

Beispielaufgaben Klausur Lösung

Aufgabe 1 Fragen zu Grundlagen

Themen: Syntax- und *Grammatikbegriff*, Syntax in der Sprachwissenschaft, Wohlgeformtheit, formale Grammatik als Syntaxmodell, Parsing als Ableitung, Anwendungsgebiete Parsing, Grammatiktypen, Komplexität

- (a) Worauf bezieht sich der Begriff der Grammatik einer natürlichen Sprache in keinem Fall?
- ☐ Sprachstruktur
 - ☐ Theorie der Sprachstruktur
 - ☒ **Bedeutung sprachlicher Zeichen**
 - ☐ Wissen um Sprachstruktur

Aufgabe 2 Fragen zu linguistischen Themen

Themen: Ambiguität, Konstituentenstruktur, X-Bar-Schema, Dependenzstruktur, Valenz und Subkategorisierung, *syntaktische Funktion*, Semantische und pragmatische Rolle, Diathesen, Grammatische Merkmale, Kasusreaktion und Agreement, Wortstellung, *Feldermodell*, Satzarten, Subordination und Koordination, Infinite Konstruktionen

Betrachten Sie die folgenden beiden Sätze:

- (1) *Es hat Kuchen gegeben.*
- (2) *Hat es Kuchen gegeben?*

- (a) Welche der folgenden Aussagen ist falsch?
- ☐ In Satz (1) steht eine Konstituente im Mittelfeld.
 - ☒ **In Satz (2) steht eine Konstituente im Vorfeld.**
 - ☐ In Satz (2) ist das Mittelfeld durch zwei Konstituenten besetzt.
 - ☐ In Satz (1) ist das Vorfeld durch eine Konstituente besetzt.
- (b) Um welches *Es* handelt es sich in (1):
- ☐ Topik-Es (Vorfeld-Platzhalter)
 - ☒ **Subjekt-Expletiv**
 - ☐ Pronominaler Ersatz für Subjekt-NP
 - ☐ Pronominaler Ersatz für Objekt-NP

Aufgabe 3 Formal-linguistische Analyseaufgaben

Themen: *Erstellen von Syntaxbäumen*, Klammerung von Konstituentenstrukturen, Erweiterung von kontextfreien Grammatiken (auch um Merkmale), Wortartanalyse, Konstituententests, Tests zur Komplement/Adjunkt-Unterscheidung

(a) Erstellen Sie den Phrasenstrukturbaum zu Satz (3) gemäß folgender Regeln:

$$S \rightarrow NP VP, \quad VP \rightarrow V, \quad VP \rightarrow V NP, \quad VP \rightarrow V S', \quad S' \rightarrow COMP S$$

(3) *Max (N) sieht (V) wie (COMP) Franz (N) geht (V)*

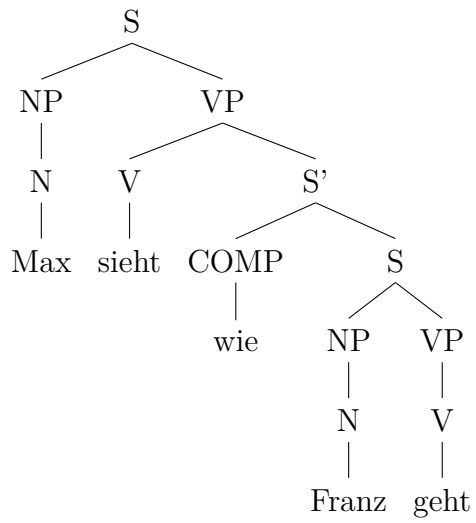
Verwendenden Sie die in Klammern angegebenen lexikalischen Kategorien!

(Bewertung: 1/2 Punkt pro korrekt angewandter syntaktischer Regel)

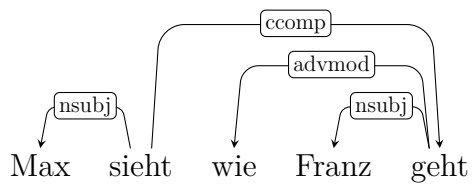
(b) Erstellen Sie zu Satz (3) auch den entsprechenden Dependenzbaum (Kopf von S = VP-Kopf; Kopf von S' = Kopf von S); verwenden Sie folgende Kantenlabel: **nsubj**, **ccomp**, **advmod**.

Lösung:

(a)



(b)



Aufgabe 4 Fragen und Aufgaben zu formalen Themen

Themen: *kontextfreie Grammatiken*, Merkmalsstrukturen (Unifikation/Typhierarchie), *Parsing-Algorithmen*, Statistisches Parsing, Dependency Parsing, Partielles Parsing

- (a) Geben Sie eine rechtsrekursive CFG-Regel an.

Lösung:

- (a) $X \rightarrow Y X$

- (b) Bzgl. welchen Kriteriums stimmen Earley- und CYK-Algorithmus überein?
- ☒ **Algorithmisches Verfahren (Dynamische Programmierung)**
 - ☐ Analyserichtung (top-down)
 - ☐ geforderte Form der Grammatik (Chomsky-Normalform)

Aufgabe 5 Fragen und Aufgaben zur angewandte Syntaxanalyse (NLTK)

Themen: Grammatiken (CFG, DepG, FCFG, *PCFG*), *Merkmalsstrukturen*, *Unifikation*, Modellierung von Subkategorisierung und Agreement, Parsing/Tracing, Grammar Induction, Datengestützte Methoden/Chunking

- (a) Was ist im Folgenden das Output von `print(fs3.unify(fs2))` (= das Ergebnis der Unifikation von fs2 mit fs3)?

```
1 fs1 = FeatStruct(number='singular', person=3)
2 fs2 = FeatStruct(agr=fs1)
3 fs3 = FeatStruct(agr=FeatStruct(person=1))
4 print(fs3.unify(fs2))
```

- ☐ identisch mit `print(fs1)`
 - ☐ identisch mit `print(fs2)`
 - ☐ identisch mit `print(fs3)`
 - ☒ **None**
- (b) Um was für eine Grammatik handelt es sich im Folgenden (** = unkenntlich gemacht)?

```
1 grammar = nltk.*****.fromstring("""
2     S      -> NP VP          [1.0]
3     VP     -> TV NP          [0.4]
4     VP     -> IV             [0.3]
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)
```

(c) Wie errechnet sich die Wahrscheinlichkeit in Zeile 15?

Lösung:

(a) PCFG = Probabilistische kontextfreie Grammatik

(b) $1,0 * 0,2 * 0,4 * 1,0 * 0,8 = 0,064$