

# Wiederholungsaufgaben

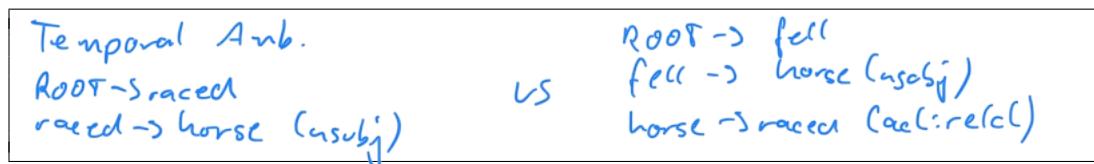
1	Syntaktische Ambiguität	2
2	Syntaktische Kategorien	4
3	Konstituentenstruktur	5
4	Parsingalgorithmen	6
5	Dependenzstruktur	8
6	Dependency Parsing	10
7	Komplexe Sätze	11
8	Grammatische Merkmale, Unifikation und Typ hierarchie	14
9	Unifikationsgrammatiken	18
10	Statistisches Parsing	20
11	Datengestützte Syntaxanalyse	21
12	Partielles Parsing	23

# 1 Syntaktische Ambiguität

## Arten syntaktischer Ambiguität .....

- (a) • Welche Art struktureller Ambiguität liegt in folgendem Satz vor?  
 • Erläutern Sie und geben Sie die Dependenz-Regeln an, die die Ambiguität verursachen.

(1) *The horse raced past the barn fell.*



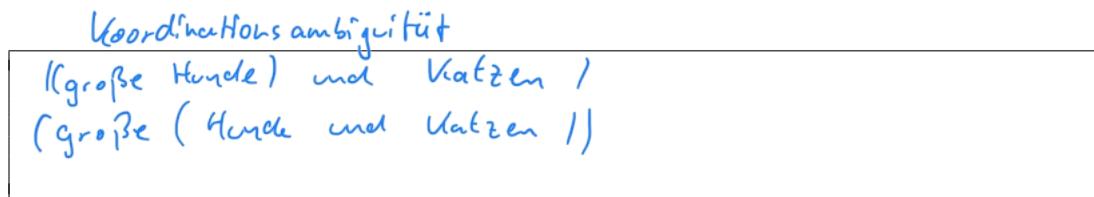
- (b) • Welche Art struktureller Ambiguität liegt in folgendem Satz vor?  
 • Erläutern Sie und geben Sie die CFG-Regeln an, die die Ambiguität verursachen (Label: NP, PP, VP, Det, N, V).

(2) *I shot an elephant in my pajamas.*



- (c) • Welche Art syntaktischer Ambiguität liegt bei folgender NP vor?  
 • Geben Sie für beide Analysen den Klammerausdruck an.

(3) *große Hunde und Katzen*

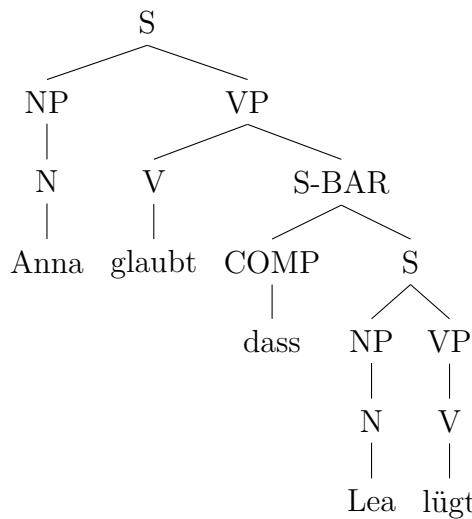


**Rekursionstypen**

- (a) Welche Art der Rekursion ermöglicht die verbale PP-Attachment-Regel  
 $VP \rightarrow VP\ PP$ ?

*Linksrekursion (direkte Rekursion)*

- (b) Erklären Sie anhand der in folgender Ableitung verwendeten Regeln, wie die rekursive Einbettung von Satzkomplementen zustande kommt?



*(S -> NP VP)  
 VP -> V S-BAR  
 S-BAR -> COMP S* → beliebig tief weitere Sätze

- (c)
- Welcher Art ist die Rekursion in (b)?
  - Liegt hier eine *center-embedding*-Konstruktion vor?

*- indirekte Rekursion  
 - kein Center-emb., sondern Edge-emb.:  
 Objektkompl. rechts eingesetzt*

## 2 Syntaktische Kategorien

### Konstituententests .....

- Identifizieren Sie in folgendem Satz in fünf Schritten alle Konstituenten (oberhalb der Wortebene).
- Pro Schritt darf ein Test angewandt werden, pro Schritt dürfen mehrere Konstituenten gleichzeitig identifiziert werden.
- Geben Sie zu jedem Schritt den verwendeten Test an.

(4) *Das Auto hält an der rot leuchtenden Ampel.*

Schritt 1: *Ersetzungstest*: Das Auto  $\in$  die Ketsche; hält ... Ampel  $\in$  steht

Schritt 2: *Eliminierungstest*: ... hält an der tol Ampel

Schritt 3: *Koordinations test*: .. hält und wortet an...

Schritt 4: *Permutations test*;

Schritt 5: *Substitutionstest*:

- Geben Sie zum Schluss das daraus folgende Kastendiagramm an; verändern Sie den Satz hierbei nicht!

Kastendiagramm:

Das   Auto   hält   an   der   roten   Ampel
Das / Auto / hält / an / der / rot / leuchtenden / Ampel
Das / Auto / hält / an / der / rot / leuchtenden / Ampel
Das / Auto / hält / an / der / rot / leuchtenden / Ampel
Das / Auto / hält / an / der / rot / leuchtenden / Ampel
Das / Auto / hält / an / der / rot / leuchtenden / Ampel

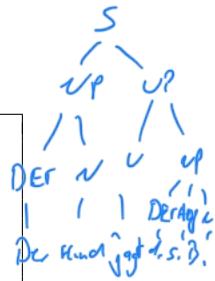
### 3 Konstituentenstruktur

#### Kontextfreie Grammatiken und Syntaxbäume .....

- Erstellen sie zu dem folgenden Satz eine kontextfreie Grammatik und den dazugehörigen Syntaxbaum.
- Verwenden Sie nur folgende syntaktische Kategorien: S, NP, VP, N, DET, ADJ, V

(5) *Der Hund jagt den schreienden Briefträger.*

$S \rightarrow NP\ VP$	$V \rightarrow "jagt"$
$VP \rightarrow V\ NP$	$ADJ \rightarrow "schreiende"$
$NP \rightarrow DEF\ N \mid DEF\ ADJ\ N$	
$DEF \rightarrow "Der"$ $\mid$ $"den"$	
$N \rightarrow "Hund" \mid "Briefträger"$	



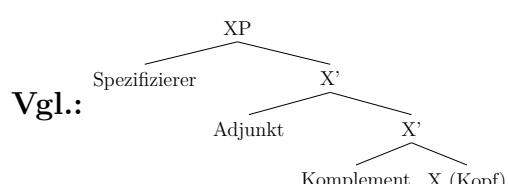
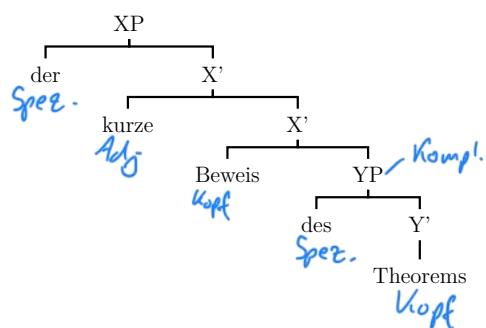
#### Erweiterung von kontextfreien Grammatiken .....

- Die Grammatik aus der vorherigen Aufgabe soll so erweitert werden, dass rekursiv PP-Adjunkte an NPs auftreten können (der Art *Franz aus München in Bayern* ...).
- Ergänzen Sie dazu im Folgenden die fehlenden rechten Seiten (PP-Regel ist gegeben; es dürfen keine neuen Kategorien eingeführt werden).

1	$PP \rightarrow P\ NP$
2	$NP \rightarrow NOM$
3	$NOM \rightarrow NOM\ PP$
4	$NOM \rightarrow N$

#### X-Bar-Konstituentenstruktur .....

Was sind in folgendem Syntaxbaum im X-Bar-Strukturschema **Spezifizierer, Komplemente, Adjunkte und Köpfe**?



## 4 Parsingalgorithmen

### Shift-Reduce-Parser .....

Gegeben sei folgende Grammatik:

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

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.

- (a) Welche Parsing-Strategie verfolgt dieser Parser?

bottom-up

- (b) Welche Operation wird als nächstes ausgeführt?

Reduce ( $V \rightarrow VP$ ) → nicht erfolgreich → Backtrack  
Shift

- (c) Wie sieht der Stack nach Ausführen der Operation aus?

NP, V, Max

### Recursive-Descent-Parser .....

Gegeben sei folgende Grammatik: wie oben

Mit dieser soll ein Recursive-Descent-Parser den Satz *der Hund kennt Max* analysieren.

- (a) Welche der beiden grundlegenden Parsing-Strategien verfolgt dieser Parser?

*top-down*

- (b) Welche zwei Operationen stehen diesem Parser zur Verfügung?

- Predict  
- Scan

- (c) Nennen Sie eine Folge von Regelanwendungen, die zu Backtracking führt.

$S \rightarrow NP VP$        $NP \rightarrow S' "Max" g$   
 $NP \rightarrow PNP$        $\rightarrow$  Backtrack

### Earley-Parser .....

Gegeben sei folgende Grammatik: wie oben.

Mit dieser soll ein Earley-Parser den Satz *der Hund kennt Max* analysieren. Während des Analysevorgangs wird folgende Zustandsmenge  $Q_1$  erzeugt:

$$(DET \rightarrow \cdot, 0) \quad (1)$$

$$(NP \rightarrow DET \cdot N, 0) \quad (2)$$

$$(N \rightarrow \cdot \text{ Hund}, 1) \quad (3)$$

- (a) Welche Operation wird als nächstes ausgeführt?

*Scan*

- (b) Welcher Zustand wird dabei hinzugefügt?

$(N \rightarrow \cdot \text{ Hund}, 1)$

- (c) Welcher Zustandsmenge wird der Zustand hinzugefügt?

$Q_1$

## 5 Dependenzstruktur

### Tests zur Komplement/Adjunkt-Unterscheidung .....

- (a) Nennen Sie die drei Tests zur Komplement/Adjunkt-Unterscheidung. Was wird geprüft? Wann handelt es sich jeweils um ein Adjunkt?

Geschehenstest; Satz unklaren  $\rightarrow$  erfolgreich: Adjunkt  
 Eliminierungs test: Teil wegnehmen. wenn wohlgeformt: Teil ist Adjunkt  
 Adverbialsatztest: unklarer zu Adv.-satz: wenn wohlgeformt: Adjunkt

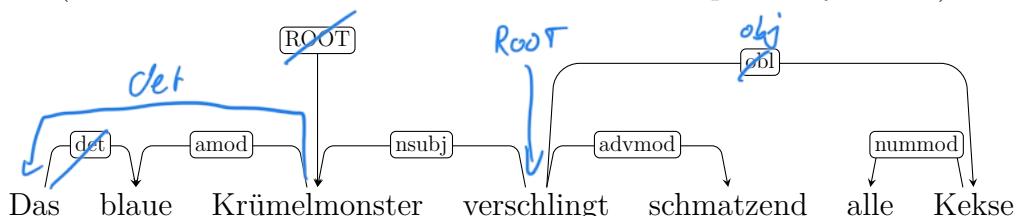
- (b) Bestimmen Sie für zwei verschiedene Konstituenten des folgenden Satzes, ob es sich jeweils um ein Komplement oder ein Adjunkt handelt. Verwenden Sie hierzu zwei der eben genannten Tests (für jede Konstituente einen).

(6) *Das blaue Krümelmonster verschlingt schmatzend alle Kekse im Park.*

"schmatzend": Eliminierungs test: wohlgeformt  $\rightarrow$  Adjunkt  
 "im Park": Geschehenstest: ..., und das geschah im Park V  $\rightarrow$  Adj.

### Dependenzbäume .....

- Folgender Dependenzbaum enthält drei Fehler; diese können sich sowohl auf die Relationen als auch die Relationslabel beziehen; der Baum soll die UD-Konventionen erfüllen.
- Finden Sie die drei Fehler; markieren Sie dazu die falsche Relation bzw. schreiben Sie das korrekte Label über das falsche.
- (siehe letzte Seite für eine Übersicht der UD-Dependency-Labels)



1: ROOT  
 2: det  
 3: obj statt obj

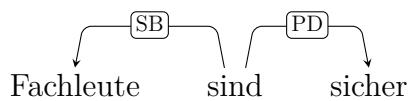
**Komplement vs. Modifikator**

- Bestimmen Sie für den folgenden Satz die Abhängigkeiten. Markieren Sie, ob es sich bei der jeweiligen Abhängigkeit um ein Komplement oder einen Modifikator handelt.

(7) Das Auto hält an der roten Ampel

**UD-Schema vs. TIGER-Schema**

- Wandeln Sie folgenden Dependenzbaum im TIGER-Dependenz-Schema um in das *Universal-Dependencies*-Schema (*primacy of content word*-Maxime).
- (siehe letzte Seite für eine Übersicht der UD-Dependency-Labels)



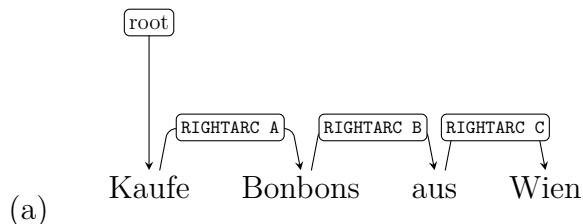
```

graph TD
    sicher -- ROOT --> sind
    sind -- cop --> fachleute
    fachleute -- usubj --> fachleute
  
```

## 6 Dependency Parsing

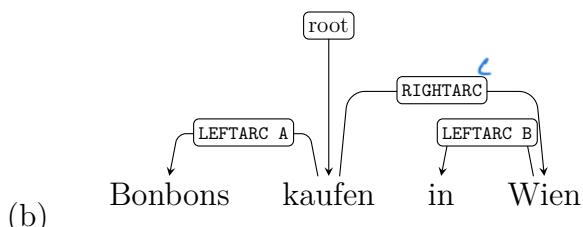
### Übergangsbasierter Shift-Reduce-Dependency-Parser .....

- In welcher Reihenfolge werden im Folgenden jeweils die angegebenen REDUCE-Übergänge durchgeführt?
- Begründen Sie.
- Wie unterscheiden sich die beiden Syntaxbäume?



C, B, A

Rightarc wird nur ausgeführt, wenn Dep. nicht selbst Kopf ist



A, B, Rightarc

Leftarc-Red. finden direkt statt. C erst, wenn 'in' "wegerziert" wurde.

a - Bonbons aus Wien

b - kaufen in Wien

## 7 Komplexe Sätze

### Konstituentenstruktur komplexer Sätze .....

- Geben Sie zu folgenden Regeln bzw. Regelgruppen für **komplexe Sätze des Englischen** einer Zeile jeweils deren Funktion an - welcher Typ komplexer Sätze wird jeweils erzeugt?
- Geben Sie zusätzlich auch die entsprechenden UD-Kantenlabel an.

S-BAR → COMP S : S-BAR Grundregel (marg, usbj ...)

NP → NOM, NOM → NOM S-BAR : Relativsatz (acl:relcl)

VP → V S-BAR : Objektkomplementsatz (ccomp)

S → S-BAR VP : Subjektkompl.s. (csubj)

S → S-BAR NP VP : Adverbialsatz (advcl)

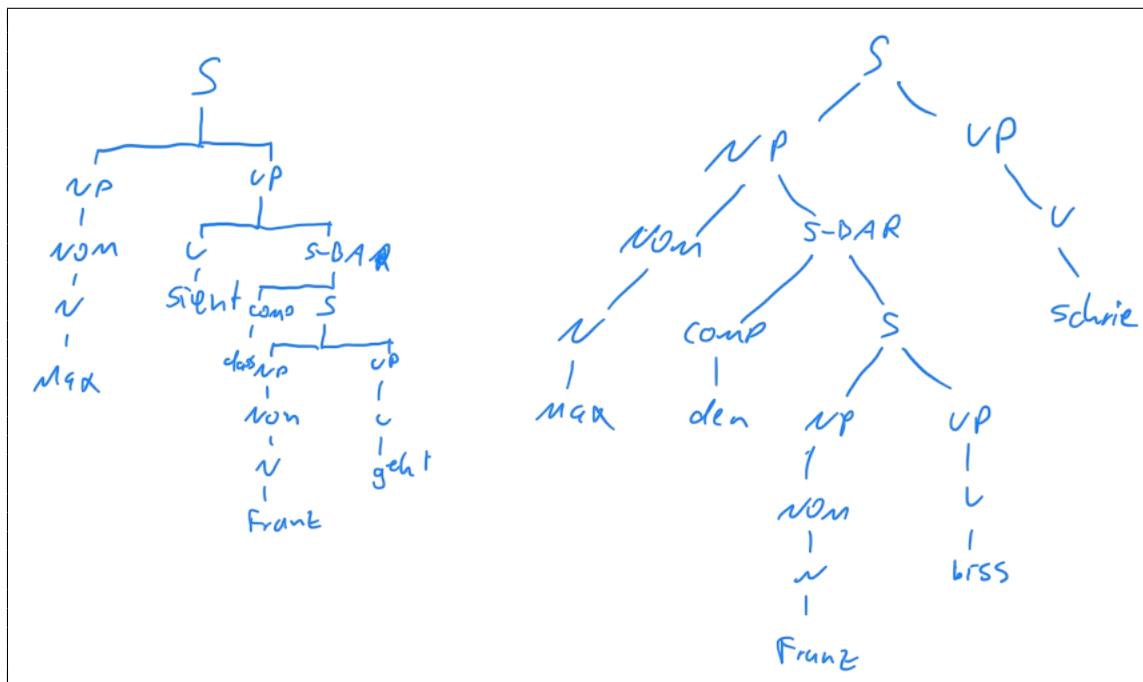
- Erstellen Sie die **Phrasenstrukturbäume** zu den folgenden beiden Sätzen; verwenden Sie dazu die dem eingebetteten Satztyp entsprechenden Regeln von oben, sowie zusätzlich folgende Regeln:

S → NP VP, NOM → N, VP → V

(8) Max (N) sieht (V) dass (COMP) Franz (N) geht (V)

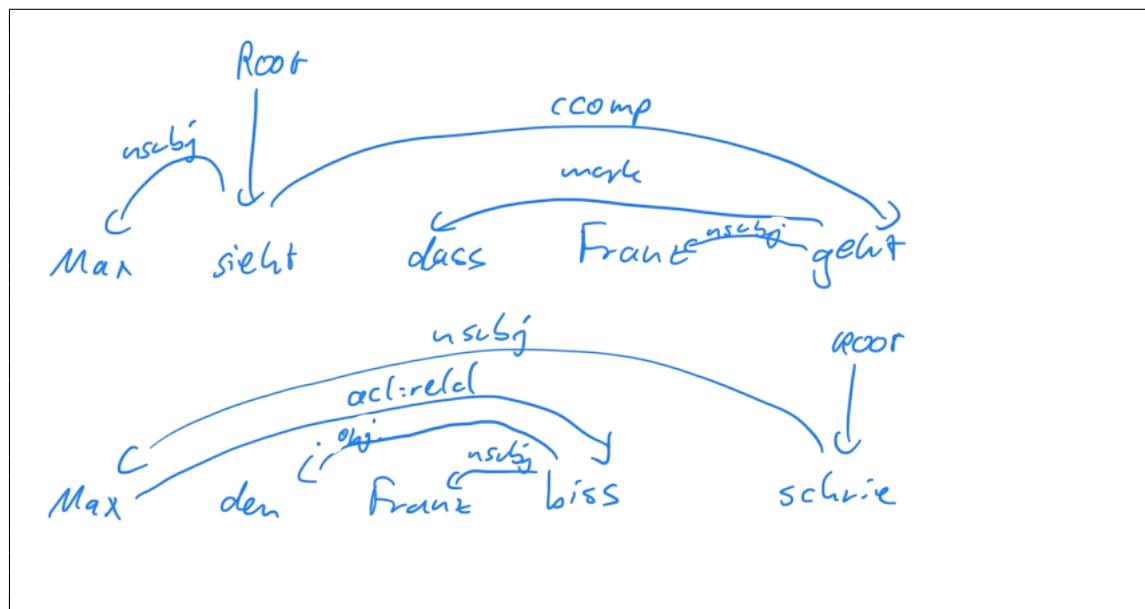
(9) Max (N), den (COMP) Franz (N) biss (V), schrie (V)

- Verwenden Sie die in Klammern angegebenen lexikalischen Kategorien!



## Dependenzstruktur komplexer Sätze 1 .....

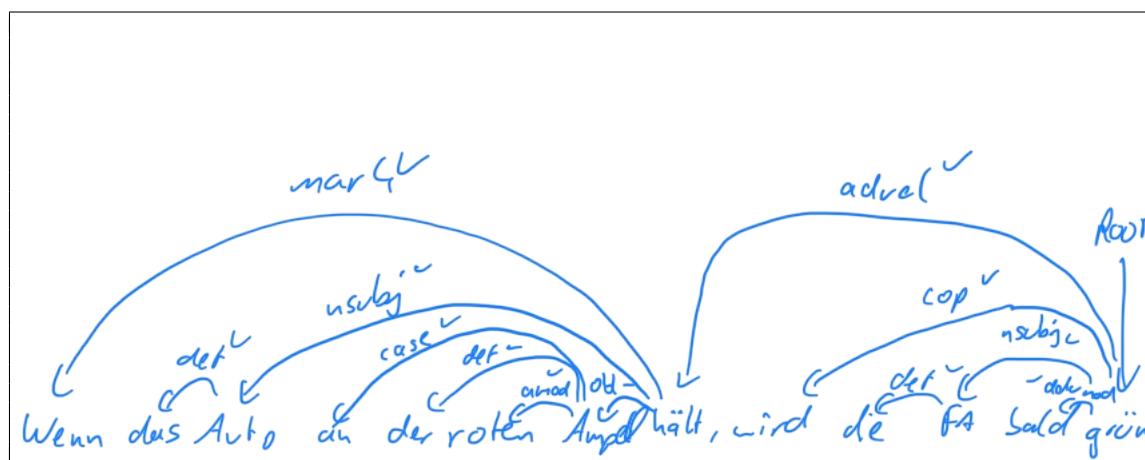
- Erstellen Sie zu den Sätzen (8) und (9) auch die entsprechenden Dependenzbäume (Kopf von S = VP-Kopf; Kopf von S-BAR = Kopf von S).
- Verwenden Sie die im Anhang auf der letzten Seite angegebenen UD-Dependency-Label.



## Dependenzstruktur komplexer Sätze 2 .....

- Bestimmen Sie alle Dependenzrelationen im folgenden Satz.
- Bestimmen Sie außerdem, um welche grammatische Relation es sich jeweils handelt.

(10) *Wenn das Auto an der roten Ampel hält, wird die Fußgängerampel bald grün.*



**Infinite Konstruktionen: Subjekt- vs. Objektkontrolle .....**

Liegt in folgendem Satz Subjekt- oder Objektkontrolle vor? Begründen Sie.

- (11) *Sie überredeten ihn, noch zu bleiben*

Objektkontrolle

↳ Objekt Matrixsatz = Subjekt Infinitivsatz

## 8 Grammatische Merkmale, Unifikation und Typ-hierarchie

### Kodierungstypen syntaktischer Funktionen .....

Wie werden in den unteren transitiven Satzpaaren einer *Kunstsprache mit englischem Lexikon* die grammatischen Relationen **Subjekt** bzw. **Objekt** jeweils kodiert?

- Geben Sie jeweils eine der folgenden drei Kodierungsarten an:
  - **Kasusmorphologie** (*morphologisch* = substantielle Kodierung am Dependenten)
  - **Verbale Kongruenz** (*morphologisch* = substantielle Kodierung am Verb)
  - **Wortstellung** (strukturelle Kodierung)

(12)    bird    cat    eat  
       'Die Katze frisst den Vogel.'

(13)    cat    bird    eat  
       'Der Vogel frisst die Katze.'

*Wortstellung*

(14)    bird-fe    cat    eat  
       'Die Katze frisst den Vogel.'

(15)    bird    cat-fe    eat  
       'Der Vogel frisst die Katze.'

*Kasusmorphologie*

(16)    bird-i    cat-o    eat-o  
       'Die Katze frisst den Vogel.'

(17)    bird-i    cat-o    eat-i  
       'Der Vogel frisst die Katze.'

*Verbale Kongruenz*

**Kasusrektion, Agreement und Subkategorisierung** .....

Welche der folgenden morphosyntaktischen Beschränkungen ist in den unteren Sätzen jeweils verletzt?

- **Kasusrektion**
- **Nominale Kongruenz**
- **Verbale Kongruenz (Agreement)**
- **Subkategorisierung** (Anzahl und Art der verbalen Argumente)

(18) *Der Auto fährt schnell.*

*Nom. Kongruenz*

(19) *Den Auto fährt schnell.*

*Kasusrektion*

(20) *Die Autos fährt schnell.*

*Verbale Kongr.*

(21) *Das Auto fährt, dass es ankommt.*

*Subkat.*

**Wortstellung: Stellungsfeldermodell** .....

(a) Identifizieren Sie in den folgenden Sätzen durch Unterstreichen das Mittelfeld und, falls vorhanden, das Vorfeld:

*VF*

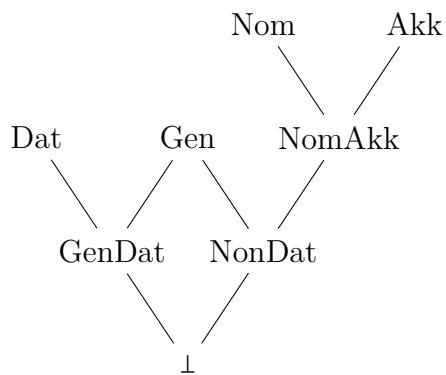
(22) *Es hat Kuchen gegeben.*

(23) *Hat es Kuchen gegeben?*

(b) Um welches *Es* handelt es sich in (22)?

- **Topik-Es** = Vorfeld-Platzhalter
- **Subjekt-Expletiv**
- **pronominaler Ersatz** für NP

*Subjekt Expletiv*

**Unifikationsparsing und getypte Merkmalstrukturen**

Gegeben sei folgende Typhierarchie:

**Typhierarchien**

Unifizieren Sie die folgenden Paare von Typen. Typen, die nicht unifizieren, markieren Sie als *undefiniert*.

(a) Nom  $\sqcup$  Akk = *undef.*

(b) GenDat  $\sqcup$  NonDat = *Gen*

(c) GenDat  $\sqcup$  Gen = *Gen*

(d) Nom  $\sqcup$  ⊥ = *Nom*

(e) Akk  $\sqcup$  NomAkk = *Akk*

**Subsumption**

Gegeben seien nun zusätzlich folgende Merkmalstrukturen mit  $\theta(\text{FS1}) = \theta(\text{FS2}) = \theta(\text{FS3}) = \perp$ .

$$\text{FS1} = \begin{bmatrix} \text{CAS} & \text{NomAkk} \\ \text{GEN} & \text{mask} \end{bmatrix} \quad \text{FS2} = \begin{bmatrix} \text{CAS} & \text{Nom} \\ \text{GEN} & \text{mask} \\ \text{PER} & 3 \end{bmatrix} \quad \text{FS3} = \begin{bmatrix} \text{CAS} & \text{Akk} \\ \text{PER} & 3 \end{bmatrix}$$

Entscheiden Sie jeweils mit *ja* oder *nein*:

(a) FS2  $\sqsubseteq$  FS3 ? *nein*

(b) FS1  $\sqsubseteq$  FS2 ? *ja*

(c) FS2  $\sqsubseteq$  FS1 ? *nein*

(d) FS3  $\sqsubseteq$  FS2 ? *nein*

(e)  $\text{FS3} \sqsubseteq \text{FS3}$  ? ja

### Unifikation .....

Unifizieren FS2 und FS3? Falls ja, unifizieren Sie; falls nein, begründen Sie.

*Nein, untersch. Kasus (Nom vs. Akk)*

### Bedingungen .....

Entscheiden Sie jeweils mit *ja* oder *nein*:

(a)  $\text{FS2} \vDash \text{CAS:}\perp$  ? ja

(b)  $\text{FS2} \vDash \text{GEN:neut}$  ? nein

(c)  $\text{FS3} \vDash \text{PER:3}$  ? ja

(d)  $\text{FS1} \vDash \text{NomAkk} \wedge \text{mask}$  ? nein

(e)  $\text{FS2} \vDash \text{CAS:NomAkk}$  ? ja

## 9 Unifikationsgrammatiken

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

- Betrachten Sie folgenden Ausschnitt aus einer merkmalsbasierten Grammatik für einen kleinen Ausschnitt des Englischen (s. NLTK/book\_grammars/feat0.fcfg).
- 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?

*CFG - Kategorisymbole*

(b) Nennen Sie eine alternative Modellierung von Subkategorisierung.

*Subcat-Merkmale als Index*

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

*NUM-Übereinstimmung zw. NP & VP*

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

*NP@NUM = VP@NUM*

## Erweiterung von kontextfreien Grammatiken um Merkmale .....

- Gegeben sei folgende Grammatik:

1	-> ADV VP
2	VP -> V NP PP
3	PP -> P NP
4	NP -> N
5	NP -> PROPN
6	ADV -> "gestern"
7	ADV -> "heute"
8	V -> "ging"
9	V -> "geht"
10	PROPN -> "Fritz"
11	N -> "Arbeit"
12	P -> "zur"

- Mit dieser können die folgenden Sätze hergeleitet werden:

*gestern ging Fritz zur Arbeit*  
*heute ging Fritz zur Arbeit*  
*heute geht Fritz zur Arbeit*

- Passen sie die Grammmatik so an, dass Imperfekt und Präsens als Merkmale in der Grammatik unterschieden werden können, so dass der folgende Satz nicht mehr erkannt wird:

(26) \**gestern geht Fritz zur Arbeit*

1	<i>S</i>	-> ADV [TIME=?x] VP [TIME=?x]	
2	VP [TIME=?x]	-> V [TIME=?x]	NP
3	PP	-> P	NP
4	NP	-> N	
5	NP	-> PROPN	
6	ADV [TIME=imp]	-> "gestern"	
7	ADV	-> "heute"	
8	V [TIME=imp]	-> "ging"	
9	V [TIME=pr]	-> "geht"	
10	PROPN	-> "Fritz"	
11	N	-> "Arbeit"	
12	P	-> "zur"	

## 10 Statistisches Parsing

PCFG: Gewichte und Ableitungswahrscheinlichkeit .....

Betrachten Sie folgendes PCFG-Parsing (\*\* = 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)

```

- (a) Geben Sie die Berechnung für die Ableitungswahrscheinlichkeit in Zeile 15 an?

$$1 \cdot 0.2 \cdot 0.4 \cdot 1 \cdot 0.8 = 0.064$$

- (b) Welchen Wert muss das Gewicht für die Regel **VP → IV** haben?

$$0.3$$

## 11 Datengestützte Syntaxanalyse

## Datengestützte Methoden: Abschätzung Regelwahrscheinlichkeiten .....

- (a) Für ein großes Korpus sei lediglich *part of speech* (POS) annotiert; Syntaxbäume stehen nicht zur Verfügung.

Welche Arten von Regeln einer kontextfreien Grammatik kann man mit diesen Daten automatisch generieren?

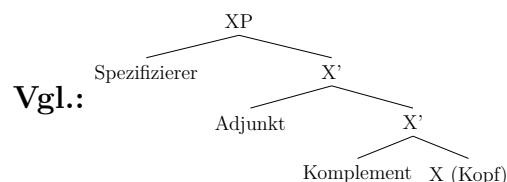
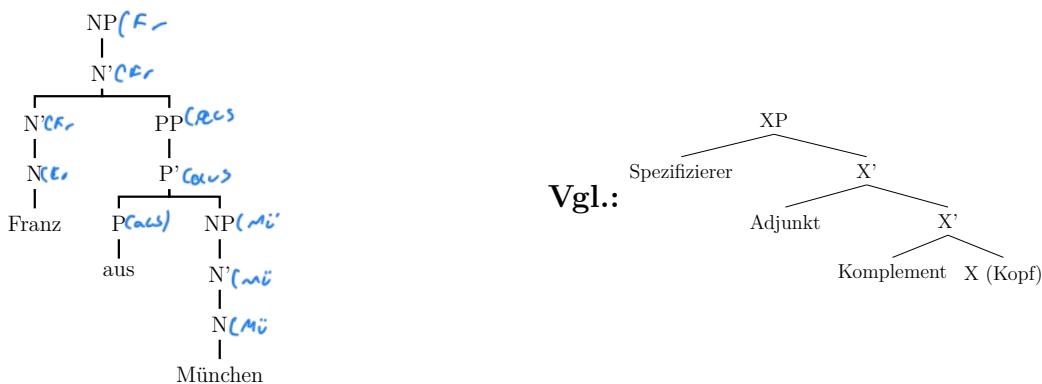
## Lexikalische Regeln

- (b) Folgende Häufigkeiten wurden aus einem Datensatz gezählt:  
 $\text{count}(VP \rightarrow V) = 200$ ,  $\text{count}(VP \rightarrow VNP) = 100$ ,  $\text{count}(VP \rightarrow \text{``*''}) = 300$ . Berechnen Sie  $P(VNP | VP)$  mit der MLE-Methode.

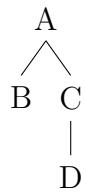
$$\frac{100}{300} = \frac{1}{3} \times 0,333 = 0,333$$

Methoden für lexikalisierte und *history-based* PCFGs .....

- (a) Führen Sie im linken Syntaxbaum eine Kopfannotation durch; Geben Sie anschließend die lexikalisierte Regel für den Wurzelknoten an. Orientieren Sie bei der Kopfannotation an der Strukturposition des Kopfes im X-Bar-Schema (vgl. rechter Syntaxbaum).



- (b) Führen Sie für die CFG-Regeln  $\mathbf{C} \rightarrow \mathbf{D}$  und  $\mathbf{A} \rightarrow \mathbf{B}$  in unterem Syntaxbaum *parent-annotation* durch.



$C^A \rightarrow D^C$   
 $A \rightarrow B^A$

## 12 Partielles Parsing

### Chunking .....

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

Token	Der	junge	Mann	gab	ihr	das	Buch	.
Tag	B-NP	I-NP	I-NP	O	B-NP	B-NP	I-NP	O

### Kaskadierende Chunk-Parser .....

- (a) Mit welcher Methode kann z.B. folgende hierarchische Struktur einer Präpositionalphrase mit flachen Chunk-Parsern erzeugt werden:

[PP auf/P [NP dem/DET Baum/N] ]

*Kaskadierender Chunk-Parser*

### Evaluationsmetriken .....

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	O	O	O	O	PP	O	PP
predict	PP	O	O	PP	O	PP	PP	O	PP	PP

Accuracy:  $\frac{3}{10}$

Precision (für die Klasse PP):  $\frac{2}{6} = \frac{1}{3}$

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

## Angabe: Hilfsmittel

### Universal Dependency Relations

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