

8. Sprachhierarchien und Komplexität

8.1 Formalismus der PS-grammar

8.1.1 Ursprüngliche Definition

Von dem amerikanischen Logiker E. Post 1936 als *rewrite* oder *Post production system* publiziert, ist dieser Formalismus in der Rekursionstheorie entstanden und steht in enger Beziehung zur Automatentheorie.

8.1.2 Erste Anwendung auf natürliche Sprache

Posts *rewrite systems* wurden von N. Chomsky 1957 erstmals auf die natürlichen Sprachen angewendet, und zwar als sogenannte *phrase structure grammars*.

8.1.3 Algebraische Definition der PS-Grammatik

Eine PS-Grammatik ist ein Quadrupel $\langle V, V_T, S, P \rangle$.

1. V ist eine Menge von Zeichen.
2. V_T ist eine echte Untermenge von V , genannt *terminale Zeichen*.
3. S ist ein Zeichen in V ohne V_T , genannt *Startsymbol*.
4. P ist eine Menge von Ersetzungsregeln der Form $\alpha \rightarrow \beta$, wobei α ein Element von V^+ und β ein Element von V^* ist.

8.1.4 Restriktionstypen des PS-Regelschemas

0. Unbeschränkte PS-Regel:

Bei einer Typ-0 Regel stehen auf der linken und rechten Regelseite beliebige Folgen von Terminalen und Variablen.

1. Kontextsensitive PS-Regel:

Bei einer Typ-1 Regel stehen auf der linken und rechten Regelseite beliebige Folgen von Terminalen und Variablen, wobei die rechte Regelseite mindestens so lang sein muß wie die linke.

Beispiel: $A B C \rightarrow A D E C$

2. Kontextfreie PS-Regel:

Bei einer Typ-2 Regel steht auf der linken Regelseite genau eine Variable. Auf der rechten Regelseite steht eine Zeichenkette aus V^+ .

Beispiele: $A \rightarrow BC$, $A \rightarrow bBCc$, etc.

3. Reguläre PS-Regel:

Bei einer Typ-3 Regel steht auf der linken Regelseite genau eine Variable. Auf der rechten Regelseite steht genau ein Terminal, gefolgt von höchstens einer Variablen.

Beispiele: $A \rightarrow b$, $A \rightarrow bC$.

8.2 Sprachklassen und ihre Komplexität

8.2.1 Verschiedene Beschränkungen der generative Regelschemata führen zu

1. unterschiedlichen *Arten von Grammatiken*, die über
2. unterschiedliche *Grade generativer Kapazität*
3. unterschiedliche *Sprachklassen* erzeugen, die wiederum
4. unterschiedliche *Komplexitätsgrade* aufweisen.

8.2.2 Grade der Komplexität

1. *Lineare Komplexität*
 $n, 2n, 3n$ etc.
2. *Polynomiale Komplexität*
 n^2, n^3, n^4 etc.
3. *Exponentielle Komplexität*
 $2^n, 3^n, 4^n$ etc.
4. *Unentscheidbar*
 $n \cdot \infty$

8.2.3 Komplexität polynomialer und exponentieller Algorithmen

	Problemgröße n		
Zeit-komplexität	10	50	100
n^3	.001 Sekunden	.125 Sekunden	1.0 Sekunden
2^n	.001 Sekunden	35.7 Jahre	10^{15} Jahrhunderte

8.2.4 Anwendung auf natürliche Sprache

Das Limaskorpus enthält insgesamt 71 148 Sätze. Von diesen bestehen genau 50 aus 100 Wortformen oder mehr, wobei der längste Satz im Korpus aus 165 Wörtern besteht.

8.2.5 PS-grammatische Hierarchie der formalen Sprachen (Chomsky-Hierarchie)

Regel Beschränkung	Unterklassen der PS-Grammatik	Sprachklassen	Komplexitätsgrad
Typ-3	reguläre PSG	reguläre Spr.	linear
Typ-2	kontextfreie PSG	kontextfreie Spr.	polynomial
Typ-1	kontextsensitive PSG	kontextsensitive Spr.	exponentiell
Typ-0	unbeschränkte PSG	rek. enumerable Spr.	unentscheidbar

8.3 Generative Kapazität und formale Sprachklassen

8.3.1 Linguistische Hauptfrage an die PS-grammatik

Gibt es einen Typ der PS-Grammatik der genau die Strukturen erzeugt, die für die natürlichen Sprachen charakteristisch sind?

8.3.2 Struktureigenschaften der regulären PS-Grammatik

Die generative Kapazität der regulären PS-Grammatik erlaubt die rekursive Wiederholung einzelner Wörter, aber ohne irgendwelche rekursive Korrespondenzen.

8.3.3 Reguläre PS-Grammatik für ab^k ($k \geq 1$)

$$V =_{def} \{S, B, a, b\}$$

$$V_T =_{def} \{a, b\}$$

$$P =_{def} \{S \rightarrow a B, \\ B \rightarrow b B, \\ B \rightarrow b\}$$

8.3.4 Reguläre PS-Grammatik für $\{a, b\}^+$

$$V =_{def} \{S, a, b\}$$

$$V_T =_{def} \{a, b\}$$

$$P =_{def} \{S \rightarrow a S, \\ S \rightarrow b S, \\ S \rightarrow a, \\ S \rightarrow b\}$$

8.3.5 Reguläre PS-Grammatik für $a^m b^k$ ($k, m \geq 1$)

Regular PS-grammar for $a^m b^k$ ($k, m \geq 1$)

$$V =_{def} \{S, S_1, S_2, a, b\}$$

$$V_T =_{def} \{a, b\}$$

$$P =_{def} \{S \rightarrow a S_1, \\ S_1 \rightarrow a S_1, \\ S_1 \rightarrow b S_2, \\ S_2 \rightarrow b\}$$

8.3.6 Struktureigenschaften der kontextfreien PS-Grammatik

Die generative Kapazität der kontextfreien PS-Grammatik erlaubt die rekursive Erzeugung von invers-paarweisen Korrespondenzen, z. B. $a b c \dots c b a$.

8.3.7 Kontextfreie PS-Grammatik für $a^k b^{3k}$

$$V =_{def} \{S, a, b\}$$

$$V_T =_{def} \{a, b\}$$

$$P =_{def} \{ S \rightarrow a S b b b, \\ S \rightarrow a b b b \}$$

8.3.8 Kontextfreie PS-Grammatik für WW^R

$$V =_{def} \{S, a, b, c, d\}, V_T =_{def} \{a, b, c, d\}, P =_{def} \{ S \rightarrow a S a, \\ S \rightarrow b S b, \\ S \rightarrow c S c, \\ S \rightarrow d S d, \\ S \rightarrow a a, \\ S \rightarrow b b, \\ S \rightarrow c c, \\ S \rightarrow d d \}$$

8.3.9 Warum WW die generative Kapazität der kontextfreien PS-Grammatik übersteigt

aa
abab
abcabc
abcdabcd
...

haben keine *inverse* Struktur. Deshalb ist es trotz der Ähnlichkeit zwischen WW^R und WW unmöglich, eine kontextfreie PS-Grammatik wie 8.3.8 für WW zu schreiben.

8.3.10 Warum $a^k b^k c^k$ die generative Kapazität der kontextfreien PS-Grammatik übersteigt

a b c
a a b b c c
a a a b b b c c c
...

kann nicht von einer kontextfreien PS-Grammatik generiert werden, weil Korrespondenzen zwischen *drei* verschiedenen Bereichen aufrecht erhalten werden müssen – was die *paarweis* inverse Struktur der kontextfreien Sprachen, wie sie z. B. von den Sprachen $a^k b^k$ und WW^R illustriert wird, übersteigt.

8.3.11 Struktureigenschaften der kontextsensitiven PS-Grammatik

Almost any language one can think of is context-sensitive; the only known proofs that certain languages are not CSL's are ultimately based on diagonalization.

[Fast jede erdenkliche Sprache ist kontextsensitiv; die einzigen bekannten Beweise, daß bestimmte Sprachen nicht kontextsensitiv sind, beruhen letztlich auf Diagonalisierung.]

J.E. Hopcroft and J.D. Ullman 1979, p. 224

8.3.12 PS-Grammatik für kontextsensitives $a^k b^k c^k$

$$V =_{def} \{S, B, C, D_1, D_2, a, b, c\}$$

$$V_T =_{def} \{a, b, c\}$$

$$P =_{def} \left\{ \begin{array}{ll} S \rightarrow a S B C, & \text{rule 1} \\ S \rightarrow a b C, & \text{rule 2} \\ C B \rightarrow D_1 B, & \text{rule 3a} \\ D_1 B \rightarrow D_1 D_2, & \text{rule 3b} \\ D_1 D_2 \rightarrow B D_2, & \text{rule 3c} \\ B D_2 \rightarrow B C, & \text{rule 3d} \\ b B \rightarrow b b, & \text{rule 4} \\ b C \rightarrow b c, & \text{rule 5} \\ c C \rightarrow c c & \text{rule 6} \end{array} \right.$$

Die Regeln 3a bis 3d haben zusammen denselben Effekt wie die

Regel 3 $C B \rightarrow B C$.

8.3.13 Ableitung von $a a a b b b c c c$

	Zwischenketten	Regeln
1.	S	
2.	a S B C	(1)
3.	a a S B C B C	(1)
4.	a a a b C B C B C	(2)
5.	a a a b B C C B C	(3)
6.	a a a b B C B C C	(3)
7.	a a a b B B C C C	(3)
8.	a a a b b B C C C	(4)
9.	a a a b b b C C C	(4)
10.	a a a b b b c C C	(5)
11.	a a a b b b c c C	(6)
12.	a a a b b b c c c	(6)

8.3.14 Struktureigenschaften der rekursiven Sprachen

Die kontextsensitiven Sprachen sind eine echte Untermenge der rekursiven Sprachen. Die Klasse der rekursiven Sprachen kann in der PS-grammatischen Hierarchie nicht dargestellt werden. Der Grund dafür ist, daß das PS-grammatische Regelschema keinen Restriktionstyp (vgl. 8.1.4) bereithält, dessen zugehörige PS-Grammatiken genau die rekursiven Sprachen generieren würden.

Eine Sprache ist genau dann rekursiv, wenn sie entscheidbar ist, d. h., wenn es einen Algorithmus gibt, der für jede beliebige Eingabe in endlich vielen Schritten entscheiden kann, ob die Eingabe zur Sprache gehört oder nicht. Eine rekursive Sprache, die nicht kontextsensitiv ist (weil sie die generative Kapazität der kontextsensitiven Grammatiken überfordert), ist die sogenannte Ackermann-Funktion.

8.3.15 Struktureigenschaften der unbeschränkten PS-Grammatik

In unbeschränkten PS-Grammatiken kann die rechte Regelseite kürzer als die linke sein, wodurch die Möglichkeit besteht, bereits erzeugte Sequenzen wieder zu *tilgen*. Deshalb ist die Klasse der rekursiv aufzählbaren Sprachen unentscheidbar.

8.4 PS-Grammatik für natürliche Sprachen

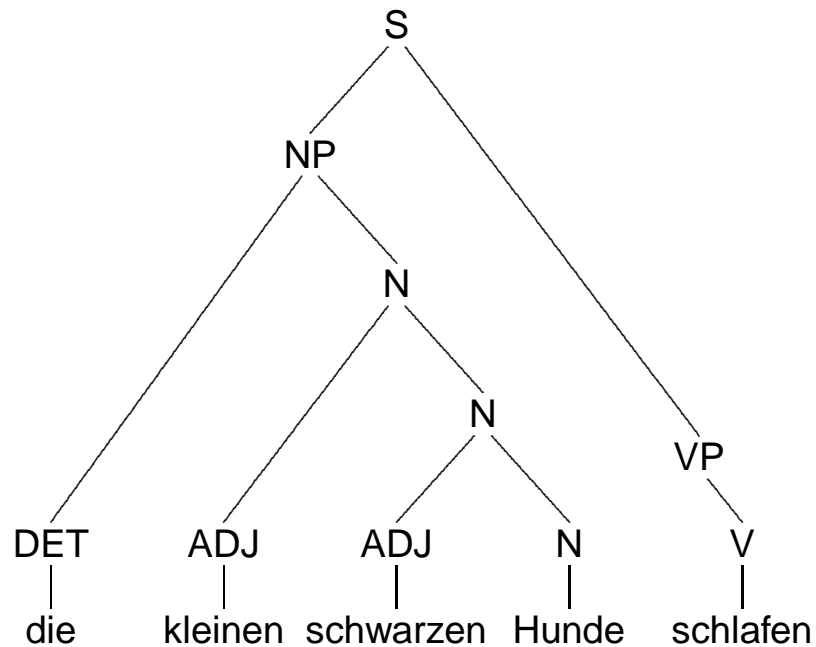
8.4.1 Eine PS-Grammatik für Beispiel 7.5.4

$V =_{def} \{S, NP, VP, V, N, DET, ADJ, \text{die}, \text{Hunde}, \text{kleinen}, \text{schlafen}, \text{schwarzen}\}$

$V_T =_{def} \{\text{die}, \text{Hunde}, \text{kleinen}, \text{schlafen}, \text{schwarzen}\}$

$P =_{def} \{$
 $S \rightarrow NP VP,$
 $VP \rightarrow V,$
 $NP \rightarrow DET N,$
 $N \rightarrow ADJ N,$
 $N \rightarrow \text{Hunde},$
 $ADJ \rightarrow \text{kleinen},$
 $ADJ \rightarrow \text{schwarzen},$
 $DET \rightarrow \text{die},$
 $V \rightarrow \text{schlafen}\}$

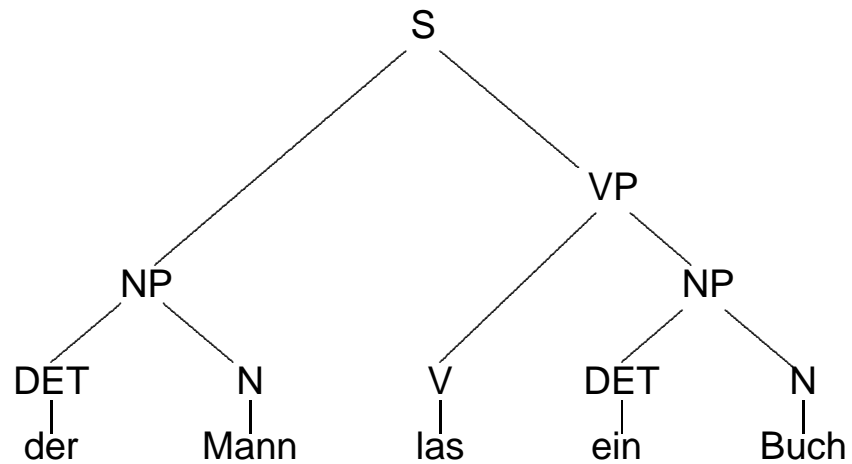
8.4.2 PS-grammatische Analyse von Beispiel 7.5.4



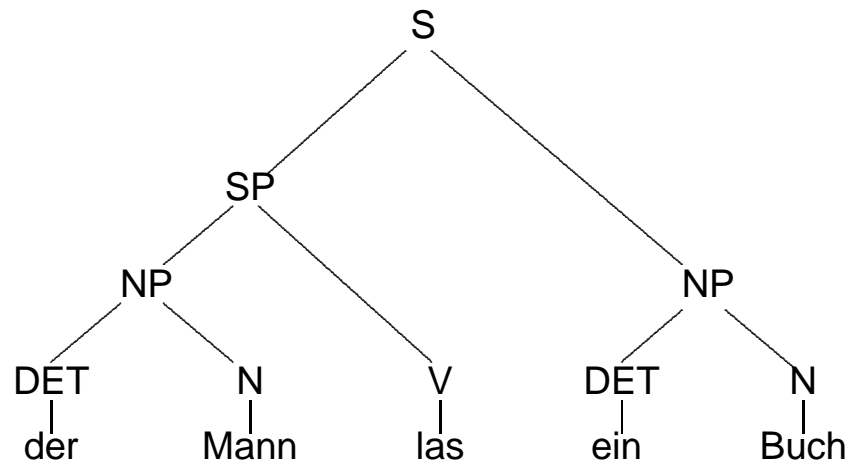
8.4.3 Definition der Konstituentenstruktur

1. Wörter oder Konstituenten, die semantisch zusammengehören, müssen direkt und exhaustiv von einem Knoten dominiert werden.
2. Die Linien einer Konstituentenstruktur dürfen sich nicht überkreuzen (*nontangling condition*).

8.4.4 Akzeptable Konstituentenstrukturanalyse



8.4.5 Nichtakzeptable Konstituentenstrukturanalyse

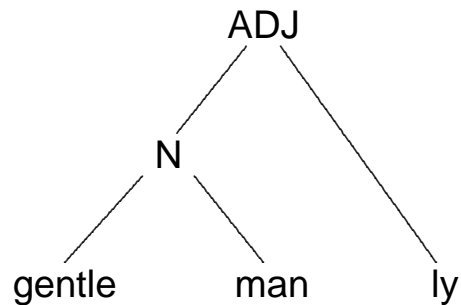


8.4.6 Ursprung der Konstituentenstruktur

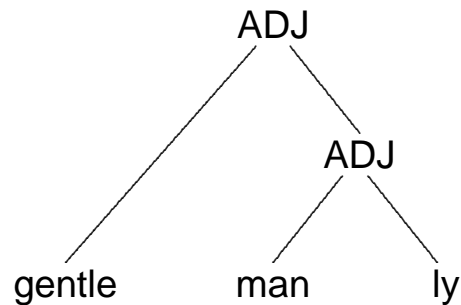
Historisch hat sich der Begriff der Konstituentenstruktur aus der *immediate constituent analysis* des amerikanischen Strukturalisten L. BLOOMFIELD (1887–1949) und den Distributionstests seines Schülers Z. Harris entwickelt.

8.4.7 Immediate constituents in PS-grammar:

Korrekt:



Falsch:



8.4.8 Substitutionsprobe

Akzeptable Substitution:

Susanne liest [ein gutes Buch]

⇓

[eine dicke Zeitung]

Nicht-akzeptable Substitution:

Susanne liest ein [gutes Buch]

⇓

*Susanne liest ein [dicke Zeitung]

8.4.9 Bewegungsprobe

Akzeptable Bewegung:

[der kleine Hund] sieht Julia \implies sieht [der kleine Hund] Julia (?)

Nicht-akzeptable Bewegung:

der [kleine Hund] sieht Julia \implies *der sieht [kleine Hund] Julia

8.4.10 Beabsichtigter Zweck der Konstituentenstruktur

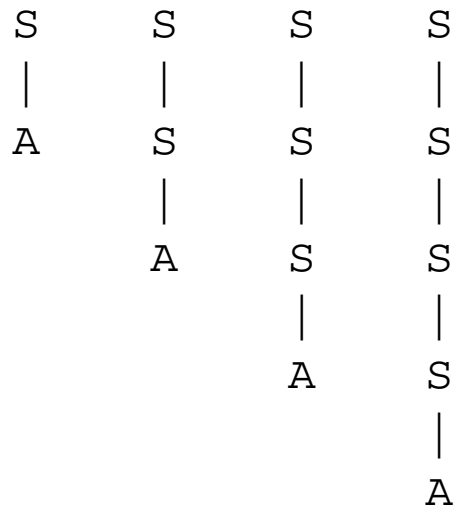
Distributionstests erschienen den amerikanischen Strukturalisten methodisch wichtig zu sein, um ihre Intuitionen über die korrekte Zerlegung (Segmentierung) von Sätzen zu objektivieren. Die Unterscheidung zwischen linguistisch wohlmotivierten und unakzeptablen *immediate-constituent*-Analysen schien wiederum notwendig, weil jeder endlichen terminalen Kette (also jeder Sequenz von Wortformen) *unendlich* viele verschiedene Baumstrukturen zugrunde gelegt werden können.

8.4.11 Unendliche Anzahl von Bäumen über einem einzigen Wort

Kontextfreie Regeln: $S \rightarrow S$, $S \rightarrow A$

Indizierte Klammerung: $(A)_S$, $((A)_S)_S$, $((((A)_S)_S)_S)$, $(((((A)_S)_S)_S)_S)$, etc.

Entsprechende Bäume:

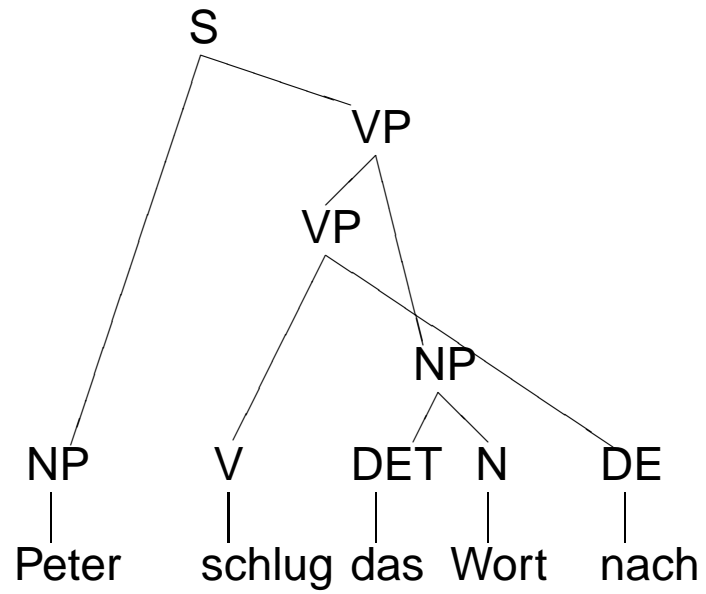


8.5 Konstituentenstrukturparadox

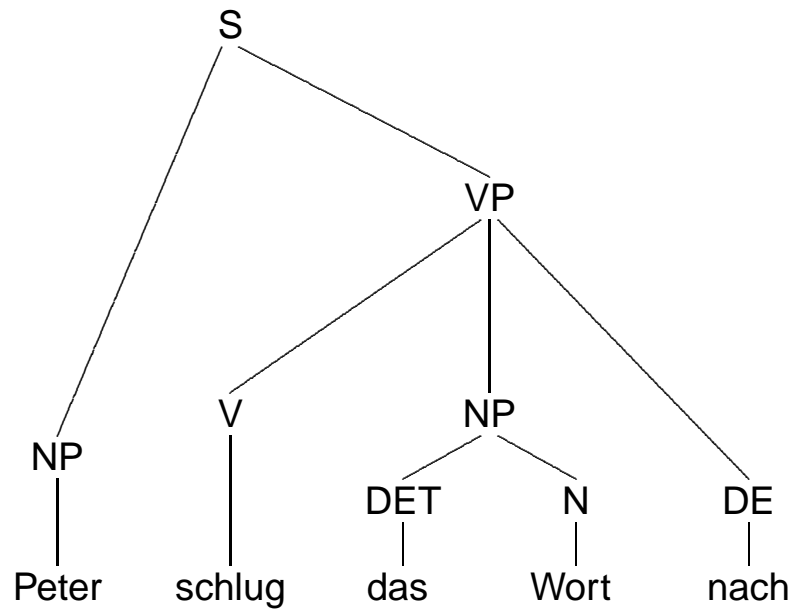
8.5.1 Konstituentenstrukturen aus der Sicht der SLIM-Sprachtheorie

- Konstituentenstrukturen und die Distributionstests, die ihnen zugrundegelegt werden, widersprechen der zeitlinearen Struktur der natürlichen Sprachen.
- Die aus der Konstituentenstrukturanalyse resultierenden Phrasenstrukturbäume haben keinerlei kommunikative Funktion.
- Die Prinzipien der Konstituentenstruktur können bei der empirischen Analyse natürlicher Sprachen nicht immer erfüllt werden.

8.5.2 Verletzung der zweiten Bedingung



8.5.3 Verletzung der ersten Bedingung



8.5.4 Annahmen der Transformationsgrammatik

Um die Konstituentenstruktur als angeboren zu erhalten, unterscheidet die Transformationsgrammatik zwischen hypothetischen Tiefenstrukturen, die angeblich universal sind, und den konkreten sprachabhängigen Oberflächenstrukturen. Dabei wird angenommen,

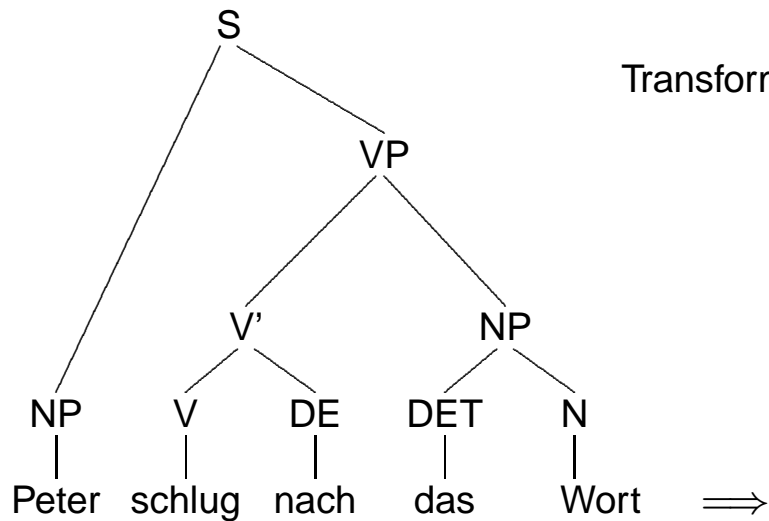
- daß die beiden Ebenen semantisch äquivalent sind,
- daß die Tiefenstrukturen nicht grammatisch wohlgeformt sein müssen, aber die Bedingungen der Konstituentenstruktur erfüllen müssen, und
- daß die Oberflächenstrukturen grammatisch sein müssen, aber nicht die Bedingungen der Konstituentenstruktur erfüllen müssen.

8.5.5 Beispiel einer formalen Transformation

$$[[V \ DE]_{V'} \ [DET \ N]_{NP}]_{VP} \Rightarrow [V \ [DET \ N]_{NP} \ DE]_{VP}$$

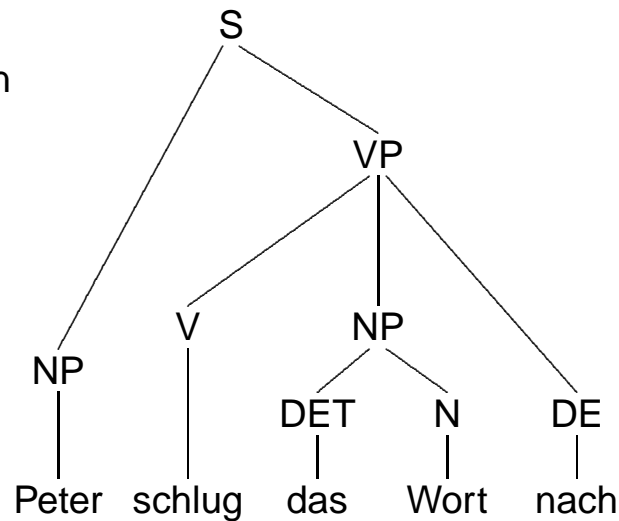
8.5.6 Anwendung der Transformation 8.5.5

Tiefenstruktur:



Transformation

Oberflächenstruktur:



8.5.7 Mathematische Folgen aus dem Einbau von Transformationen in die PS-Grammatik

Während die kontextfreie Tiefenstruktur von niedriger polynomialer Komplexität ist (n^3), hebt der Einbau von Transformationen die Komplexität zu rekursiv aufzählbar. M.a.W., die Transformationsgrammatik ist unentscheidbar.

8.5.8 Beispiel eines Bach-Peters-Satzes

Der Mann, der ihn verdient, bekommt den Preis, den er will.

8.5.9 Tiefenstruktur eines Bach-Peters-Satzes

