

- (32) a. * Der Mann hilft er.
b. * Er gibt ihr den Buch.

Dabei sollen Sie nicht für jeden Satz einzeln eigene Regeln für NP usw. aufstellen, sondern gemeinsame Regeln für alle aufgeführten Sätze entwickeln. Sie können für Ihre Arbeit auch Prolog benutzen: <https://swish.swi-prolog.org> zur Syntax für die Grammatiken siehe https://en.wikipedia.org/wiki/Definite_clause_grammar.



Literaturverzeichnis

- Ajdukiewicz, Kazimierz. 1935. Die syntaktische Konnexität. *Studia Philosophica* 1. 1–27.
- Berman, Judith. 2003. *Clausal syntax of German*. (Studies in Constraint-Based Lexicalism). Stanford, CA: CSLI Publications.
- Berman, Judith & Anette Frank (eds.). 1996. *Deutsche und französische Syntax im Formalismus der LFG*. (Linguistische Arbeiten 344). Tübingen: Max Niemeyer Verlag. DOI: 10.1515/9783110955354.
- Bierwisch, Manfred. 1963. *Grammatik des deutschen Verbs*. (studia grammatica 2). Berlin: Akademie Verlag.
- Bloomfield, Leonard. 1933. *Language*. London: George Allen & Unwin.
- Bresnan, Joan (ed.). 1982. *The mental representation of grammatical relations*. (MIT Press Series on Cognitive Theory and Mental Representation). Cambridge, MA: MIT Press.
- Bresnan, Joan. 2001. *Lexical-Functional Syntax*. 1st edn. Oxford: Blackwell Publishers Ltd.
- Chomsky, Noam. 1957. *Syntactic structures*. (Janua Linguarum / Series Minor 4). The Hague: Mouton.
- Chomsky, Noam. 1981. *Lectures on government and binding*. (Studies in Generative Grammar 9). Dordrecht: Foris Publications. DOI: 10.1515/9783110884166.
- Chomsky, Noam. 1995. *The Minimalist Program*. (Current Studies in Linguistics 28). Cambridge, MA: MIT Press.
- Dowty, David R. 1979. *Word meaning and Montague Grammar*. (Synthese Language Library 7). Dordrecht: D. Reidel Publishing Company.
- Eroms, Hans-Werner. 2000. *Syntax der deutschen Sprache*. (de Gruyter Studienbuch). Berlin: Walter de Gruyter Verlag.
- Fillmore, Charles J., Paul Kay & Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of *let alone*. *Language* 64(3). 501–538.
- Fischer, Kerstin & Anatol Stefanowitsch (eds.). 2006. *Konstruktionsgrammatik: Von der Anwendung zur Theorie*. (Stauffenburg Linguistik 40). Tübingen: Stauffenburg Verlag.
- Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.
- Goldberg, Adele E. 1995. *Constructions: A Construction Grammar approach to argument structure*. (Cognitive Theory of Language and Culture). Chicago: The University of Chicago Press.
- Goldberg, Adele E. 2006. *Constructions at work: The nature of generalization in language*. (Oxford Linguistics). Oxford: Oxford University Press.
- Grewendorf, Günther. 1988. *Aspekte der deutschen Syntax: Eine Rektions-Bindungs-Analyse*. (Studien zur deutschen Grammatik 33). Tübingen: original Gunter Narr Verlag jetzt Stauffenburg Verlag.
- Grewendorf, Günther. 2002. *Minimalistische Syntax*. (UTB für Wissenschaft: Uni-Taschenbücher 2313). Tübingen, Basel: A. Francke Verlag GmbH.
- Heringer, Hans Jürgen. 1996. *Deutsche Syntax dependentiell*. (Stauffenburg Linguistik). Tübingen: Stauffenburg Verlag.
- Jackendoff, Ray S. 1977. *\bar{X} syntax: A study of phrase structure*. Cambridge, MA: MIT Press.
- Joshi, Aravind K. 1987. Introduction to Tree Adjoining Grammar. In Alexis Manaster-Ramer (ed.), *The mathematics of language*, 87–114. Amsterdam: John Benjamins Publishing Co.
- Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree Adjunct Grammar. *Journal of Computer and System Science* 10(2). 136–163.
- Kroch, Anthony S. & Aravind K. Joshi. 1985. *The linguistic relevance of Tree Adjoining Grammar*. Tech. rep. MS-CIS-85-16. University of Pennsylvania. http://repository.upenn.edu/cis_reports/671/ (18 August, 2020).



- Kunze, Jürgen. 1975. *Abhängigkeitsgrammatik*. (studia grammatica 12). Berlin: Akademie Verlag.
- Müller, Stefan. 1999. *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche*. (Linguistische Arbeiten 394). Tübingen: Max Niemeyer Verlag. DOI: 10.1515/9783110915990.
- Müller, Stefan. 2002. *Complex predicates: Verbal complexes, resultative constructions, and particle verbs in German*. (Studies in Constraint-Based Lexicalism 13). Stanford, CA: CSLI Publications.
- Müller, Stefan. 2013a. *Grammatiktheorie*. 2nd edn. (Stauffenburg Einführungen 20). Tübingen: Stauffenburg Verlag. <https://hpsg.hu-berlin.de/~stefan/Pub/grammatiktheorie.html> (30 June, 2019).
- Müller, Stefan. 2013b. *Head-Driven Phrase Structure Grammar: Eine Einführung*. 3rd edn. (Stauffenburg Einführungen 17). Tübingen: Stauffenburg Verlag. <https://hpsg.hu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html> (30 June, 2019).
- Müller, Stefan. 2020. *Grammatical theory: From Transformational Grammar to constraint-based approaches*. 4th edn. (Textbooks in Language Sciences 1). Berlin: Language Science Press. DOI: 10.5281/zenodo.3992307.
- Müller, Stefan, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.). 2021. *Head-Driven Phrase Structure Grammar: The handbook*. (Empirically Oriented Theoretical Morphology and Syntax). erscheint. Berlin: Language Science Press.
- Pollard, Carl & Ivan A. Sag. 1987. *Information-based syntax and semantics*. (CSLI Lecture Notes 13). Stanford, CA: CSLI Publications.
- Pollard, Carl & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. (Studies in Contemporary Linguistics). Chicago: The University of Chicago Press.
- Steedman, Mark. 2000. *The syntactic process*. (Language, Speech, and Communication). Cambridge, MA: MIT Press.
- Tesnière, Lucien. 1980. *Grundzüge der strukturalen Syntax*. Translated by Ulrich Engel. Stuttgart: Klett-Cotta.
- Tesnière, Lucien. 2015. *Elements of structural syntax*. Translated by Timothy Osborne and Sylvain Kahane. Amsterdam: John Benjamins Publishing Co.
- Uszkoreit, Hans. 1987. *Word order and constituent structure in German*. (CSLI Lecture Notes 8). Stanford, CA: CSLI Publications.
- von Stechow, Arnim & Wolfgang Sternefeld. 1988. *Bausteine syntaktischen Wissens: Ein Lehrbuch der Generativen Grammatik*. Opladen/Wiesbaden: Westdeutscher Verlag.
- Weber, Heinz J. 1997. *Dependenzgrammatik: Ein interaktives Arbeitsbuch*. 2nd edn. (Narr Studienbücher). Tübingen: Gunter Narr Verlag.