

# Probeklausur: Syntax natürlicher Sprachen Lösung

Axel Wisiolek  
Centrum für Informations- und Sprachverarbeitung

21.01.2025

1	Konstituenten- und Adjunkttests . . . . .	2
2	Konstituentengrammatik . . . . .	3
3	X-Bar-Struktur . . . . .	4
4	Parsingalgorithmen . . . . .	5
5	Dependenzstruktur . . . . .	7
6	Syntaktische Funktion . . . . .	8
7	Komplexer Satz . . . . .	9
8	Unifikation und Subsumption . . . . .	11
9	Unifikationsgrammatiken . . . . .	13
10	Statistisches Parsing . . . . .	14
11	Datengestützte Syntaxanalyse . . . . .	16
12	Chunk-Analysen . . . . .	17
	<i>Angabe: Hilfsmittel</i> . . . . .	18

# 1 Konstituenten- und Adjunkttests

## 1. Permutations- und geschehens-Test .....

- (a) Geben Sie für folgenden Satz eine Permutation an, welche das finite Verb als Konstituente bestätigt.

*Das ist ein langer Satz.*

**Lösung:**

*ist das ein langer Satz (?)*

- (b) Gegeben sei folgender Satz, dessen drittes Satzglied den geschehens-Test besteht. Geben Sie (unter Erhalt der Wohlgeformtheit des Ausgangssatzes) ein alternatives drittes Satzglied an, so dass der geschehens-Test fehlschlägt.

*Er wartet im Park.*

**Lösung:**

*Er wartet auf die Sonne.*

*(\*er wartet, und das geschieht auf die Sonne.)*

## 2 Konstituentengrammatik

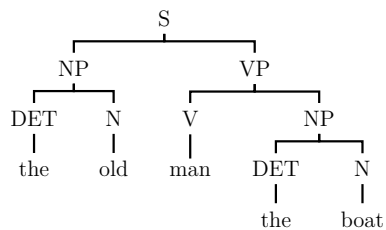
### 2. Temporale Ambiguität .....

Schreiben Sie zu dem folgenden Beispielsatz für temporale syntaktische Ambiguität ein minimale CFG, die die intendierte Struktur des Beispielsatzes erkennt.

*the old (man the boat)VP* (intendierte Struktur)

**Lösung:**

- |   |  |              |
|---|--|--------------|
| 1 |  | S → NP VP    |
| 2 |  | NP → DET N   |
| 3 |  | VP → V NP    |
| 4 |  |              |
| 5 |  | DET → 'the ' |
| 6 |  | N → 'boat '  |
| 7 |  | N → 'old '   |
| 8 |  | V → 'man '   |



### 3 X-Bar-Struktur

#### 3. Erweiterung einer CFG .....

Gegeben sei folgender Satz und eine entsprechende Grammatik.

*der Hund jagt den Briefträger*

1	S	→	NP VP
2	VP	→	V NP
3	NP	→	DET N
4	DET	→	"der"   "den"
5	N	→	"Hund"   "Briefträger"
6	V	→	"jagt"

- (a) Erweitern Sie den Satz der Angabe um ein präpositionales Adverbial.

**Lösung:**

*der Hund jagt den Briefträger um die Stadt*

- (b) Erweitern Sie die Grammatik um entsprechende lexikalische und syntaktische Regeln für den erweiterten Satz. Verwenden Sie das X-Bar-Schema, so dass eine *rekursive Adjunktion* von Adverbialen an die VP ermöglicht wird.

**Lösung:**

1	VP	→	VERBAL
2	VERBAL	→	VERBAL PP
3	PP	→	P NP
4	VERBAL	→	V NP
5			
6	P	→	"um"
7	DET	→	"die"
8	N	→	"Stadt"

## 4 Parsingalgorithmen

### 4. Laufzeit, indirekte Rekursion, Shift-Reduce .....

- (a) Geben Sie CFG-Regeln an, die bei einem Top-Down-Parser wegen Ambiguität zu längerer Laufzeit führen können.

**Lösung:**

1	VP → V
2	VP → V NP
3	VP → V NP NP

- (b) Geben Sie CFG-Regeln an, die eine indirekte Rekursion ermöglichen.

**Lösung:**

1	NP → N PP
2	PP → P NP

- (c) Gegeben sei folgende Grammatik:

1	S → NP VP
2	NP → PROPN
3	NP → DET N
4	VP → V NP
5	VP → V
6	DET → "der"
7	N → "Hund"
8	PROPN → "Max"
9	V → "kennt"

Mit dieser soll ein Shift-Reduce-Parser den Satz *der Hund kennt Max* analysieren. Während des Analysevorgangs wurden bereits die ersten drei Wörter eingelesen; dies ergab den folgenden Stack-Zustand:

V
NP

Der Stack-Zustand ist hier wie in der Vorlesung graphisch so dargestellt, dass neue Elemente immer oben auf den Stack kommen.

- i. Welche Parsing-Strategie verfolgt dieser Parser?
- ii. Welche Operation wird als nächstes ausgeführt?

iii. Wie sieht der Stack nach Ausführen der Operation aus?

**Lösung:**

i. Bottom-Up

- ii.
- zunächst REDUCE (V zu VP); dies führt aber zu keinem vollständigen Parse (da NP-VP auf S zurückgeführt wird, aber "Max" noch nicht verarbeitet)
  - nach Backtracking: SHIFT ("Max")

iii.

Max
V
NP

## 5 Abhängenzstruktur

### 5. Abhängenzgrammatik und Dependency-Parsing .....

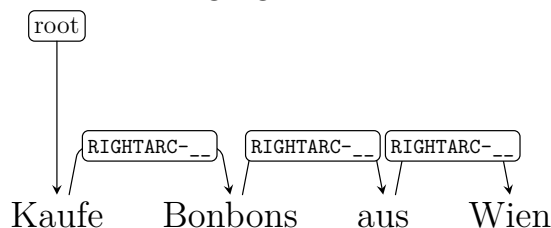
- (a) Schreiben Sie zu dem folgenden Beispielsatz für syntaktische Ambiguität eine ungelabelte Abhängenzgrammatik gemäß der UD-Abhängenzregeln.

*I shot an elephant in my pajamas*

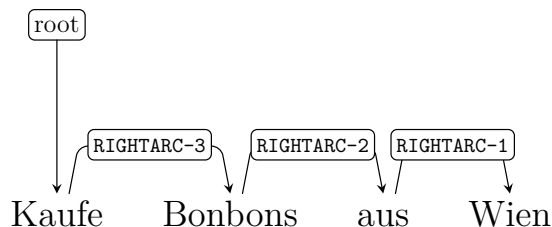
**Lösung:**

1	'shot' → 'I'   'elephant'   'pajamas'
2	'elephant' → 'an'   'pajamas'
3	'pajamas' → 'my'   'in'

- (b) Geben Sie in folgender Abhängenzanalyse die Reihenfolge der Durchführung der 'REDUCE'-Übergänge mit einem Shift-Reduce-Dependency-Parser an.



**Lösung:**

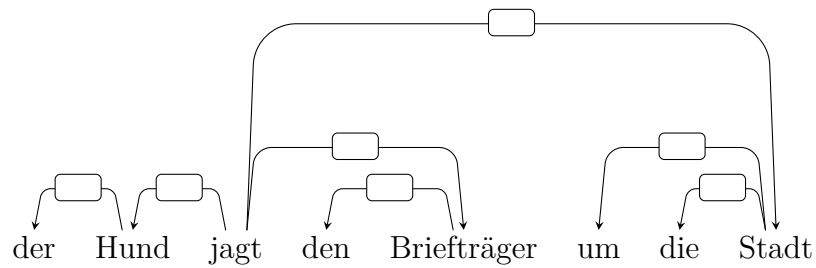


Hinweis zur **RIGHTARC-Regel**: RIGHTARC-Übergang nur durchführen, wenn der Dependent der möglichen Relation nicht Kopf einer der Relationen aus der Menge offener Relationen ist (sonst: SHIFT-Übergang).

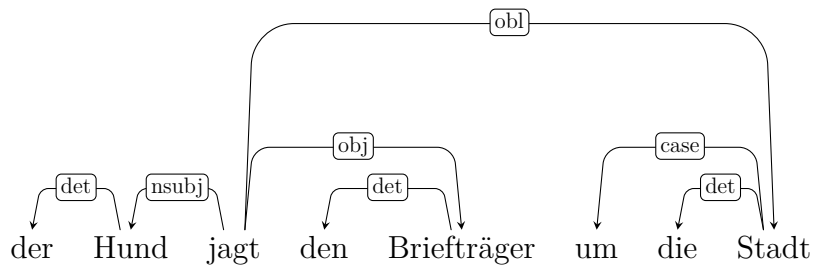
## 6 Syntaktische Funktion

### 6. Dependenzanalyse .....

Vervollständigen Sie die Analyse der Dependenzbeziehungen des folgenden Satzes, indem Sie die entsprechenden UD-Label ergänzen.



**Lösung:**





## 7 Komplexer Satz

### 7. CFG- und UD-Analyse komplexer Satz .....

Gegeben sei folgender Satz:

*der Briefträger schreibt dass der Hund den Briefträger jagt*

- (a) Um welche Art komplexen Satz handelt es sich hier?

**Lösung:**

Objekt-Komplementsatz

- (b) Ergänzen Sie folgende CFG um die entsprechenden lexikalischen und syntaktischen Regeln für die Modellierung des komplexen Satzes.

Verwenden Sie nur **SBAR** und **COMP** als neue Nonterminale.

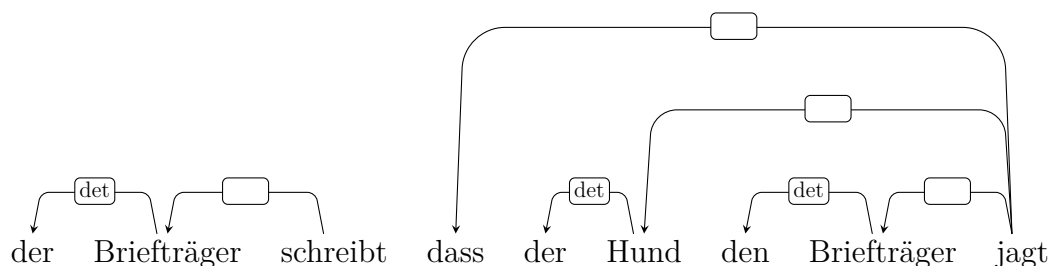
Beachten Sie die invertierte Wortstellung von Verb und Objekt im Nebensatz.

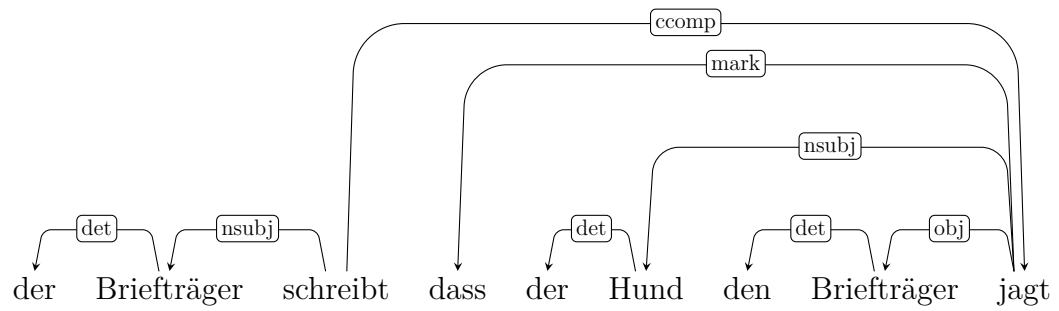
1	S	→	NP VP
2	VP	→	V NP
3	NP	→	DET N
4	DET	→	"der"   "den"
5	N	→	"Hund"   "Briefträger"
6	V	→	"jagt"   "schreibt"

**Lösung:**

1	VP	→	V SBAR
2	SBAR	→	COMP S
3	VP	→	NP V
4	COMP	→	"dass"

- (c) Vervollständigen Sie die Analyse der Abhängigkeitsbeziehungen des komplexen Satzes, indem Sie die fehlenden UD-Label sowie Relation(en) ergänzen.



**Lösung:**

## 8 Unifikation und Subsumption

### 8. Unifikation von Merkmalsstrukturen und Typhierarchie .....

- (a) Gegeben sei folgende Merkmalsstruktur. Geben Sie eine nicht-leere Merkmalsstruktur an, die *nicht* mit der gegebenen unifiziert.

$$\left[ \begin{array}{ll} \text{CASE} & \textit{nom} \\ \text{AGR} & \left[ \begin{array}{ll} \text{GEN} & \textit{mask} \\ \text{PERS} & 1 \end{array} \right] \end{array} \right]$$

**Lösung:**

z.B.  $\left[ \text{AGR} \left[ \text{GEN} \textit{fem} \right] \right]$

- (b) Gegeben sei folgende Typhierarchie, die mit abgekürztem Typnamen durch ein \*CASE\*-Feature implementiert wird.

$$\perp \sqsubseteq \text{Genitiv}$$

$$\perp \sqsubseteq \text{nicht-Genitiv}$$

$$\text{nicht-Genitiv} \sqsubseteq \text{Nominativ-Akkusativ}$$

$$\text{nicht-Genitiv} \sqsubseteq \text{Dativ}$$

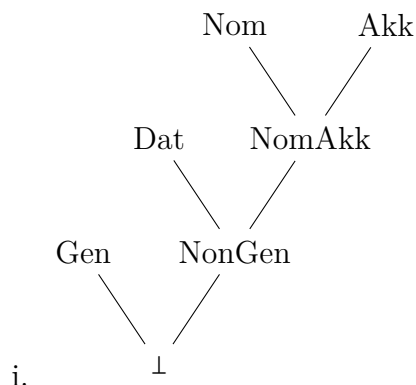
$$\text{Nominativ-Akkusativ} \sqsubseteq \text{Nominativ}$$

$$\text{Nominativ-Akkusativ} \sqsubseteq \text{Akkusativ}$$

- i. Zeichnen Sie die Beziehungen der Typhierarchie als Baumstruktur.
- ii. Geben Sie zu folgender Merkmalstruktur **f1** eine (nicht-leere) Merkmalstruktur **f2** an, sodass gilt: **f2** subsumiert **f1**.

$$\mathbf{f1} = \left[ *CASE* \quad \textit{Nominativ-Akkusativ} \right]$$

**Lösung:**



ii.  $F2 = \left[ *CASE* \quad nicht-Genitiv \right]$

## 9 Unifikationsgrammatiken

### 9. Modellierung von Subkategorisierung, Rektion und Agreement .....

Betrachten Sie folgenden Ausschnitt aus einer merkmalsbasierten Grammatik für einen kleinen Ausschnitt des Englischen.

Beantworten Sie untenstehende Fragen und geben Sie jeweils die Zeilennummern an, auf die sich Ihre Antwort bezieht.

```
1 | S -> NP [NUM=?n] VP [NUM=?n]
2 |
3 | NP [NUM=?n] -> N [NUM=?n]
4 | NP [NUM=?n] -> PropN [NUM=?n]
5 | NP [NUM=?n] -> Det [NUM=?n] N [NUM=?n]
6 | NP [NUM=pl] -> N [NUM=pl]
7 |
8 | VP [TENSE=?t, NUM=?n] -> IV [TENSE=?t, NUM=?n]
9 | VP [TENSE=?t, NUM=?n] -> TV [TENSE=?t, NUM=?n] NP
```

(a) Wie wird hier Subkategorisierung modelliert?

**Lösung:**

durch Erweiterung von CFG-Kategoriensymbolen (Zeilen 8 und 9)

(b) Nennen Sie eine alternative Modellierung von Subkategorisierung.

**Lösung:**

mit SUBCAT-Merkmal als Index ( $V[\text{SUBCAT}=1]$  oder  $V[\text{SUBCAT}=\text{trans}]$ )

(c) Wie wird hier das Subjekt-Verb-Agreement modelliert?

**Lösung:**

Constraint auf Übereinstimmung im NUM-Merkmal von NP und VP in Zeile 1

(d) Geben Sie das entsprechende Constraint für das Subjekt-Verb-Agreement mittels einer Gleichung an (z.B. als Pfadgleichung der Form  $\text{CAT1@Feat} = \text{CAT2@Feat}$ ).

**Lösung:**

$\text{NP@NUM} = \text{VP@NUM}$

## 10 Statistisches Parsing

### 10. PCFG und Mini-Treebank .....

- (a) Geben Sie zu folgendem PCFG-Parsing die Berechnung für die Ableitungswahrscheinlichkeit in Zeile 15 an (\*\* = unkenntlich gemacht).

```

1 grammar = nltk.PCFG.fromstring("""
2     S    -> NP VP                [1.0]
3     VP   -> TV NP                [0.4]
4     VP   -> IV                   [**]
5     VP   -> DatV NP NP           [0.3]
6     TV   -> 'saw'                [1.0]
7     IV   -> 'ate'                [1.0]
8     DatV -> 'gave'                [1.0]
9     NP   -> 'telescopes'         [0.8]
10    NP   -> 'Jack'                [0.2]
11    """)
12 viterbi_parser = nltk.ViterbiParser(grammar)
13 for tree in viterbi_parser.parse(['Jack', 'saw', 'telescopes']):
14     print(tree)
15 (S (NP Jack) (VP (TV saw) (NP telescopes))) (p=0.064)

```

**Lösung:**

$1.0 * 0.2 * 0.4 * 1.0 * 0.8$  oder  $0.2 * 0.4 * 0.8$  oder beliebige Permutationen

- (b) Welchen Wert muss in der PCFG aus der ersten Teilaufgabe das Gewicht für die Regel **VP** → **IV** haben?

**Lösung:**

0.3

- (c) Gegeben sei folgende Mini-Treebank mit PP-Attachment-Sätzen. Welches Attachment bevorzugt die daraus induzierte Grammatik? Begründen Sie.

```

1 (S
2     (NP Ich)
3     (VP (V gehe))
4     (PP auf dem Weg)
5 )
6
7 (S
8     (NP Ich)
9     (VP (V steige) (PP auf den Berg))

```

```
10 | )
11 |
12 | (S
13 |     (NP Ich)
14 |     (VP (V klettere) (PP auf den Berg))
15 | )
16 |
17 | (S
18 |     (NP Ich)
19 |     (VP (V laufe))
20 |     (PP auf dem Weg)
21 | )
22 |
23 | (S
24 |     (NP Ich)
25 |     (VP (V renne))
26 |     (PP auf dem Weg)
27 | )
```

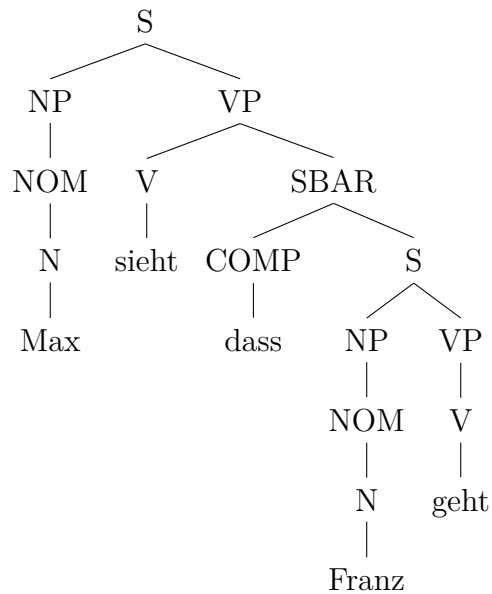
**Lösung:**

PP als Satz-Adjunkt, also Attachment an S (statt VP-Adjunkt, also Attachment an VP), da Frequenz häufiger.

## 11 Datengestützte Syntaxanalyse

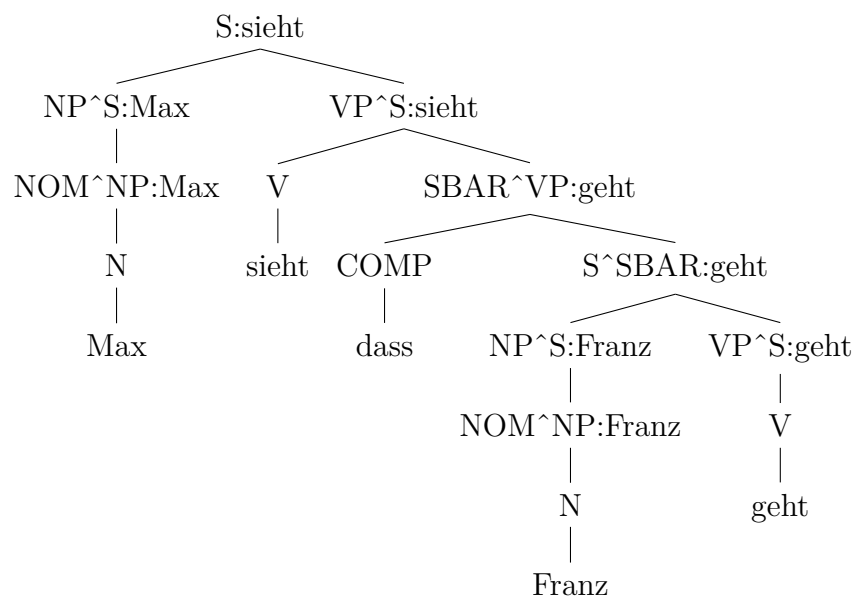
### 11. Annotation für history-based sowie lexikalisierte PCFG .....

Gegeben sei folgender Syntaxbaum:



- Führen Sie zunächst für die syntaktischen Regeln eine Parent-Annotation im Syntaxbaum durch (mit  $\wedge$  als Trennerzeichen).
- Nehmen Sie anschließend für die syntaktischen Regeln eine Kopfannotation im Syntaxbaum vor (mit  $:$  als Trennerzeichen).

**Lösung:**





## 12 Chunk-Analysen

### 12. IOB-Tagging und Evaluationsmetriken .....

- (a) Markieren Sie alle Nominalphrasen (NPs), indem Sie den folgenden Satz nach dem IOB-Tagging-Schema annotieren; verwenden Sie nur folgende Label: B-NP, I-NP, O.

**Lösung:**

Token	Der	kleine	Hund	bringt	ihm	einen	Knochen	.
Tag	B-NP	I-NP	I-NP	O	B-NP	B-NP	I-NP	O

- (b) Berechnen Sie Accuracy, Precision und Recall für folgende korrekte Annotationen (**truth**) und folgende Hypothesen (**predict**). Geben Sie bitte jeweils Brüche an.

Sample	0	1	2	3	4	5	6	7	8	9
truth	PP	PP	PP	0	0	0	0	PP	0	PP
predict	PP	0	0	PP	0	PP	PP	0	PP	PP

**Lösung:**

$$\text{Accuracy: } \frac{3}{10} = 30\%$$

$$\text{Precision (für die Klasse PP): } \frac{2}{6} \approx 33.3\%$$

$$\text{Recall (für die Klasse PP): } \frac{2}{5} = 40\%$$

**Angabe: Hilfsmittel****Universal Dependency Relations**

	Nominals	Clauses	Modifier words	Function Words
Core arguments	<a href="#"><u>nsubj</u></a> <a href="#"><u>obj</u></a> <a href="#"><u>iobj</u></a>	<a href="#"><u>csubj</u></a> <a href="#"><u>ccomp</u></a> <a href="#"><u>xcomp</u></a>		
Non-core dependents	<a href="#"><u>obl</u></a> <a href="#"><u>vocative</u></a> <a href="#"><u>expl</u></a> <a href="#"><u>dislocated</u></a>	<a href="#"><u>advcl</u></a>	<a href="#"><u>advmod</u></a> * <a href="#"><u>discourse</u></a>	<a href="#"><u>aux</u></a> <a href="#"><u>cop</u></a> <a href="#"><u>mark</u></a>
Nominal dependents	<a href="#"><u>nmod</u></a> <a href="#"><u>appos</u></a> <a href="#"><u>nummod</u></a>	<a href="#"><u>acl</u></a>	<a href="#"><u>amod</u></a>	<a href="#"><u>det</u></a> <a href="#"><u>clf</u></a> <a href="#"><u>case</u></a>
Coordination	MWE	Loose	Special	Other
<a href="#"><u>conj</u></a> <a href="#"><u>cc</u></a>	<a href="#"><u>fixed</u></a> <a href="#"><u>flat</u></a> <a href="#"><u>compound</u></a>	<a href="#"><u>list</u></a> <a href="#"><u>parataxis</u></a>	<a href="#"><u>orphan</u></a> <a href="#"><u>goeswith</u></a> <a href="#"><u>reparandum</u></a>	<a href="#"><u>punct</u></a> <a href="#"><u>root</u></a> <a href="#"><u>dep</u></a>

**X-Bar Grundregeln**

- 1 NP → NOM | DET NOM
  - 2 NOM → ADJP NOM | NOM PP
  - 3 NOM → N | N NP
- 
- 1 VP → VERBAL | AUX VERBAL
  - 2 VERBAL → VERBAL PP | VERBAL ADJP
  - 3 VERBAL → V | V NP | V NP NP

**SBAR Grundregeln**

- 1 SBAR → COMP S
- 2 S → SBAR VP
- 3 VP → V SBAR
- 4 S → NP VP SBAR
- 5 NP → NP SBAR

**Koordination Grundregeln**

- 1 S → S CC S
- 2 NP → NP CC NP