

Syntax natürlicher Sprachen

Tutorium

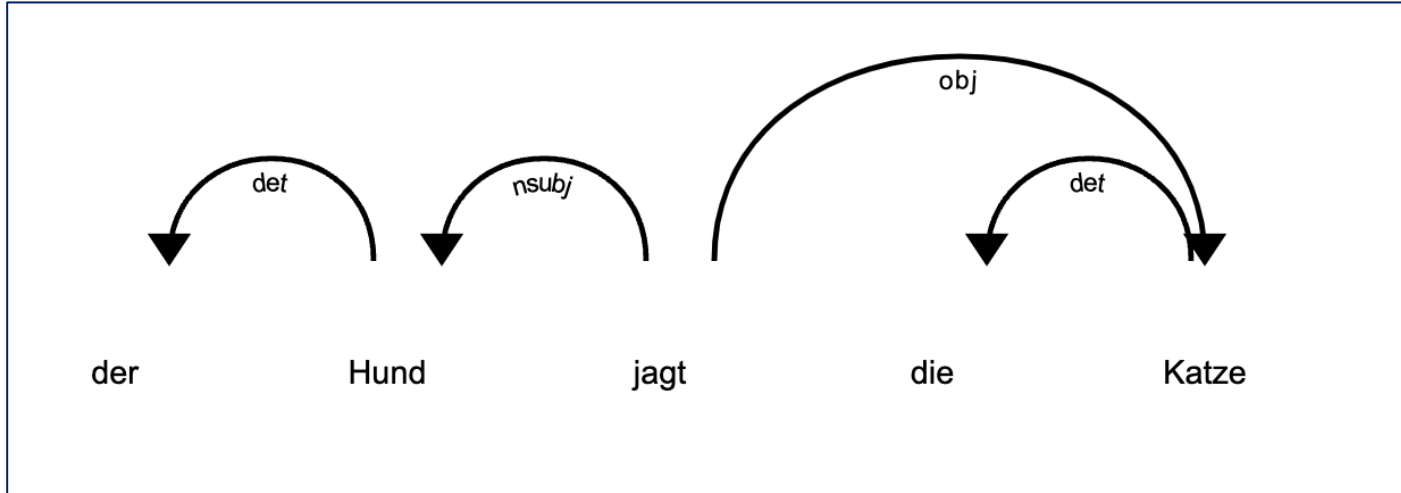
Grammatische Merkmale

Sarah Anna Uffelmann

15.12.2023

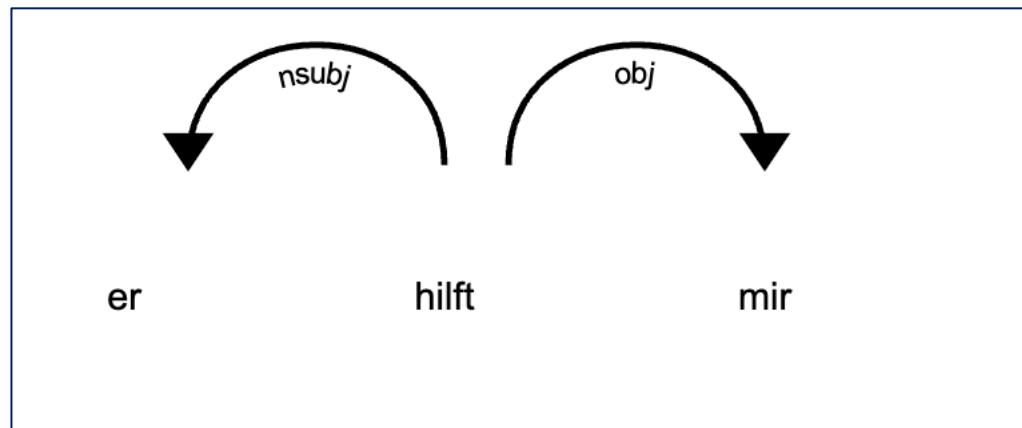
Nachtrag zu Dependenz:
iobj, obj, cop, aux

obj



Akkusativobjekt

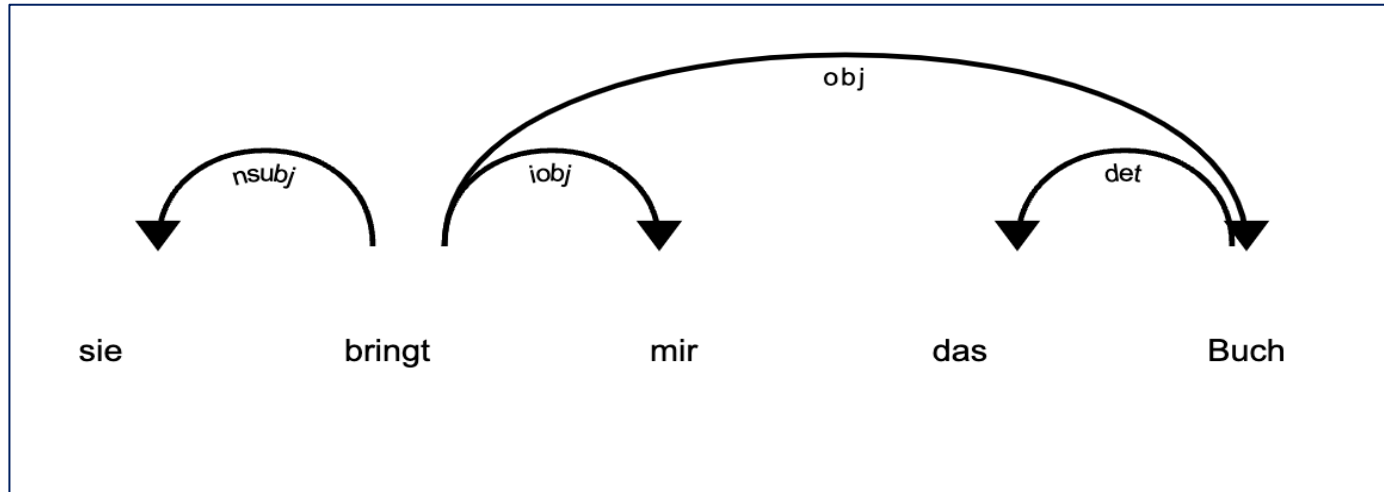
```
sent_nr = ""  
1 der 2 det  
2 Hund 3 nsubj  
3 jagt 0 ROOT  
4 die 5 det  
5 Katze 3 obj  
""
```



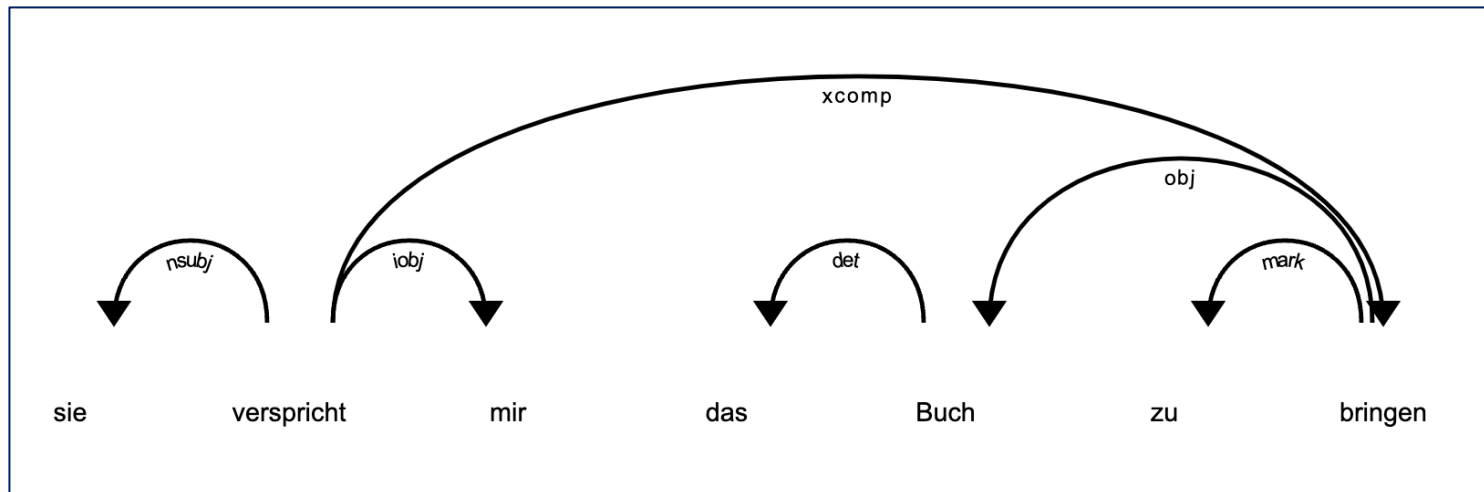
Dativobjekt

```
sent_nr = ""  
1 er 2 nsubj  
2 hilft 0 ROOT  
3 mir 2 obj  
""
```

obj und iobj



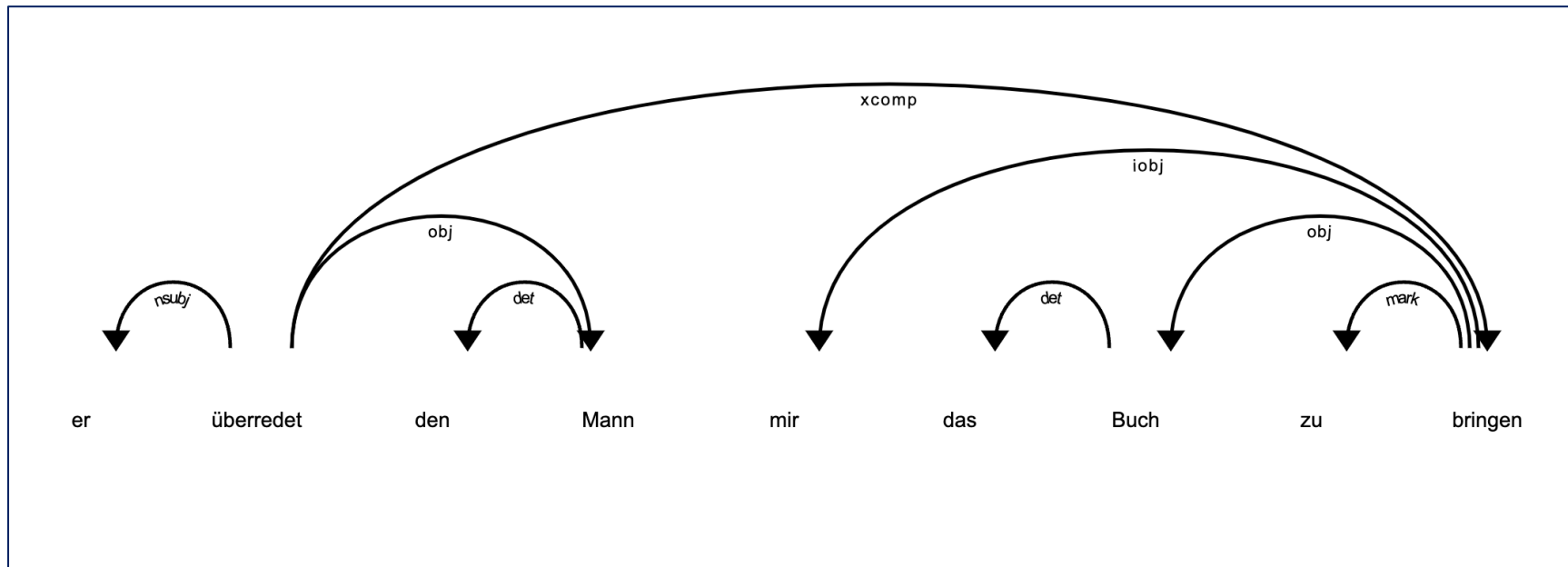
```
sent_nr = ""  
1 sie 2 nsubj  
2 bringt 0 ROOT  
3 mir 2 iobj  
4 das 5 det  
5 Buch 2 obj  
""
```



```
sent_nr = ""  
1 sie 2 nsubj  
2 verspricht 0 ROOT  
3 mir 2 iobj  
4 das 5 det  
5 Buch 7 obj  
6 zu 7 mark  
7 bringen 2 xcomp  
""
```

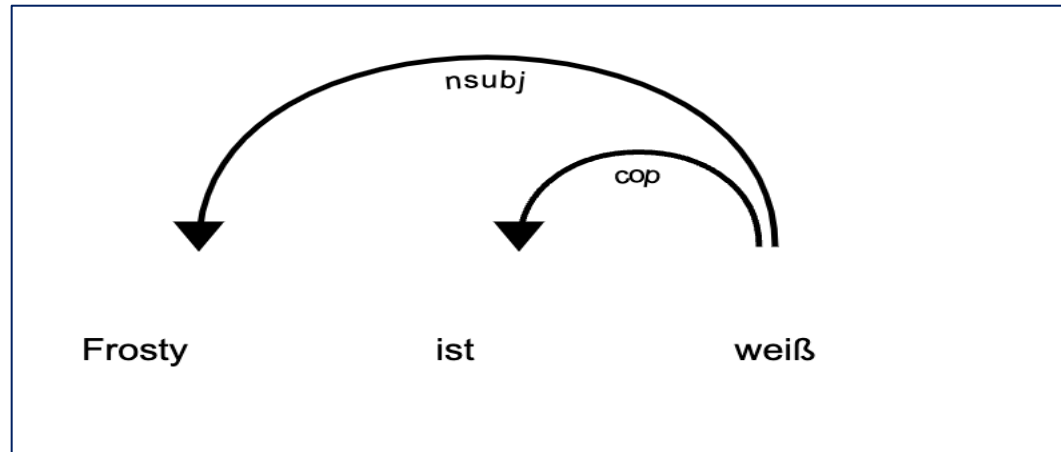
obj und iobj

Es kann aber auch obj und xcomp (bzw. ccomp) in einem Satz geben!
(Hier zusätzlich mit obj und iobj auch im eingebetteten Satz.)



```
sent_nr = ""  
1 er 2 nsubj  
2 überredet 0 ROOT  
3 den 4 det  
4 Mann 2 obj  
5 mir 9 iobj  
6 das 7 det  
7 Buch 9 obj  
8 zu 9 mark  
9 bringen 2 xcomp  
""
```

cop

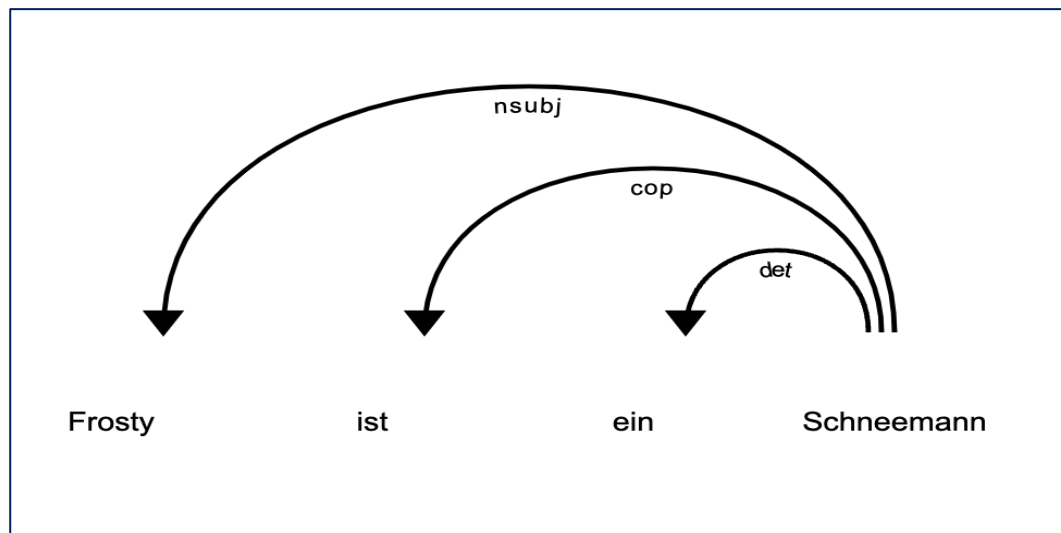


```
sent_nr = ""  
1 Frosty 3 nsubj  
2 ist 3 cop  
3 weiß 0 ROOT
```

""

Prädikative
Zuschreibung:

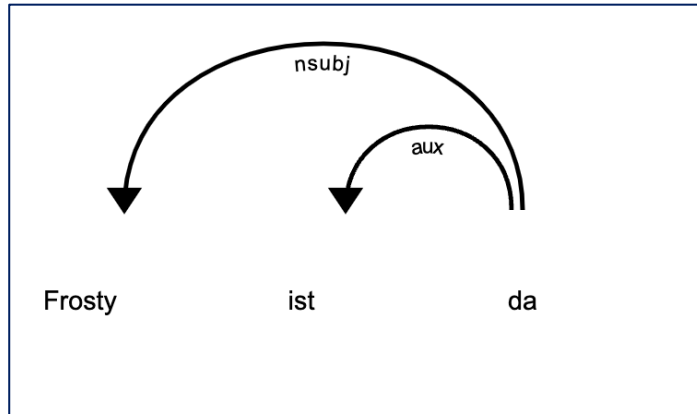
X ist Y



```
sent_nr = ""  
1 Frosty 4 nsubj  
2 ist 4 cop  
3 ein 4 det  
4 Schneemann 0 ROOT
```

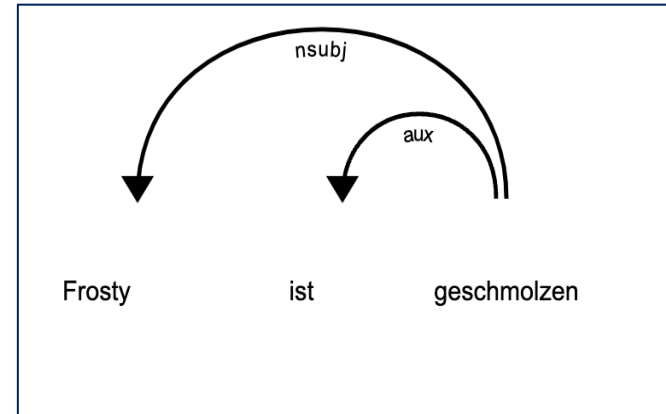
""

aux



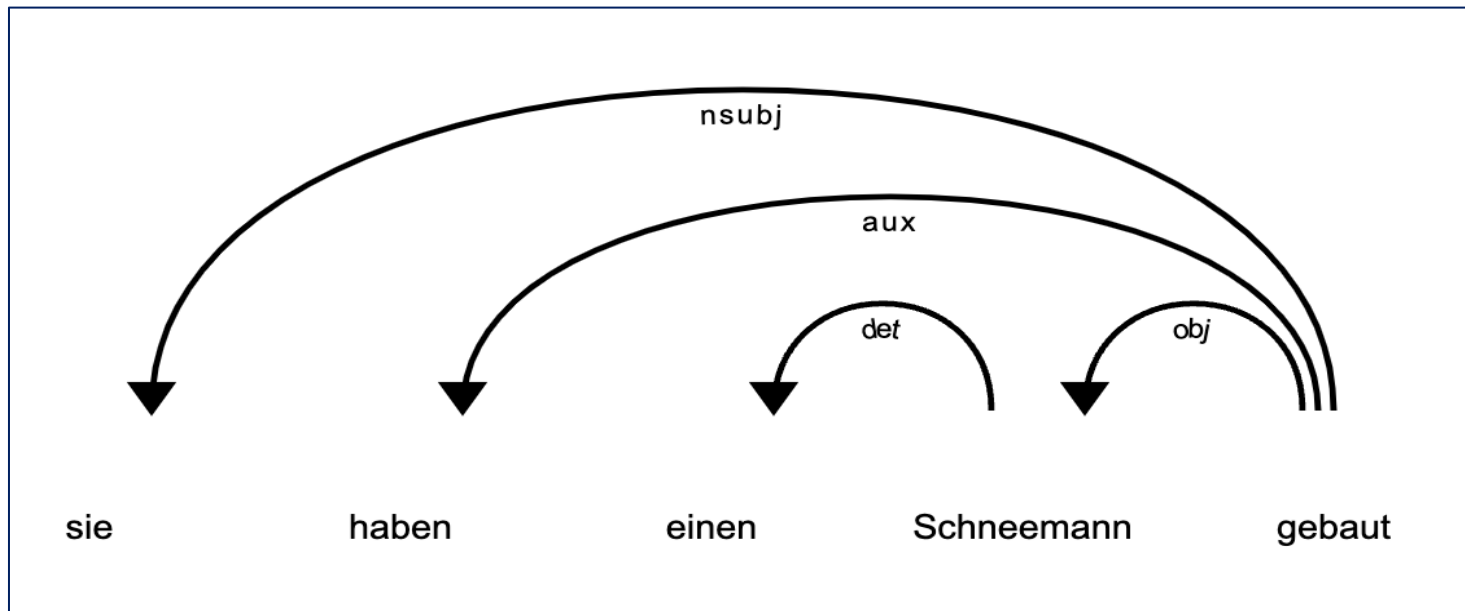
```
sent_nr = ""  
1 Frosty 3 nsubj  
2 ist 3 aux  
3 da 3 ROOT  
""
```

Verb: dasein



```
sent_nr = ""  
1 Frosty 3 nsubj  
2 ist 3 aux  
3 geschmolzen 0 ROOT  
""
```

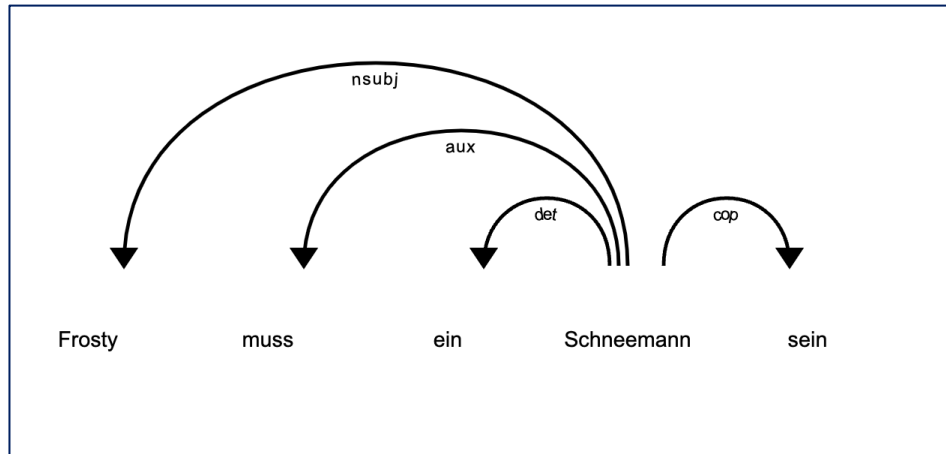
Perfekt von ‚schmelzen‘



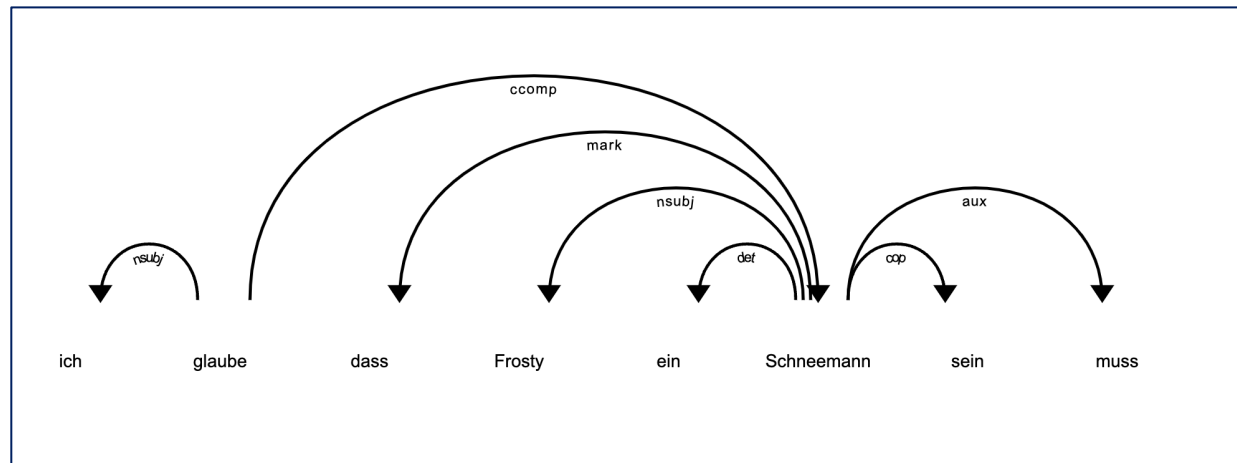
```
sent_nr = ""  
1 sie 5 nsubj  
2 haben 5 aux  
3 einen 4 det  
4 Schneemann 5 obj  
5 gebaut 0 ROOT  
""
```

Perfekt von ‚bauen‘

cop und aux in einem Satz



```
sent_nr = ""  
1 Frosty 4 nsubj  
2 muss 4 aux  
3 ein 4 det  
4 Schneemann 0 ROOT  
5 sein 4 cop  
""
```



```
sent_nr = ""  
1 ich 2 nsubj  
2 glaube 0 ROOT  
3 dass 6 mark  
4 Frosty 6 nsubj  
5 ein 6 det  
6 Schneemann 2 ccomp  
7 sein 6 cop  
8 muss 6 aux  
""
```


Grammatische Merkmale

Warum brauchen wir Feature-based Grammars?

Angenommen, wir haben folgende Grammatik:

$S \rightarrow NP VP$

$NP \rightarrow Det N$

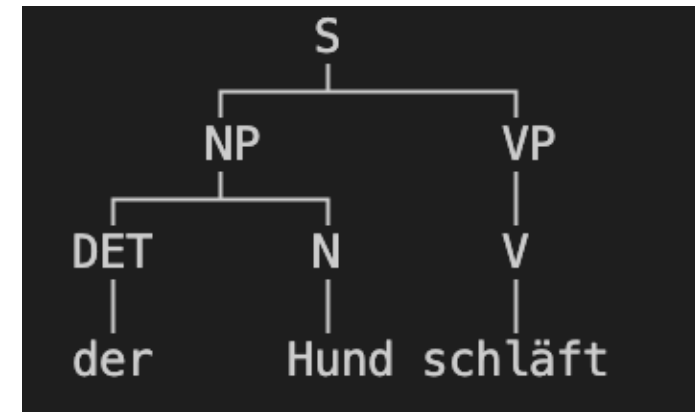
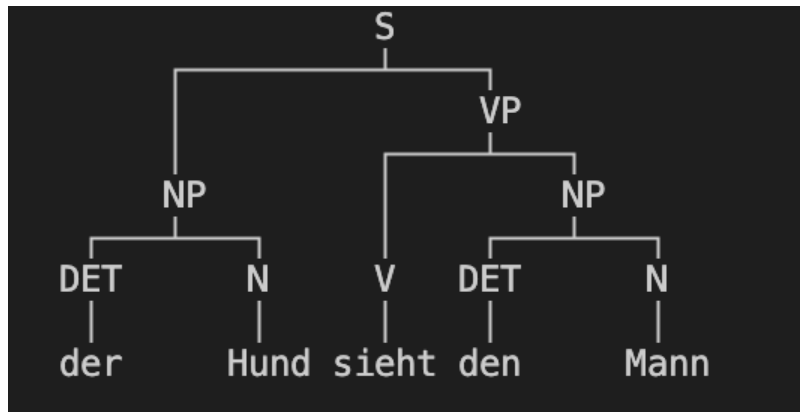
$VP \rightarrow V \mid V NP$

$Det \rightarrow 'der' \mid 'den'$

$N \rightarrow 'Hund' \mid 'Mann'$

$V \rightarrow 'schläft' \mid 'sieht'$

Damit können wir z.B. diese beiden Sätze erfolgreich analysieren:

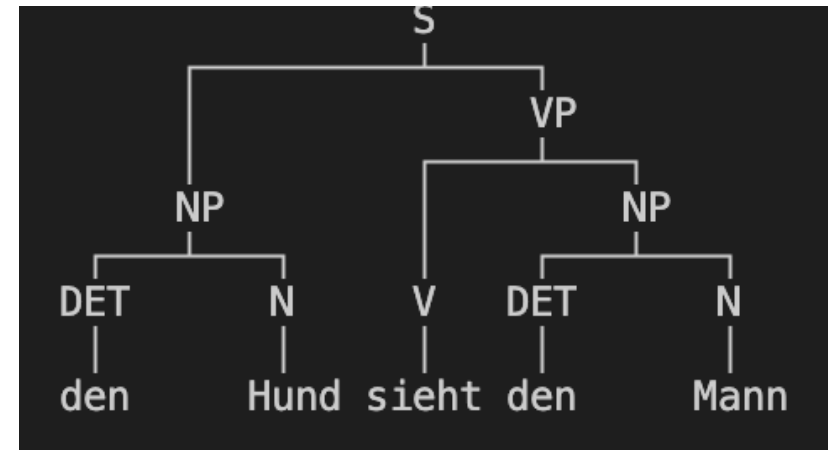
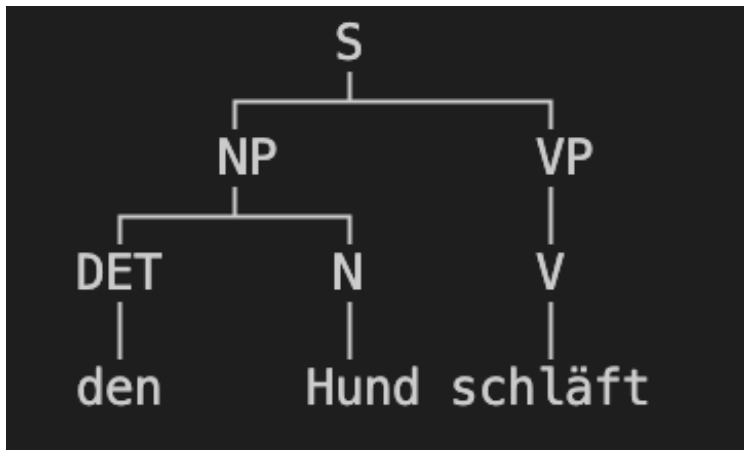


Warum brauchen wir Feature-based Grammars?

Allerdings erlaubt diese Grammatik auch die Analyse von nicht-wohlgeformten Sätzen:

S → NP VP
NP → Det N
VP → V | V NP

Det → 'der' | 'den'
N → 'Hund' | 'Mann'
V → 'schläft' | 'sieht'



-> Problem der Übergenerierung!

Feature-based Grammars

Wir lösen dieses Problem, indem wir morphosyntaktische Merkmalsstrukturen (Features) modellieren.

-> **Feature-based Grammar = FCFG** (Feature-based context free grammar)

Wir müssen unsere **Regeln** so **einschränken**, dass ungrammatische Sätze nicht mehr abgeleitet werden können. Dabei müssen wir vor allem die folgenden drei zentralen **morphosyntaktischen Constraints** berücksichtigen:

- Kasusreaktion
- Kongruenz
- Subkategorisierung

Morphosyntaktische Constraints

Kasusreaktion

Bestimmte Wörter erfordern (regieren) einen bestimmten Fall.

Beispiele:

- "warten auf" + Akkusativ
- "folgen" + Dativ
- „an etwas glauben“ + Akkusativ
- „gedenken“ + Genitiv
- „mit“ + Dativ
- „für“ + Akkusativ

Morphosyntaktische Constraints

Kongruenz

Übereinstimmung von Wörtern oder Satzteilen hinsichtlich grammatischer Merkmale

Beispiele:

„**Ein** schöner grüner **Baum** steht im Wald.“ (Nom. Sg. mask. --- 3. Pers. Sg.)

„**Der** schöne grüne **Baum** ist eine Tanne.“ (Nom. Sg. mask. --- 3. Pers. Sg.)

„Im Wald stehen viele große **Tannen**.“ (Nom. Pl. fem. --- 3. Pers. Pl.)

„Siehst **Du** den dunklen **Wald**?“ (Akk. Sg. mask. --- 2. Pers. Sg.)

-> Übereinstimmung von Subjekt und Verb in Person und Numerus

-> Übereinstimmung von Kasus, Numerus und Genus in NPs

Morphosyntaktische Constraints

Subkategorisierung

Art und Anzahl der Argumente (Valenz)

Beispiele:

- **intransitive Verben**: Subjekt im Nominativ
wachsen -> „Der Baum wächst.“
- **transitive Verben**: Subjekt im Nominativ + Objekt (meistens im Akk., aber nicht immer)
beantworten -> „Meine Freundin beantwortet den Brief.“ (Akkusativobjekt)
gehören -> „Der Hund gehört ihm.“ (Dativobjekt)
- **ditransitive Verben**: Subjekt im Nominativ + Akkusativobjekt + Dativobjekt
bringen -> „Der Postbote bringt ihr ein Paket.“

Flexion

Deklination = nominale Flexion (Nomen, Artikel, Adjektive, Pronomen)

Nominale Flexionskategorien des Deutschen:

Kasus, Numerus, Genus, Person, Definitheit

Konjugation = verbale Flexion (Verben)

Verbale Flexionskategorien des Deutschen:

Person, Numerus, Tempus, Modus, Genus Verbi

Deklination (im Deutschen)

Kasus	Nominativ, Genitiv, Akkusativ, Dativ
Numerus	Singular, Plural
Genus	Maskulinum, Femininum, Neutrum
Person	1, 2, 3 (bei Pronomen, Substantive sind immer 3. Person)
Definitheit	Definit, Indefinit (z.B. bestimmter vs. unbestimmter Artikel)

Konjugation (im Deutschen)

Person	1, 2, 3
--------	---------

Numerus	Singular, Plural
---------	------------------

Tempus	Präsens, Präteritum, Perfekt, Plusquamperfekt, Futur I, Futur II
--------	------------------------------------------------------------------

Modus	Indikativ, Konjunktiv, Imperativ
-------	----------------------------------

Genus Verbi	Aktiv, Passiv
-------------	---------------

Grammatische Merkmale

Wir unterscheiden zwischen (grammatischen) **Merkmalen** und **Merkmalsausprägungen** (= **Werten**)

Beispiele:

Grammatisches Merkmal	Werte (Merkmalsausprägungen)
Numerus	Singular, Plural
Person	1, 2, 3
Kasus	Nominativ, Genitiv, Akkusativ, Dativ

Merkmalskongruenz = Agreement

Agreement oder Merkmalskongruenz heißt, dass bestimmte Satzteile hinsichtlich ihrer Merkmale übereinstimmen.

Im Deutschen müssen z.B. Subjekt und Verb eines Satzes in Person und Numerus übereinstimmen (= sie müssen kongruent sein):

„Der Baum steht im Wald.“

* „Der Baum stehen im Wald.“

Merkmalsstrukturen

Merkmalsstrukturen (= Feature Structures) sind Merkmal-Wert-Paare:

$$\text{Merkmalsstruktur} = \begin{bmatrix} \text{MERKMAL1} & \text{WERT1} \\ \text{MERKMAL2} & \text{WERT2} \end{bmatrix}$$

Grammatische Merkmale lassen sich als Merkmalsstruktur darstellen:

$$N \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix} \quad \text{oder} \quad \begin{bmatrix} \text{CAT} & N \\ \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix}$$

Merkmalsstrukturen

Merkmalsstrukturen können selbst Werte einer Merkmalsstruktur sein:

CAT	<i>N</i>						
AGR	<table><tr><td>NUM</td><td><i>SG</i></td></tr><tr><td>GEN</td><td><i>MASK</i></td></tr><tr><td>CASE</td><td><i>ACC</i></td></tr></table>	NUM	<i>SG</i>	GEN	<i>MASK</i>	CASE	<i>ACC</i>
NUM	<i>SG</i>						
GEN	<i>MASK</i>						
CASE	<i>ACC</i>						

Verschiedene Kongruenz-Merkmale können so unter „Agreement“ zusammengefasst werden.

<i>Hund</i>	<table><tr><td>CAT</td><td><i>N</i></td></tr><tr><td>AGR</td><td><table><tr><td>NUM</td><td><i>SG</i></td></tr><tr><td>GEN</td><td><i>MASK</i></td></tr></table></td></tr></table>	CAT	<i>N</i>	AGR	<table><tr><td>NUM</td><td><i>SG</i></td></tr><tr><td>GEN</td><td><i>MASK</i></td></tr></table>	NUM	<i>SG</i>	GEN	<i>MASK</i>
CAT	<i>N</i>								
AGR	<table><tr><td>NUM</td><td><i>SG</i></td></tr><tr><td>GEN</td><td><i>MASK</i></td></tr></table>	NUM	<i>SG</i>	GEN	<i>MASK</i>				
NUM	<i>SG</i>								
GEN	<i>MASK</i>								

Unifikation

Zwei Merkmalsstrukturen **unifizieren**, wenn sie vereinbar sind.

Das bedeutet, dass es **keine widersprüchlichen Merkmal-Wert-Paare** in den Merkmalsstrukturen geben darf.

Das **Ergebnis einer Unifikation** enthält alle Merkmal-Wert-Paare beider Merkmalsstrukturen.

Wenn es Widersprüche in den Merkmalsstrukturen gibt, kommt die Unifikation nicht zustande – das Ergebnis der Unifikation ist dann **undefiniert**.

-> **Vereinigung zweier Mengen, aber nur bei Widerspruchsfreiheit**

$$\begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix} \sqcup \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \end{bmatrix} = \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix}$$

z.B. „der Baum“, „der Hund“

Unifikation

Unifizieren diese beide Merkmalsstrukturen?

$$\begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{FEM} \end{bmatrix} \sqcup \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \end{bmatrix}$$

Unifikation

Unifizieren diese beide Merkmalsstrukturen?

$$\begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{FEM} \end{bmatrix} \sqcup \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \end{bmatrix}$$

-> Nein, denn es besteht ein Widerspruch zwischen Gen: Fem und Gen: Mask

Unifikation

In diesem Beispiel ist die zweite Merkmalsstruktur **unterspezifiziert**, weil kein Merkmal-Wert-Paar für den Kasus angegeben wird:

$$\begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix} \sqcup \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \end{bmatrix} = \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASK} \\ \text{CASE} & \text{NOM} \end{bmatrix}$$

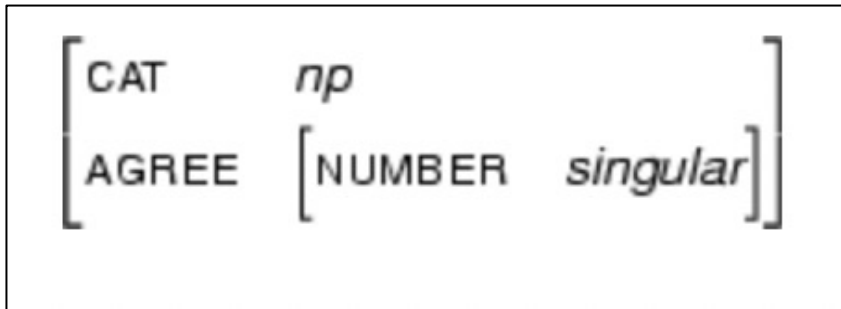
Beide Merkmalsstrukturen unifizieren, da sie nicht widersprüchlich sind.

Zum Beispiel könnte die erste Merkmalsstruktur den Artikel „der“ repräsentieren, die zweite das Substantiv „Baum“. „Baum“ kann aber nicht nur Nominativ sondern auch Akkusativ und Dativ sein. Deshalb legen wir in der Merkmalsstruktur den Kasus nicht fest. In dem Beispiel unifiziert „Baum“ mit dem Artikel „der“ zu der NP „der Baum“. Dadurch wird auch der Kasus von „Baum“ (Nominativ) in diesem konkreten Fall festgelegt.

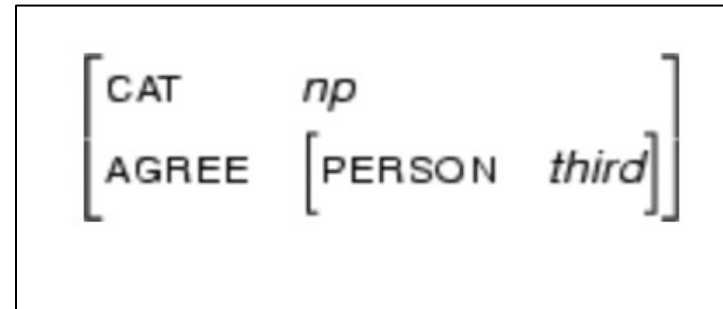
Unifikation

Unifizieren diese Merkmalsstrukturen?

f1



f2



Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node83.html#l11.fsu>

Unifikation

Unifizieren diese Merkmalsstrukturen?

f1

CAT	<i>np</i>		
AGREE	<table><tr><td>NUMBER</td><td><i>singular</i></td></tr></table>	NUMBER	<i>singular</i>
NUMBER	<i>singular</i>		

f2

CAT	<i>np</i>		
AGREE	<table><tr><td>PERSON</td><td><i>third</i></td></tr></table>	PERSON	<i>third</i>
PERSON	<i>third</i>		

Ja. Das Ergebnis der Unifikation ist:

CAT	<i>np</i>				
AGREE	<table><tr><td>NUMBER</td><td><i>singular</i></td></tr><tr><td>PERSON</td><td><i>third</i></td></tr></table>	NUMBER	<i>singular</i>	PERSON	<i>third</i>
NUMBER	<i>singular</i>				
PERSON	<i>third</i>				

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node83.html#l11.fsu>

Unifikation

Unifizieren diese Merkmalsstrukturen?

f1

f2

$$\begin{bmatrix} \text{NUMBER} & sg \\ \text{GENDER} & masc \end{bmatrix}$$
$$\begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & sg \end{bmatrix}$$

Unifikation

Unifizieren diese Merkmalsstrukturen?

f1

f2

$$\begin{bmatrix} \text{NUMBER} & sg \\ \text{GENDER} & masc \end{bmatrix}$$
$$\begin{bmatrix} \text{PERSON} & 3 \\ \text{NUMBER} & sg \end{bmatrix}$$

Ja.

[Number	sg
Gender	masc
Person	3]

Subsumption

Eine Merkmalsstruktur **f1** **subsumiert** eine andere Merkmalsstruktur **f2** genau dann, wenn die **in f1 enthaltene Information auch in f2** enthalten ist.

Die folgenden beiden Merkmalsstrukturen subsumieren sich gegenseitig, da sie genau dieselbe Information enthalten:

NUMBER	<i>sg</i>
PERSON	3

PERSON	3
NUMBER	<i>sg</i>

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node82.html#l11.sec.subsumption>

Subsumption

Subsumiert eine Merkmalsstruktur die andere?

f1

f2

[NUMBER *sg*]

[PERSON 3
NUMBER *sg*]

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node82.html#l11.sec.subsumption>

Subsumption

Subsumiert eine Merkmalsstruktur die andere?

f1

f2

[NUMBER *sg*]

[PERSON 3
NUMBER *sg*]

Ja, f1 subsumiert f2.

f2 subsumiert f1 nicht, da f2 Informationen enthält, die nicht in f1 enthalten sind.

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node82.html#l11.sec.subsumption>

Subsumption

Subsumiert eine Merkmalsstruktur die andere?

f1

f2

NUMBER	<i>sg</i>
GENDER	<i>masc</i>

PERSON	3
NUMBER	<i>sg</i>

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node82.html#l11.sec.subsumption>

Subsumption

Subsumiert eine Merkmalsstruktur die andere?

f1

f2

NUMBER	<i>sg</i>
GENDER	<i>masc</i>

PERSON	3
NUMBER	<i>sg</i>

Nein.

f1 enthält Informationen, die nicht in f2 enthalten sind, und vice versa.

Quelle: <https://cs.union.edu/~striegnk/courses/nlp-with-prolog/html/node82.html#l11.sec.subsumption>

Unifikation und Subsumption in NLTK

Methode unify():

`f1.unify(f2)`

-> gibt bei erfolgreicher Unifikation die unifizierte Merkmalsstruktur zurück, ansonsten NONE

Methode subsumes():

`f1.subsumes(f2)`

-> gibt bei erfolgreicher Subsumption TRUE zurück, ansonsten FALSE

Unifikation und Subsumption in NLTK

Beispiele:

```
f0 = FeatStruct("[AGR=[NUM=p1]]")  
f1 = FeatStruct("[AGR=[NUM=sg, PERS=3]]")
```

```
#Subsumption (f0  $\sqsubseteq$  f1)  
f0.subsumes(f1)
```

Was ist das Output?

Unifikation und Subsumption in NLTK

Beispiele:

```
f0 = FeatStruct("[AGR=[NUM=p1]]")  
f1 = FeatStruct("[AGR=[NUM=sg, PERS=3]]")
```

```
#Subsumption (f0  $\sqsubseteq$  f1)  
f0.subsumes(f1)
```

FALSE

```
#Unification (f0  $\sqcup$  f1)  
print(f0.unify(f1))
```

Was ist das Output?

Unifikation und Subsumption in NLTK

Beispiele:

```
f0 = FeatStruct("[AGR=[NUM=p1]]")  
f1 = FeatStruct("[AGR=[NUM=sg, PERS=3]]")
```

```
#Subsumption (f0  $\sqsubseteq$  f1)  
f0.subsumes(f1)
```

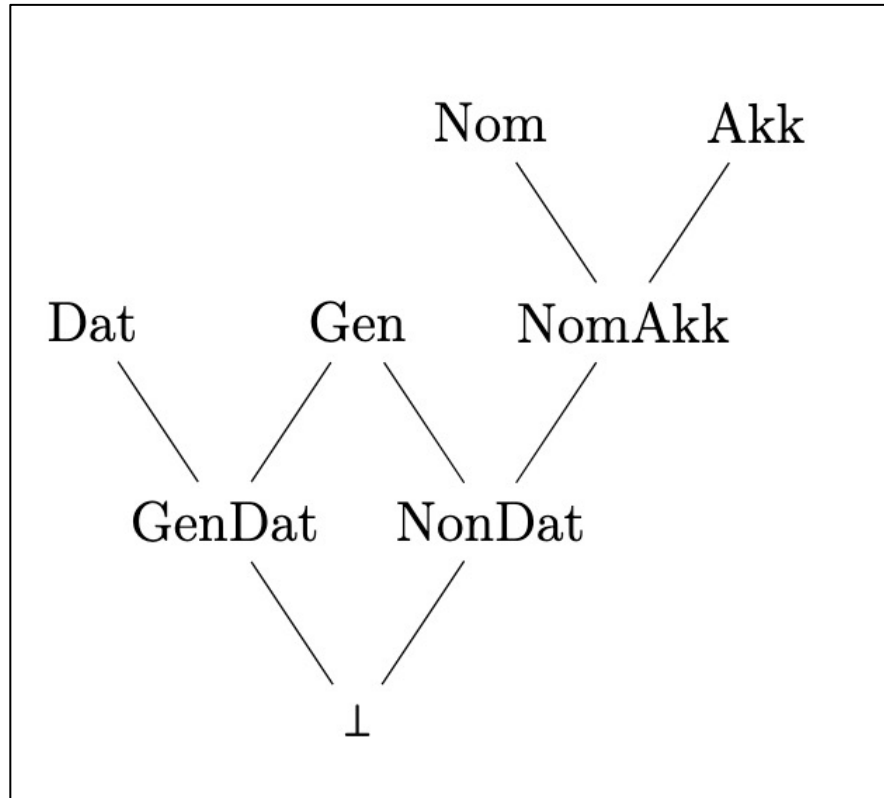
FALSE

```
#Unification (f0  $\sqcup$  f1)  
print(f0.unify(f1))
```

NONE

Getypete Merkmalsstrukturen

Beispiel einer Typenhierarchie:



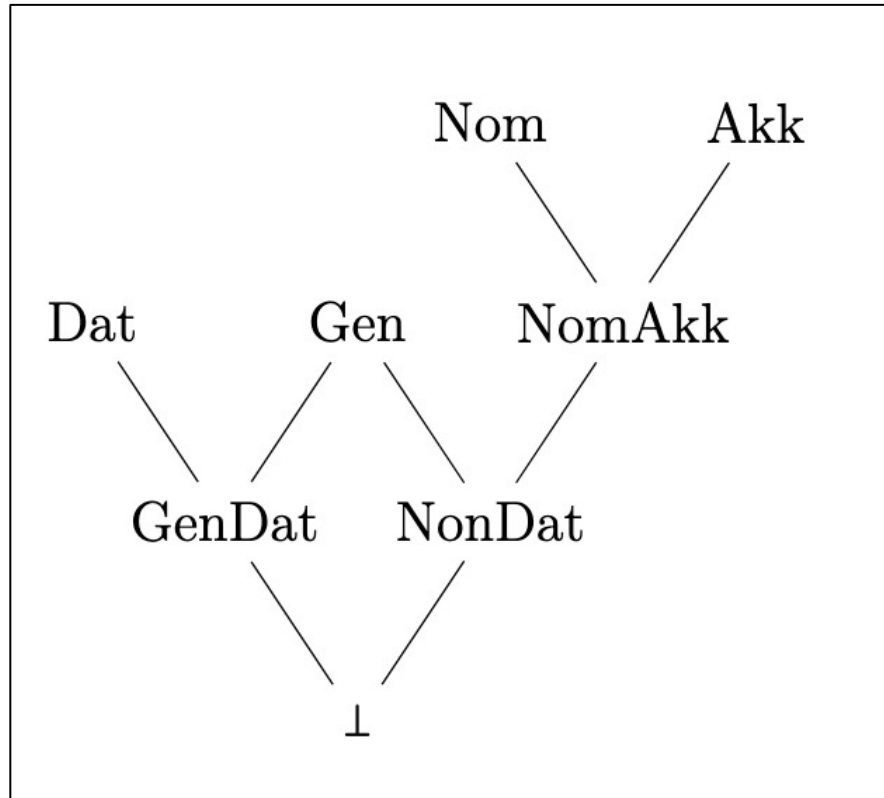
\perp = **Bottom, leere Merkmalsstruktur**
- unifiziert mit jeder Merkmalsstruktur
- subsumiert jede Merkmalsstruktur

Die Kanten des Baumes drücken Subsumption aus (von unten nach oben gelesen).

Je höher wir im Baum kommen, desto spezifischer die Information.

Getypete Merkmalsstrukturen

Beispiel einer Typenhierarchie:



In NLTK definieren wir die Typenhierarchie als Dictionary:

```
case_hierarchy = {  
    "nondat": ["gen", "nomakk"],  
    "gendat": ["gen", "dat"],  
    "nomakk": ["nom", "akk"],  
    "nom": [],  
    "gen": [],  
    "dat": [],  
    "akk": []  
}
```

Getypte Merkmalsstrukturen

In NLTK definieren wir die Typenhierarchie als Dictionary:

```
case_hierarchy = {  
    "nondat": ["gen", "nomakk"],  
    "gendat": ["gen", "dat"],  
    "nomakk": ["nom", "akk"],  
    "nom": [],  
    "gen": [],  
    "dat": [],  
    "akk": []  
}
```

Verwenden wir eine Typenhierarchie in unserer Grammatik, markieren wir die getypten Features mit *

(z.B. *CASE* statt CASE):

```
DET[GEN=mask,*CASE*=Nom] -> "der"  
DET[GEN=mask,*CASE*=Gen] -> "des"  
DET[GEN=fem,*CASE*=NomAkk] -> "die"
```