

Wiederholungsaufgaben 1: Themen 1-9

Lösung

1 Automatische Syntaxanalyse A

Grammatiktypen

- Nennen Sie die wichtigsten differenzierenden Merkmale der unten genannten Modellierungen der Syntax natürlicher Sprachen.
- Berücksichtigen Sie dabei auch folgende Kriterien:
 - Gegenstand der Strukturanalyse
 - Analysetiefe (hierarchische vs. flache Strukturanalyse)
 - formale Modellierung
 - Verarbeitung / verwendbare Parsingalgorithmen

1. Phrasenstrukturgrammatik

Lösung:

- Modellierung der Phrasenstruktur natürlicher Sprache
 - Zerlegung Satz in Einheiten (Konstituenten/Phrasen)
- hierarchische Konstituenten-Strukturanalyse
 - Strukturinformationen in den Knoten (NP, VP usw.)
- formale Modellierung mit Kontextfreien Grammatiken
- Verarbeitung mit verschiedenen CFG-Parsing-Algorithmen

2. Dependenz-Grammatik

Lösung:

- Modellierung der syntaktischen Abhängigkeitsrelationen zwischen Wörtern (Wortgrammatik)
- hierarchische Dependenz-Strukturanalyse
 - Strukturinformationen in den Kanten = Label der Relationen (syntaktische Funktionen)
- formale Modellierung u.a. mit (lexikalisierten) formalen Grammatiken oder statistischen Modellen (z. B. *transition-based*-Modelle)
- Verarbeitung von *transition-based*-Modellen mit SHIFT-REDUCE-Parser (SHIFT-RIGHTARC-LEFTARC)

3. Felder-Modell

Lösung:

- Modellierung der linearen Anordnung der Satzglieder im Deutschen
 - Zerlegung Satz in Einheiten (Satzglieder)
- nicht-hierarchische Strukturanalyse
 - Analyse der möglichen Positionen (ausgehend von der Satzklammer)
- linguistisches Modell (kein formales Modell)

2 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.*

Lösung:

- Temporale Ambiguität

```

1 | ROOT → raced
2 | raced → horse (nsubj)
  |
vs.
1 | ROOT → fell
2 | fell → horse (nsubj)
3 | horse → raced (acl:relcl)
  
```

- (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.*

Lösung:

- PP-Attachment-Ambiguität

```

1 | NP → Det N | Det N PP
2 | VP → V NP | VP PP
  
```

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

(3) *große Hunde und Katzen*

Lösung:

- Koordinationsambiguität
- (große (Hunde und Katzen)) **vs.** ((große Hunde) und Katzen)

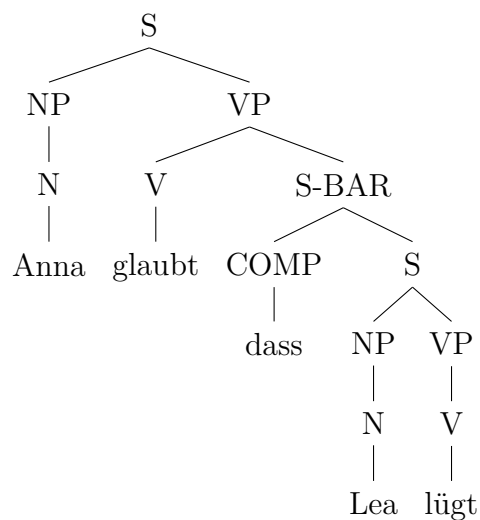
Rekursionstypen

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

Lösung:

direkte Rekursion (linksrekursiv): VP ist unmittelbare Konstituente von VP

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



Lösung:

- durch die Regeln $VP \rightarrow V S\text{-}BAR$ und $S\text{-}BAR \rightarrow COMP S$ können beliebig tief weitere Sätze (S) in den Hauptsatz (S) als Objektkomplemente eingebettet werden.

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

Lösung:

- indirekte Rekursion (eingebettetes S ist nicht unmittelbare Konstituenten von S)
- kein *center-embedding* (sondern *edge-embedding*):
 - die Objektkomplemente sind rechts eingebettet
 - dagegen *center-embedding*, z.B. bei Relativsätzen:
Die Katze, die der Hund biss, lief weg.
Die Katze, die der Hund, der entlaufen war, biss, lief weg.

3 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.*

Lösung:

Schritt 1: *Das Auto hält an der Ampel.* - **Eliminierungstest**

Schritt 2: *Das Auto hält an der Straße.* - Substitutionstest

Schritt 3: *Das Auto hält an der rot leuchtenden Ampel und dem gelben Briefkasten.* - **Koordinationstest**

Schritt 4: *An der rot leuchtenden Ampel hält das Auto.* - Permutationstest

Schritt 5: *Das Auto fährt.* - Substitutionstest

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

Kastendiagramm:

Lösung:

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
-----	------	------	----	-----	-----	-------------	-------

Das Auto hält an der rot leuchtenden Ampel

4 Konstituentenstruktur

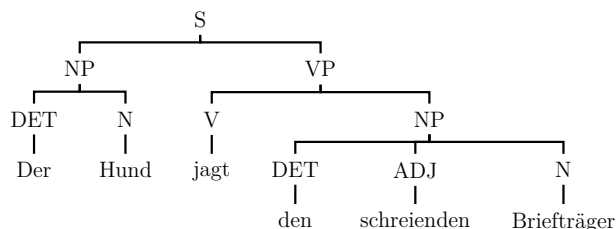
Kontextfreie Grammatiken und Syntaxbäume

- 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.*

Lösung:

1	S	→	NP VP
2	VP	→	V NP
3	NP	→	DET N DET ADJ N
4	DET	→	"Der" "den"
5	N	→	"Hund" "Briefträger"
6	ADJ	→	"schreienden"
7	V	→	"jagt"



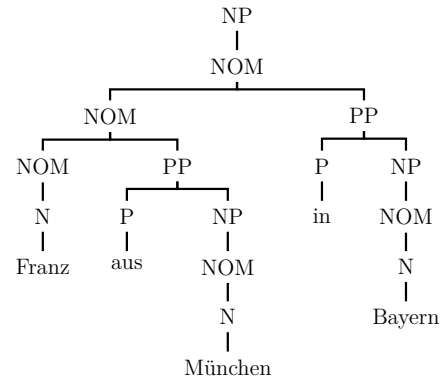
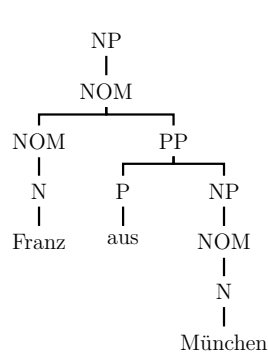
Erweiterung von kontextfreien Grammatiken

- 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).

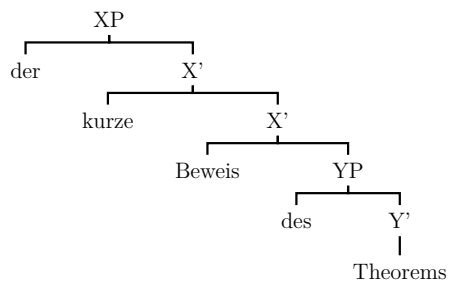
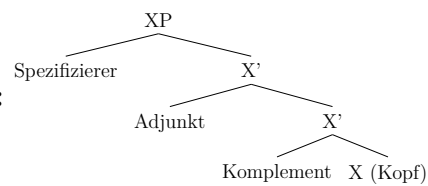
1	PP	→	P NP
2	NP	→	
3	NOM	→	
4	NOM	→	

Lösung:

- | | | | | | |
|---|--|-----|---|-----|----|
| 1 | | PP | → | P | NP |
| 2 | | NP | → | NOM | |
| 3 | | NOM | → | NOM | PP |
| 4 | | NOM | → | N | |

**X-Bar-Konstituentenstruktur**

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

**Vgl.:****Lösung:**

- **Spezifizierer:** der, des
- **Komplemente:** YP
- **Adjunkte:** kurze
- **Köpfe:** Beweis, Theorems

5 Abhängigkeitsstruktur

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?

Lösung:

- **Eliminierungstest:** wenn bei der Eliminierung der Konstituente die Grammatikalität erhalten bleibt, dann handelt es sich um ein Adjunkt
- **Adverbialsatz-Test:** wenn die Konstituente in einen Adverbialsatz ausgelagert werden kann (=grammatisch), dann Adjunkt
- **geschehens-Test:** wenn die Konstituente in einen Satz mit dem Verb *geschehen* ausgelagert werden kann und der Satz grammatisch bleibt, dann 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.*

Lösung:

**Das blaue Krümelmonster verschlingt schmatzend im Park.*

- **Eliminierungstest:** *alle Kekse* ist **Komplement (auch: Ergänzung)**.

Das blaue Krümelmonster verschlingt schmatzend alle Kekse, während es im Park ist.

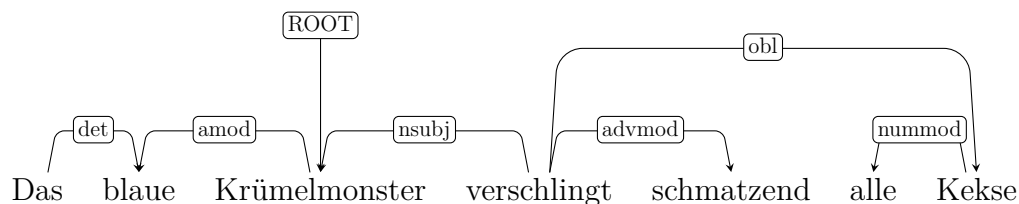
- **Adverbialsatz-Test:** *im Park* ist **Adjunkt (auch: Angabe)**.

Das blaue Krümelmonster verschlingt alle Kekse im Park und das geschieht schmatzend.

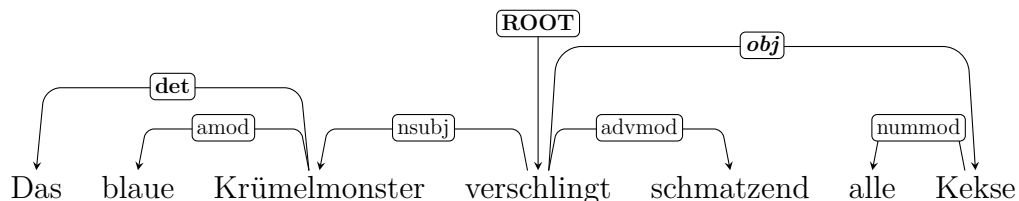
- **geschehens-Test:** *schmatzend* ist **Adjunkt (auch: Angabe)**.

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)



Lösung:

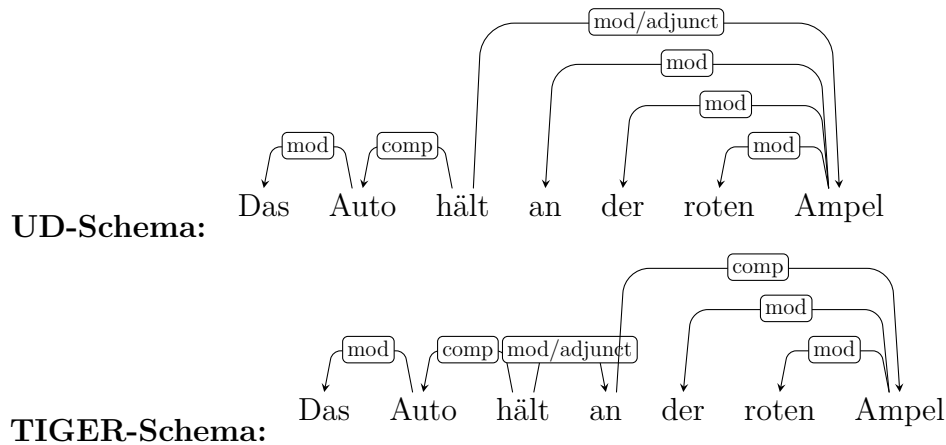


1. der **ROOT-Knoten** (typischerweise das finite Verb) **kontrolliert unmittelbar oder mittelbar alle anderen Knoten** (kann also nicht Dependent sein wie *Krümelmonster* in der Angabe).
2. *Das* ist **det**-Modifikator von *Krümelmonster* (d.h. *Das* kann nicht ohne *Krümelmonster* auftreten); *Das* könnte zwar grundsätzlich auch **det**-Modifikator von *blaue* (als nominalisierte Form) sein; da aber in diesem Satz *blaue* **amod**-Modifikator von *Krümelmonster* ist, ist dies ausgeschlossen, da **kein Wort von mehr als einem Kopf abhängen kann**.
3. *Kekse* sind **Objekt-Komplement** von *verschlingt*; d.h. es ist valenzgefordertes Element (**Auftreten und Form (syntaktischer Typ) vom Kopf gefordert**, hier: Akkusativ; *verschlingt* kann also nicht ohne Akkusativ-Objekt auftreten); *Kekse* kann also nicht **nominaler Dependent des finiten Verbs mit der syntaktischen Funktion Adverbial (= obl)** sein (diese werden im Deutschen üblicherweise auch präpositional kodiert und sind Adjunkte, d.h. verbale Modifikatoren; ein solches Adverbial kann aber durchaus auch Komplement sein, also vom Verb gefordert, vgl. *Er stellt es neben/auf den Tisch (obl)*; allerdings ist hier nur das Auftreten des Adverbials gefordert, nicht dessen Form wie bei Präpositionalobjekten, vgl. *Er wartet auf sie (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.

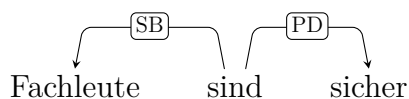
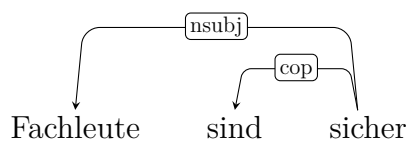
(7) *Das Auto hält an der roten Ampel*

Lösung:

- Komplement = Ergänzung (Auftreten und Form vom Kopf gefordert)
- Verbaler Modifikator = Adjunkt / Angabe
- Nominaler Modifikator = Attribut

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)

**Lösung:**

6 Grammatische Merkmale und Wortstellung

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)

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

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

Lösung:

- Kodierung durch Wortstellung OSV

bird	cat	eat
• O	S	V
cat	bird	eat

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

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

Lösung:

- Kodierung über Kasusmorphologie
- *-fe* = **Objektkasus**

bird-fe	cat	eat
• O	S	V

(vgl.: *den Vogel die Katze frisst*)

bird	cat-fe	eat
• S	O	V

(vgl.: *der Vogel die Katze frisst*)

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

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

Lösung:

- Kodierung über verbale Kongruenz
- das Verb kongruiert mit einer nominalen Kategorie (z.B. Genus)
- das Verb spiegelt die nominalen Marker *-i* und *-o*, es herrscht dann also **Subjektkongruenz** (bei Annahme Kongruenz durch gleiche Form)
- bird-i cat-o eat-o
 O S V
(vgl. Numerus-Kongruenz im Deutschen: *die Vögel die Katze frisst*)
- bird-i cat-o eat-i
 S O V
(vgl.: *die Vögel die Katze fressen*)

Kasusreaktion, Agreement und Subkategorisierung

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

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

(14) *Der Auto fährt schnell.*

Lösung:

Nominale Kongruenz

(15) *Den Auto fährt schnell.*

Lösung:

Kasusreaktion

- (16) *Die Autos fährt schnell.*

Lösung:
Verbale Kongruenz

- (17) *Das Auto fährt, dass es ankommt.*

Lösung:
Subkategorisierung

Wortstellung: Stellungsfeldermodell

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

(18) *Es hat Kuchen gegeben.*

(19) *Hat es Kuchen gegeben?*

- (b) Um welches *Es* handelt es sich in (18)?

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

Lösung:

(a) (20) *Es hat Kuchen gegeben.*

(21) *Hat es Kuchen gegeben?*

- (b)
- Subjekt-Expletiv
 - Begründung: verschwindet in der Frage (V1-Wortstellung, d.h. ohne Vorfeld) nicht, sondern taucht im Mittelfeld auf
 - ist also kein Topik-Es, das bei leerem Vorfeld im V2-Satz diese notwendig zu besetzende Position einnimmt (wie in *Es wird getanzt?* : *Wird getanzt?*)
 - valenzsemantisch ist das Verb nullwertig, d.h. *es* ist hier kein pronominaler NP-Ersatz sondern syntaktisch gefordertes Element, dass die Subjektposition einnimmt

7 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=p1] -> N [NUM=p1]
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:

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

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

(22) **gestern geht Fritz zur Arbeit*

Lösung:

```

1  | S  →  ADV[TIME=?x] VP[TIME=?x]
2  | VP[TIME=?x] → V[TIME=?x] NP PP
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"

```


8 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 \rightarrow COMP S

Lösung:

- S-BAR-Grundregel (eingebettete Sätze)
- UD-Label: meist **mark** für Komplementierer (z. B. für *dass*, *wenn*), z.T. auch **nsubj** usw., etwa beim Relativsatz (*der ...*)

NP \rightarrow NOM, NOM \rightarrow NOM S-BAR

Lösung:

Relativsätze (= Attributsätze): **acl:relcl**

VP \rightarrow V S-BAR

Lösung:

Objektkomplementsatz: **ccomp**

S \rightarrow S-BAR VP

Lösung:

Subjektkomplementsatz: **csbj**

S \rightarrow S-BAR NP VP

Lösung:

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:

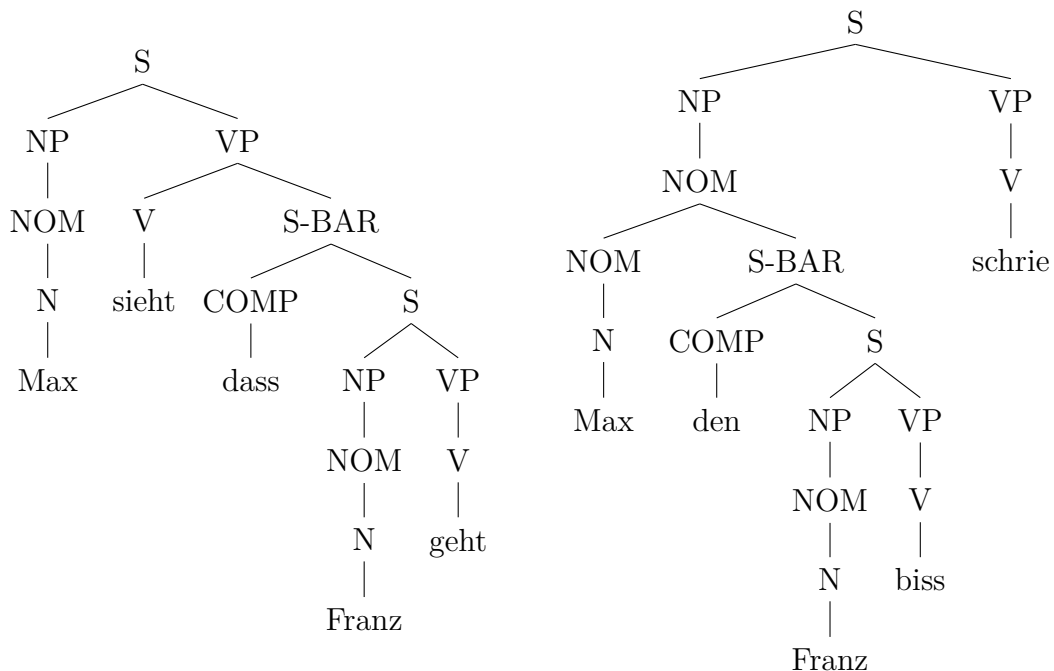
$S \rightarrow NP VP$, $NOM \rightarrow N$, $VP \rightarrow V$

(23) *Max (N) sieht (V) dass (COMP) Franz (N) geht (V)*

(24) *Max (N), den (COMP) Franz (N) biss (V), schrie (V)*

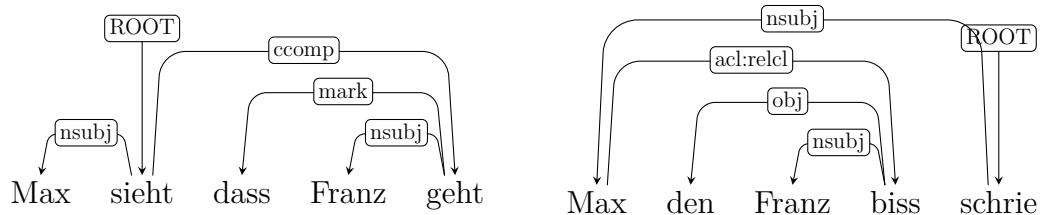
- Verwenden Sie die in Klammern angegebenen lexikalischen Kategorien!

Lösung:



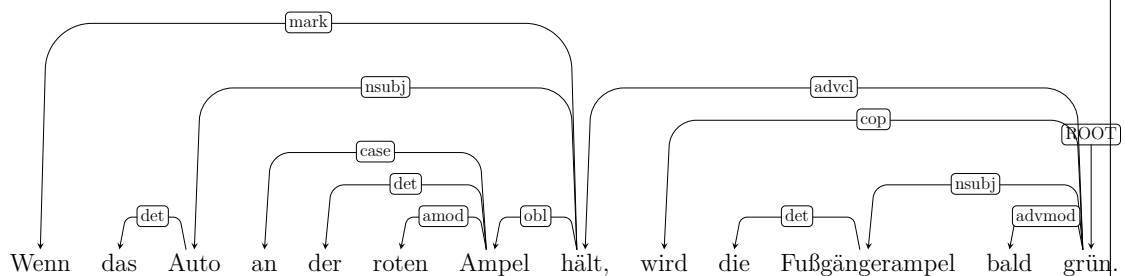
Dependenzstruktur komplexer Sätze 1

- Erstellen Sie zu den Sätzen (23) und (24) 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.

Lösung:**Dependenzstruktur komplexer Sätze 2**

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

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

Lösung:

Stellungsfeldermodell Nebensätze

Analysieren Sie folgenden Satz im Stellungsfeldermodell. Betrachten Sie sowohl den Haupt-, als auch den Nebensatz.

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

Lösung:

- Hauptsatz:
 - Vorfeld: *Wenn das Auto an der roten Ampel hält*
 - linke SK: *wird*
 - Mittelfeld: *die Fußgängerampel bald*
 - rechte SK: *grün*
- Nebensatz:
 - linke SK: *Wenn*
 - Mittelfeld: *das Auto an der roten Ampel*
 - rechte SK: *hält*

Infinite Konstruktionen: Subjekt- vs. Objektkontrolle

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

(27) *Sie überredeten ihn, noch zu bleiben*

Lösung:

- Objektkontrolle
- das Objekt des Matrixsatzes ist referenzidentisch mit dem (nicht explizit versprachlichten) Subjekt der Infinitivkonstruktion, also sinngemäß: *Sie überredeten ihn, dass er noch bleibt.*

9 Parsingalgorithmen

Shift-Reduce-Parser

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:

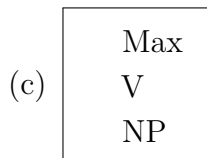
V
NP

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

- (a) Welche Parsing-Strategie verfolgt dieser Parser?
- (b) Welche Operation wird als nächstes ausgeführt?
- (c) Wie sieht der Stack nach Ausführen der Operation aus?

Lösung:

- (a) Bottom-Up
- (b)
 - 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")

**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?
- (b) Welche zwei Operationen stehen diesem Parser zur Verfügung?
- (c) Nennen Sie eine Folge von Regelanwendungen, die zu Backtracking führt.

Lösung:

- (a) Top-Down
- (b) PREDICT und SCAN
- (c) 1-2 oder 1-2-8

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:

$$(\text{DET} \rightarrow \text{der } \cdot, 0) \quad (1)$$

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

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

- (a) Welche Operation wird als nächstes ausgeführt?
- (b) Welcher Zustand wird dabei hinzugefügt?
- (c) Welcher Zustandsmenge wird der Zustand hinzugefügt?

Lösung:

(a) SCAN

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

(c) Q_2

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>