

Frohe Weihnachten

Syntax natürlicher Sprachen

Tutorium 09:

Feature-Based Grammatik

Shuyan Liu

20.12.2024

Einige Beispiele kommt aus der Vorlesungsfolien, Aufgaben sowie Übungen.

Die Hauptteile der Slides dieser Woche stammen von Sarah Anna Uffelmann aus Wintersemester 2023/24 und wurden bearbeitet. Verwendung mit Dank.

Subkategorisierung

- handelt sich um syntaktische Valenz eines Wortes

1 $_ V _ \rightarrow _ \text{Vintrans} _$

2 $_ V NP \rightarrow _ \text{Vtrans} NP$

3 $_ V PP \rightarrow _ \text{Vprepobj} PP$ meist Verben mit **festen** Präpositionen

4 $_ V NP NP \rightarrow _ \text{Vditrans} NP NP$

5 $_ V NP PP \rightarrow _ \text{Vplace} NP PP$ Ich lege den Apfel in die Kiste

6 $_ V S\text{-}BAR \rightarrow _ \text{Vclause} S\text{-}BAR$ I know [that he has already left]

7

8 $VP \rightarrow _ V _ \mid _ V NP \mid \dots$

Constraint-Regeln

```
gramstring = r"""
% start S
  S  -> NP VP

  VP -> V[OBJCASE=?y] NP[CASE=?y]
  VP -> V

  NP[CASE=?x] -> DET[CASE=?x] N

  DET[CASE=nom] -> "der"
  DET[CASE=akk] -> "den"
  DET[CASE=dat] -> "dem"
  N  -> "Hund" | "Briefträger"
  V  -> "schläft"
  V[OBJCASE=akk] -> "jagt"
  V[OBJCASE=dat] -> "gehört"
"""
```

OBJCASE von V stimmt mit CASE von NP überein,
damit V NP eine gültige VP formt

Beispiel von Constraint-Regeln in NLTK

Vererbung: CASE der NP wird vom CASE des DET vererbt

Subkategorisierung

Koreferenz

Zwei Merkmale sind **koreferent**, wenn sie denselben Wert (bzw. bei komplexen Merkmalen dieselbe Merkmalsstruktur) teilen.

```
gramstring = r"""
% start S
  S  -> NP VP

  VP -> V[OBJCASE=?y] NP[CASE=?y]
  VP -> V

  NP[CASE=?x] -> DET[CASE=?x] N

  DET[CASE=nom] -> "der"
  DET[CASE=akk] -> "den"
  DET[CASE=dat] -> "dem"
  N  -> "Hund" | "Briefträger"
  V  -> "schläft"
  V[OBJCASE=akk] -> "jagt"
  V[OBJCASE=dat] -> "gehört"
"""
```

Z.B. fordern wir in dieser Grammatik bei der Regel

VP -> V[OBJCASE=?y] NP[CASE=?y]

dass, die Merkmale OBJCASE und CASE koreferent sind.

Constraintregeln

<i>bellt</i>	CAT	V	
		TEMP	PRES
	AGR	NUM	SG
		PER	3
	SUBCAT	NONE	

intransitives Verb:
SUBCAT: NONE (ohne Komplement)

<i>jagt</i>	CAT	V	
		TEMP	PRES
	AGR	NUM	SG
		PER	3
	SUBCAT	CAT	NP
		CASE	ACC

transitives Verb:
SUBCAT: Akkusative NP

Constraintregeln: Unifikation



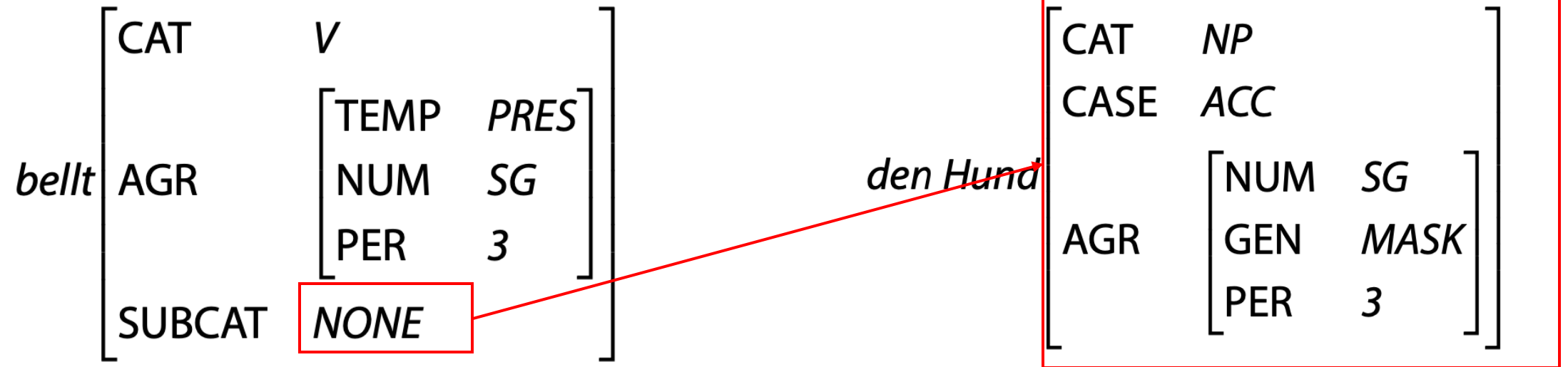
intransitives Verb:

SUBCAT: NONE (ohne Komplement)

Unifikation zwischen SUBCAT [CAT: NP, CASE: ACC] und [CAT: NP, CASE: ACC, AGR = [...]] **erfolgt**

Richtige Kombination = Erfolgreiche Unifikation

Constraintregeln: Unifikation



intransitives Verb:

SUBCAT: NONE (ohne Komplement)

Unifikation zwischen NONE und [CAT: NP, CASE: ACC, AGR = [...]] schlägt fehl

Achtung: **NONE** != []

[] unifiziert mit allem, da es kein Widerspruch gibt

Kongruenz

Wenn CASE und AGR in DET und N unifizieren, dann formt DET N eine NP

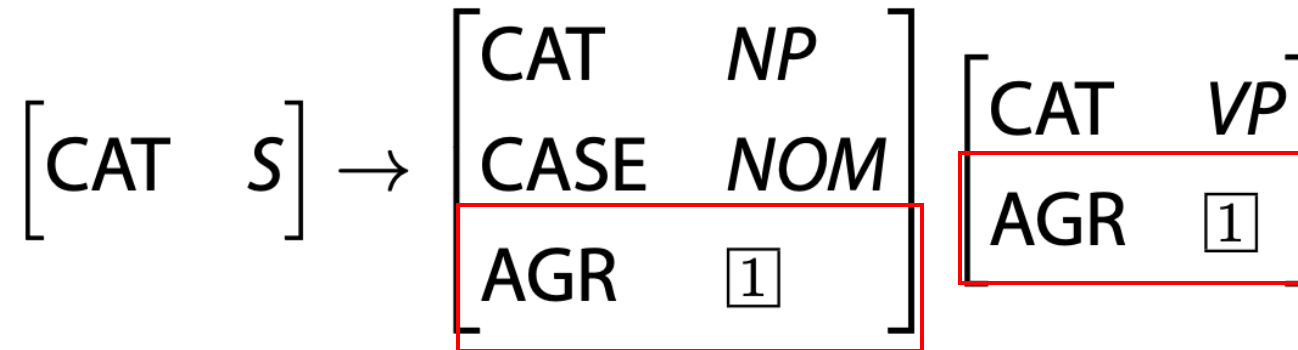
CAT	<i>DET</i>	CAT	<i>N</i>
CASE	2	CASE	2
AGR	1	AGR	1

[
CAT: NP
CASE: [2]
AGR: [1]
]

Artikel-Nomen-Kongruenz: Kasus [Genus, Numerus, ...]

Kongruenz

vermeidet Überproduktion!



Dieser Regel zufolge kann S nur dann zu NP VP expandieren, wenn die NP Nominativ steht (**CASE NOM**).

Zusätzlich müssen NP und VP hinsichtlich ihrer Merkmalsausprägungen von **AGR koreferent** sein.

(Die 1 ist hier eine Variable, die für eine bestimmte Merkmalsausprägung steht.)

Subjekt-Verb-Kongruenz

CAT	NP					CAT	VP				
CASE	NOM										
AGR		NUM	SG			AGR		TEMP	PRES		
		GEN	MASK					NUM	SG		
		PER	3					PER	3		
						SUBCAT	NONE				

-> AGR **unifiziert**

CAT	NP					CAT	VP				
		NUM	SG					TEMP	PRES		
AGR		GEN	FEM			AGR		NUM	PL		
		PER	3					PER	3		
						SUBCAT	NONE				

-> AGR **unifiziert nicht**

(Widerspruch im Numerus)

Boolsche Merkmale: Invertierte Wortstellung

Aussagesatz: “You like chocolate.”

Fragesatz: “Do you like chocolate?” -> Hilfsverb an erster Position

Die invertierte Wortstellung lässt sich mittels boolscher Merkmale modellieren:

Inversionsmerkmal: [+/-Inv] **[+Inv] = [Inv=True], [-Inv] = [Inv=False]**

Auxiliarmerkmal: [+/-Aux]

Entsprechend können wir die CFG-Regeln für invertierte Sätze so formulieren:

S[+Inv] -> V[+Aux] NP VP[-Aux]

VP[-Aux] -> V[-Aux] NP

-> Bei einem Satz mit invertierter Wortstellung muss an erster Position ein Hilfsverb stehen, gefolgt von einer NP und einer VP, wobei das Verb der VP kein Hilfsverb sein darf.

Boolsche Merkmale: Hilfsverbkonstruktionen

Das Auxiliarmedial können wir auch verwenden, um Hilfsverbkonstruktionen im Deutschen zu modellieren:

Präsens: „Das Eichhörnchen **vergräbt** die Nuss.“

Perfekt: „Das Eichhörnchen **hat** die Nuss **vergraben**.“

Um Übergenerierung zu vermeiden, brauchen wir zusätzlich ein boolsches Merkmal für das Partizip Perfekt: **[+/-PP]**

Boolsche Merkmale: Hilfsverbkonstruktionen

S → NP VP

VP[-INV] → VERBAL

VERBAL → V[-PP] NP

VP[+INV] → V[+AUX] VERBAL

VERBAL → NP V[-AUX, +PP]

NP → DET NOM

NOM → N

DET → "das" | "die"

N → "Eichhörnchen" | "Nuss"

V[-AUX, -PP] → "vergräbt"

V[+AUX] → "hat"

V[-AUX, +PP] → "vergraben"

Wir verwenden dafür das **XBAR-Schema**.

Spezifiziererregeln:

VP[-INV] → VERBAL (für Präsens)

VP[+INV] → V[+AUX] VERBAL (für Perfekt)

Komplementregeln:

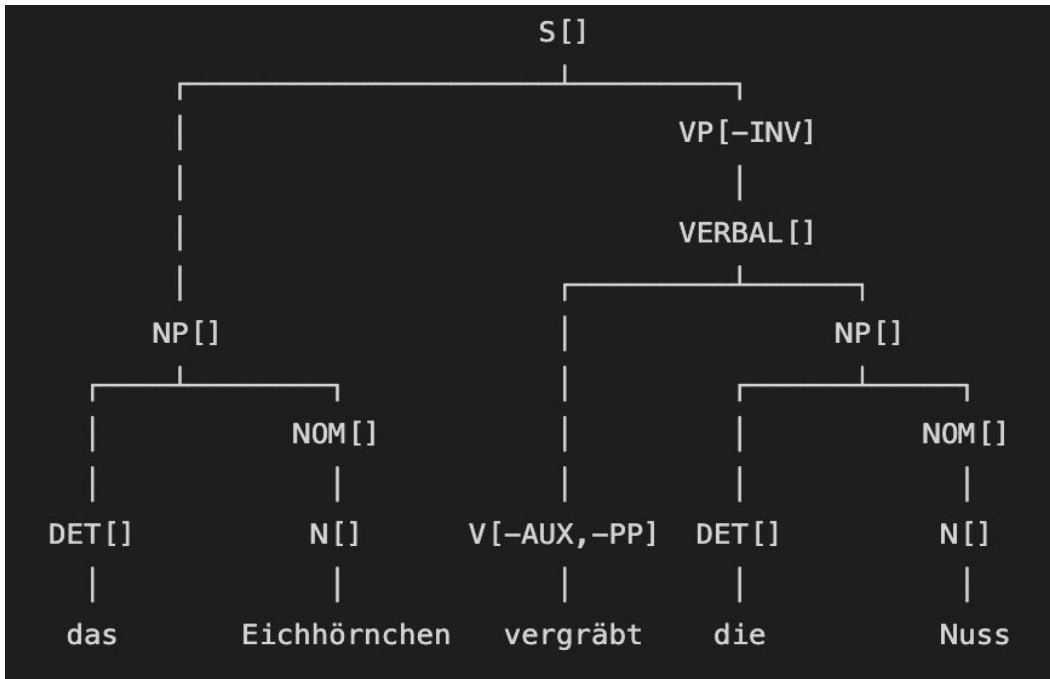
VERBAL → V[-PP] NP (für Präsens)

VERBAL → NP V[-AUX, +PP] (für Perfekt)

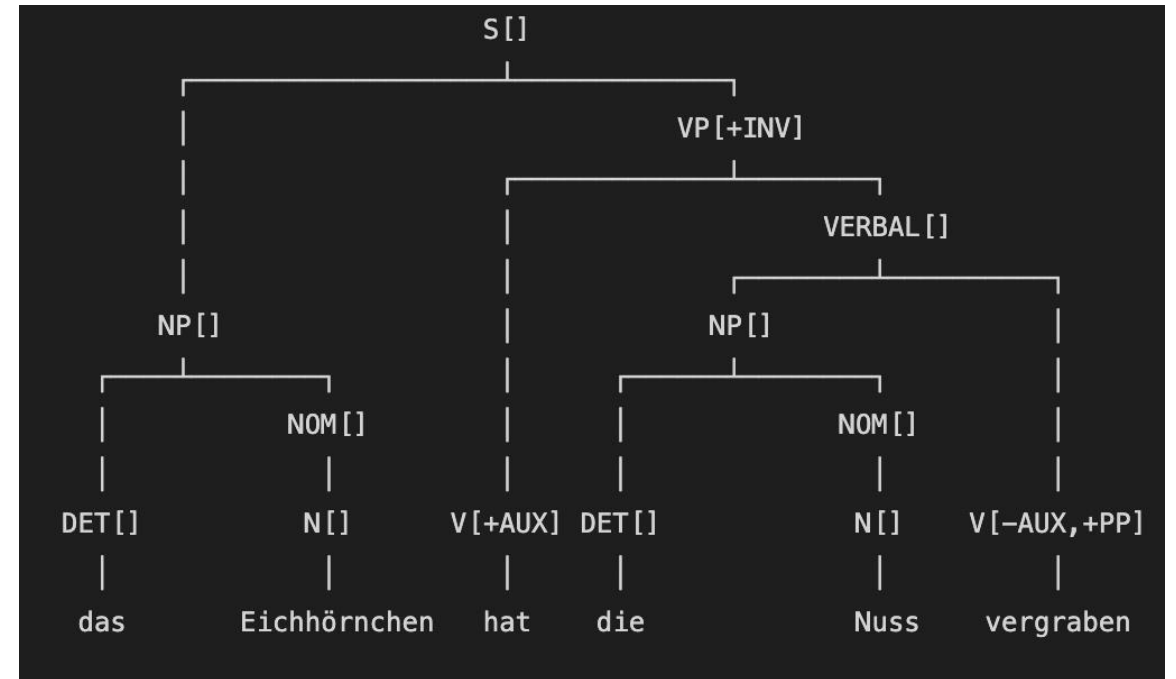
(Adjunktregeln brauchen wir hierfür nicht.)

Partizip Perfekt: **[+/-PP]**

Boolsche Merkmale: Hilfsverbkonstruktionen



Präsens
(nicht invertiert)



Perfekt
(invertiert)

Gap-Feature

- Gap-Feature beschreibt die Existenz von Lücken oder *missing positions* innerhalb eines Satzes. Diese Lücken entstehen meist aufgrund von Bewegung oder Extraktion (z. B. bei WH-Bewegungen oder relativen Konstruktionen).
- WH-Bewegung
 - Was hast du ____ gesehen?
 - Das Fragewort „was“ wurde verschoben, und das Gap (____) verweist auf die Position, an der ursprünglich ein Objekt stand.
- Extraktion
 - Das Buch, das ich ____ gekauft habe.
 - Das Relativpronomen "das" füllt die Lücke aus, die durch die Bewegung des Objekts verursacht wurde (____ steht für die Position von "das").

Gap-Feature

- Das Gap-Feature wird oft durch Merkmale wie [+/-WH] ausgedrückt:
 - +WH zeigt an, dass ein Satz oder Knoten ein WH-Element enthält, das eine Lücke (Gap) verursacht hat, die aufgelöst oder mit einer passenden Ergänzung verbunden werden muss.
 - -WH: - bedeutet, dass keine Lücke enthalten ist.

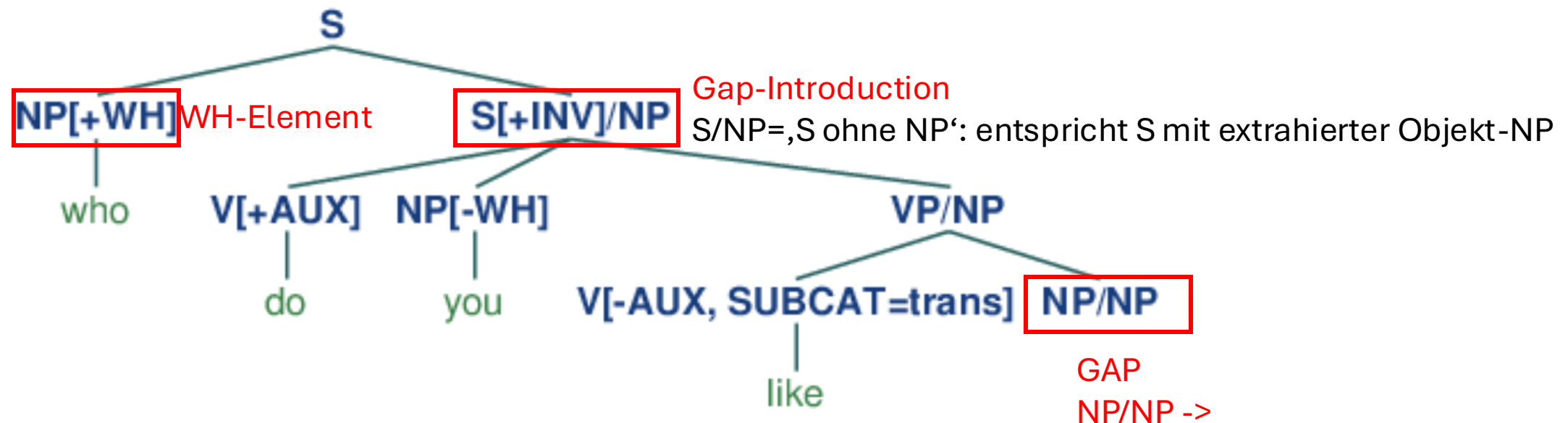


Abbildung: (http://www.nltk.org/book/tree_images/ch09-tree-16.png)

Gap-Features mit Slash-Kategorien

Wir wollen eine Grammatik schreiben, aus der wir die Sätze „**Sie hat ihn gesehen**“ und „**Wen hat sie gesehen?**“ ableiten können.

Dazu führen wir zwei boolsche Merkmale ein:

[+/-QUEST] für das Fragepronomen

[+/-MOVEMENT] für den Ergänzungsfragesatz

```
#Movement Objekt (Gap-Introduction):
|   S[+MOVEMENT] -> NP[+QUEST] S[+INV]/NP

#Gap-Informationen herunterreichen:
|   S[+INV]/?x -> V[+AUX] NP VP/?x
|   VP/?x -> NP/?x V[+PP]

#Gap-Realisierung:
|   NP/NP ->
```

Wir legen fest,

- dass im Ergänzungsfragesatz die NP ein Fragepronomen sein muss
- dass nach dem Fragepronomen ein Satz mit invertierter Wortfolge ohne NP stehen muss

(Statt der Variable ?x könnten wir auch /NP weiter herunterreichen.)

Gap-Features mit Slash-Kategorien

```
S[-INV] -> NP[-QUEST] VP[+INV]
VP[+INV] -> V[+AUX] NP[-QUEST] V[+PP]
NP[QUEST=?x] -> PRON[QUEST=?x]
```

```
S[+INV] -> V[+AUX] NP VP
VP -> NP V[+PP]
```

#Movement Objekt (Gap-Introduction):

```
S[+MOVEMENT] -> NP[+QUEST] S[+INV]/NP
```

#Gap-Informationen herunterreichen:

```
S[+INV]/?x -> V[+AUX] NP VP/?x
```

```
VP/?x -> NP/?x V[+PP]
```

#Gap-Realisierung:

```
NP/NP ->
```

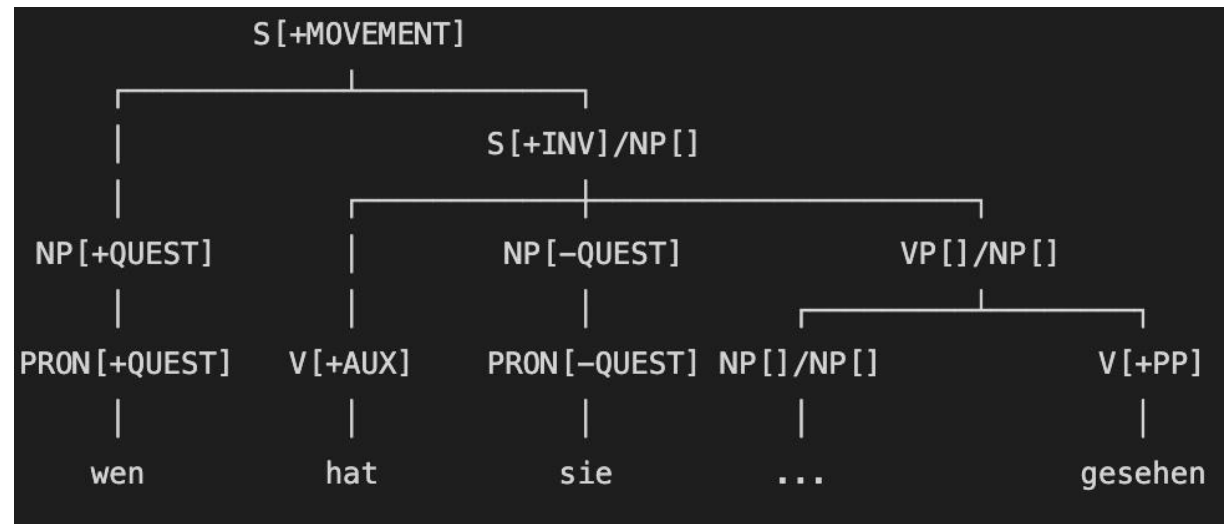
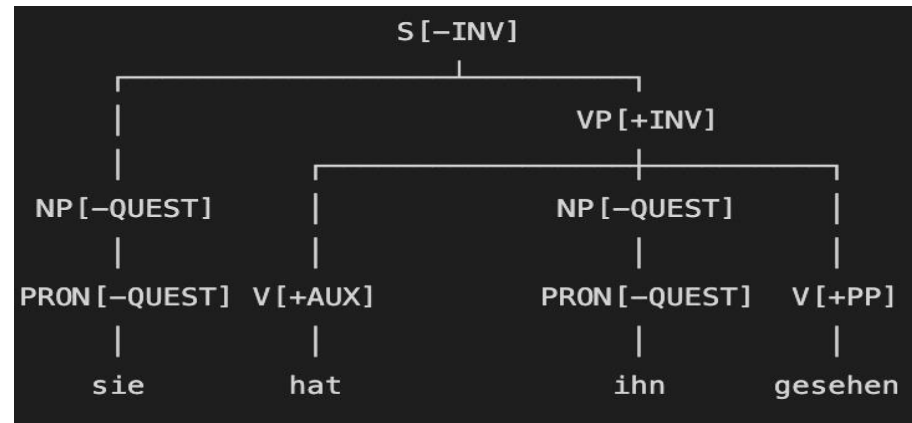
```
PRON[-QUEST] -> "sie"
```

```
PRON[-QUEST] -> "ihn"
```

```
PRON[+QUEST] -> "wen"
```

```
V[+AUX] -> "hat"
```

```
V[+PP] -> "gesehen"
```



(Statt der Variable ?x könnten wir auch /NP weiter herunterreichen.)