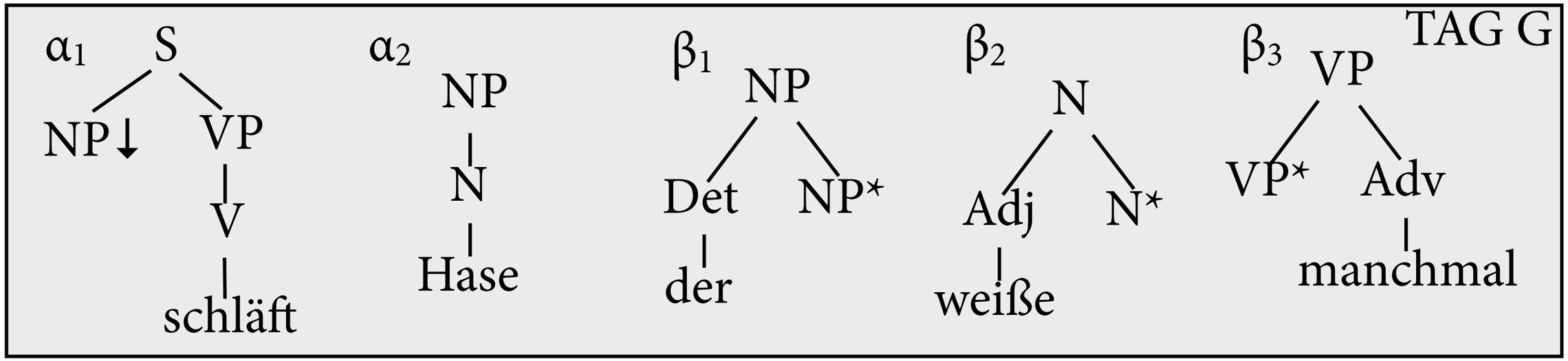
Featurestrukturen in TAG

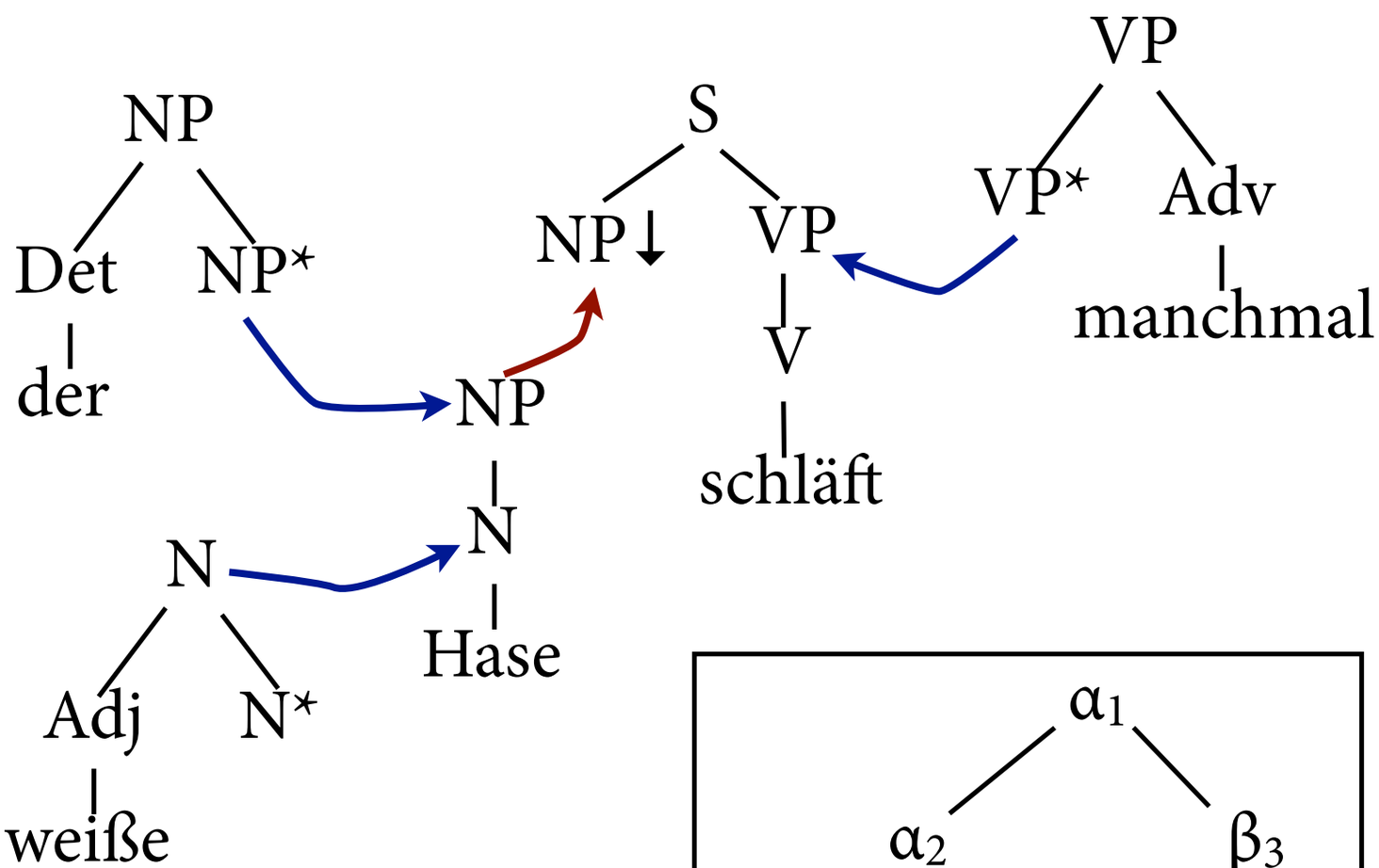
Vorlesung “Grammatikformalismen”

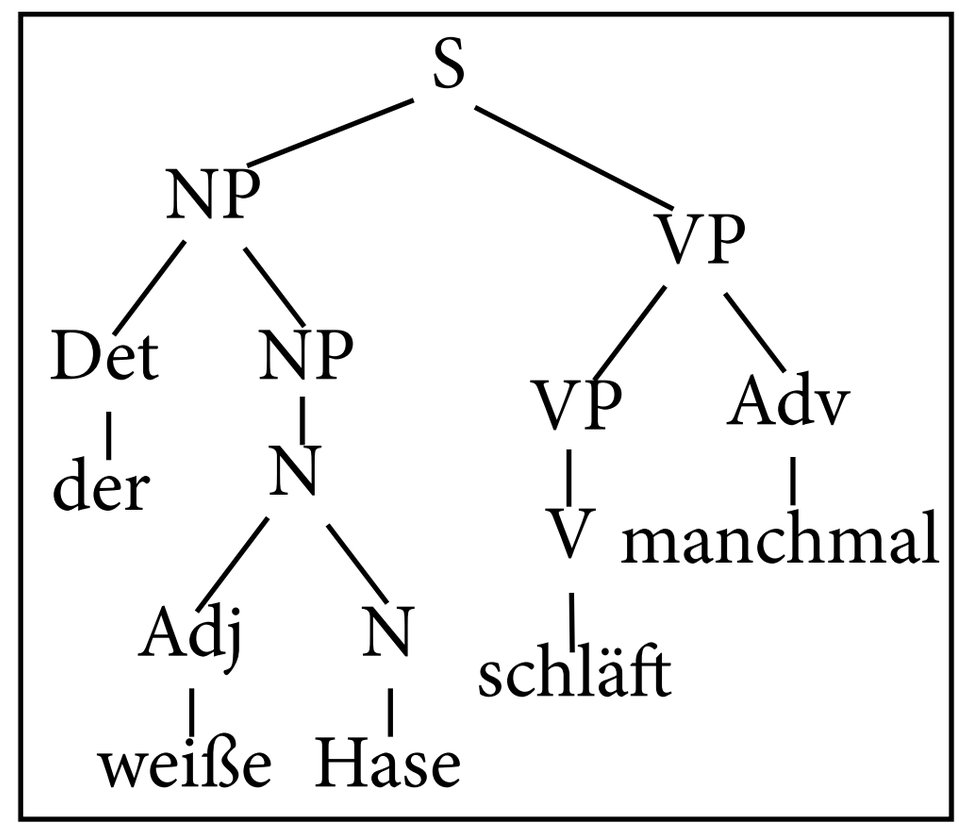
Alexander Koller

23. April 2019

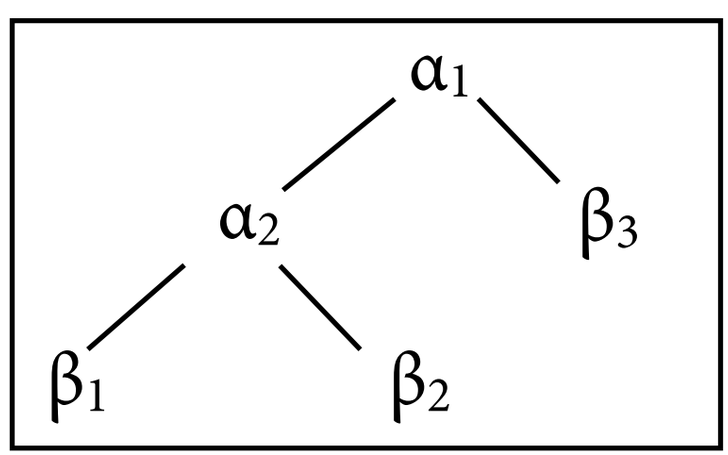
Ein Beispiel







abgeleiteter Baum



Ableitungsbaum

Extended domain of locality

- Lokalität: Informationen, die in einem einzigen Grammatikeintrag vorliegen.
  - vom Grammatikentwickler als Einheit gedacht
  - vom Parser als Einheit verarbeitet
- KfG: einzelne Produktionsregel.
- TAG: Elementarbaum, d.h. viel größere Einheit.

Extended domain of locality

- Einheit Elementarbaum:
  - enthält Substitutionsknoten für alle Komplemente des lexikalischen Ankers
  - d.h. alles, was für semantische Interpretation des Wortes nötig ist
  - “Bewegungen” innerhalb des Elementarbaums können vorausberechnet und lexikalisiert werden; Parser muss deshalb nicht mehr drüber nachdenken.
- Das erleichtert Grammatikentwicklung und Parsing.

Kongruenz mit kfGs

S → NPsg VPsg
VPsg → IVsg
VPsg → TVsg NP
VPsg → DVsg NP NP
IVsg → schläft
TVsg → isst
DVsg → gibt

S → NPpl VPpl
VPpl → IVpl
VPpl → TVpl NP
VPpl → DVpl NP NP
IVpl → schläft
TVpl → isst
DVpl → gibt

Das Problem

- KfGs: Kongruenzartige Information muss in Nichtterminalen codiert werden.
- Kongruenz und Subkategorisierung nur zwei von vielen Constraints, die interagieren können.
- Grammatik wird groß und unwartbar.

Features

- Alternative: An Nichtterminale *Merkmale* (engl. *Features*) anhängen, deren Werte separat verarbeitet werden können.

- In Grammatik enthalten Features Variablen, die durch *Unifikation* ihre Werte bekommen.

Kongruenz mit Features

S → NP[num:sg] VP[num:sg]
VP[num:sg] → IV[num:sg]
VP[num:sg] → TV[num:sg] NP
VP[num:sg] → DV[num:sg] NP NP
IV[num:sg] → schläft
TV[num:sg] → isst
DV[num:sg] → gibt

S → NP[num:pl] VP[num:pl]
VP[num:pl] → IV[num:pl]
VP[num:pl] → TV[num:pl] NP
VP[num:pl] → DV[num:pl] NP NP
IV[num:pl] → schläft
TV[num:pl] → isst
DV[num:pl] → gibt

Kongruenz mit Features

S → NP[num:X] VP[num:X]
VP[num:X] → IV[num:X]
VP[num:X] → TV[num:X] NP
VP[num:X] → DV[num:X] NP NP
IV[num:sg] → schläft
TV[num:sg] → isst
DV[num:sg] → gibt

S → NP[num:X] VP[num:X]
VP[num:X] → IV[num:X]
VP[num:X] → TV[num:X] NP
VP[num:X] → DV[num:X] NP NP
IV[num:pl] → schläft
TV[num:pl] → isst
DV[num:pl] → gibt

Kongruenz mit Features

S → NP[num:X] VP[num:X]
VP[num:X] → IV[num:X]
VP[num:X] → TV[num:X] NP
VP[num:X] → DV[num:X] NP NP

IV[num:sg] → schläft
TV[num:sg] → isst
DV[num:sg] → gibt

IV[num:pl] → schläft
TV[num:pl] → isst
DV[num:pl] → gibt

Feature-Strukturen

- Formalisierung mit *Feature-Strukturen (FSen)*: gruppiert mehrere Zuweisungen von Werten an Features.

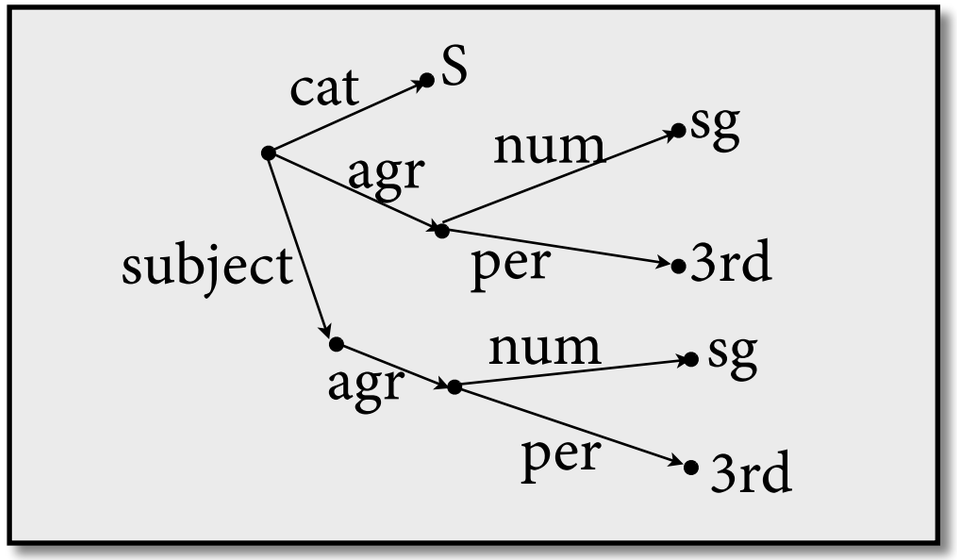
- Verschiedene Featurestrukturen können verschiedene Features enthalten.

- Werte von Features können sein:
  - atomare Werte
  - Featurestrukturen

Zwei Sichten auf FSen

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Attribut-Wert-Matrix
(AVM)



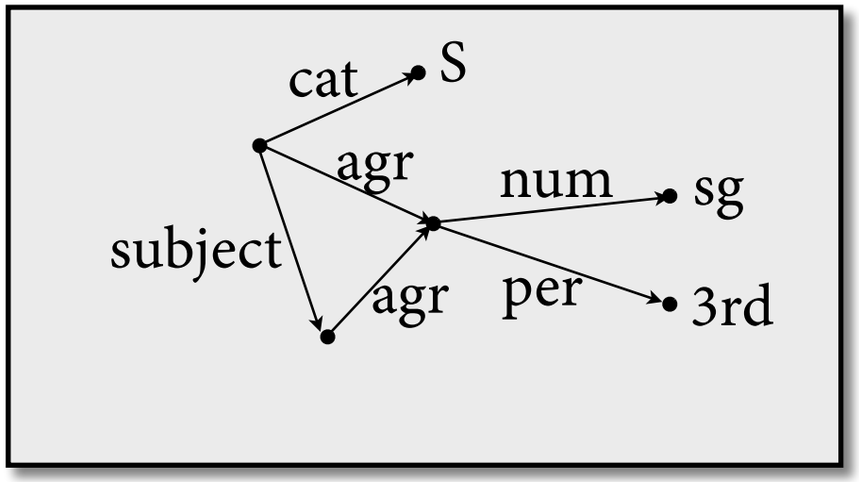
gerichteter azyklischer Graph
(DAG)

Manche Theorien von Featurestrukturen nehmen
AVMs als Beschreibungen von DAGs.
Wir differenzieren hier nicht dazwischen.

Reentrancy

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{AGR} & \boxed{1}\begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Attribut-Wert-Matrix
(AVM)



gerichteter azyklischer Graph
(DAG)

Verschiedene Pfade im Graphen können zum gleichen Knoten führen.

Identität wird in der AVM durch *Koindizierung* dargestellt.

Subsumption

- Eine FS F_1 *subsumiert* die FS F_2 ($F_1 \sqsubseteq F_2$), wenn alle Informationen in F_1 auch in F_2 stehen:
  - jeder Pfad in F_1 existiert in F_2
  - wenn an einem Pfad in F_1 ein atomarer Wert steht, dann steht der gleiche Wert auch an diesem Pfad in F_2
  - wenn zwei Pfade in F_1 zum gleichen Knoten führen, dann auch in F_2

$$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqsubseteq \not\sqsupseteq \begin{bmatrix} \text{CAT} & \text{S} \\ \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERSON} & \text{3rd} \end{bmatrix} \\ \text{SUBJECT} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Unifikation

- In Featuregrammatik trägt jede Regel nur ein bisschen Information über die FS bei.

- *Unifikation*: Informationen aus zwei FSen F_1, F_2 zusammenführen. $F = F_1 \sqcup F_2$ ist FS, so dass
  - subsumiert beide: $F_1 \sqsubseteq F$, $F_2 \sqsubseteq F$
  - erfindet nichts dazu: für alle F' mit $F_1 \sqsubseteq F'$, $F_2 \sqsubseteq F'$ gilt $F \sqsubseteq F'$.

- Unifikation kann fehlschlagen, wenn F_1 und F_2 widersprüchliche Informationen enthalten.

Unifikation

- Unifikation kann fehlschlagen:

$$[\text{num}\ \ \text{pl}] \sqcup [\text{num}\ \ \text{sg}] \quad \rightarrow \text{fail}$$

- Unifikation sensibel für Reentrancy:

$$\begin{bmatrix} \text{agr} & [\text{pers}\ \ \text{3rd}] \\ \text{subject} & [\text{agr}\ [\text{pers}\ \ \text{3rd}]] \end{bmatrix} \sqcup \begin{bmatrix} \text{agr} & [\text{num}\ \ \text{sg}] \\ \text{subject} & [\text{agr}\ [\text{num}\ \ \text{pl}]] \end{bmatrix} = \begin{bmatrix} \text{agr} & \begin{bmatrix} \text{pers} & \text{3rd} \\ \text{num} & \text{sg} \end{bmatrix} \\ \text{subject} & \begin{bmatrix} \text{agr} & \begin{bmatrix} \text{pers} & \text{3rd} \\ \text{num} & \text{pl} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{agr} & \boxed{1}\ [\text{pers}\ \ \text{3rd}] \\ \text{subject} & [\text{agr}\ \boxed{1}[\text{pers}\ \ \text{3rd}]] \end{bmatrix} \sqcup \begin{bmatrix} \text{agr} & [\text{num}\ \ \text{sg}] \\ \text{subject} & [\text{agr}\ [\text{num}\ \ \text{pl}]] \end{bmatrix} \quad \rightarrow \text{fail}$$

Featuregrammatiken

- Eine *kontextfreie Grammatik mit Features* (FCFG) ist eine kfG, in der jedes Auftreten eines Nichtterminals in einer Produktionsregel mit einer Featurestruktur versehen sein kann.

S → NP[num $\boxed{1}$] VP[num $\boxed{1}$]
VP[num $\boxed{1}$] → IV[num $\boxed{1}$]
IV[num: sg] → sleeps

Ableitungen in FCFGs

- Regel A[F0] → B[F1] C[F2] kann man als größere Featurestruktur sehen:

$$F = \begin{bmatrix} A & F0 \\ B & F1 \\ C & F2 \end{bmatrix}$$

- Denke Ableitungen bottom-up:
  - um B[G1] und C[G2] zu kombinieren,
  - berechne G = F ⊔ [B G1] ⊔ [C G2] (darf nicht fehlschlagen)
  - und leite daraus A[G.A] ab.

- Sprache, Wort-, Parsingproblem wie für kfGs.

Ein Beispiel

S → NP[num $\boxed{1}$] VP[num $\boxed{1}$]
VP[num $\boxed{1}$] → IV[num $\boxed{1}$]
IV[num: sg] → sleeps
NP[num: sg] → John

S ⇒ NP[num: sg] VP[num: sg]
⇒ NP[num: sg] IV[num: sg]
⇒ John IV[num: sg] ⇒ John sleeps

$$\begin{bmatrix} S & \\ NP & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \\ VP & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup [NP\ [num\ sg]] \sqcup [VP\ [num\ sg]] = \begin{bmatrix} S & \\ NP & \begin{bmatrix} num & \boxed{1} & sg \end{bmatrix} \\ VP & \begin{bmatrix} num & \boxed{1} & sg \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} VP & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \\ IV & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup [IV\ [num\ sg]] = \begin{bmatrix} VP & \begin{bmatrix} num & \boxed{1} & sg \end{bmatrix} \\ IV & \begin{bmatrix} num & \boxed{1} & sg \end{bmatrix} \end{bmatrix}$$

Ein Beispiel

S → NP[num $\boxed{1}$] VP[num $\boxed{1}$]
VP[num $\boxed{1}$] → IV[num $\boxed{1}$]
IV[num: sg] → sleeps
NP[num: sg] → John

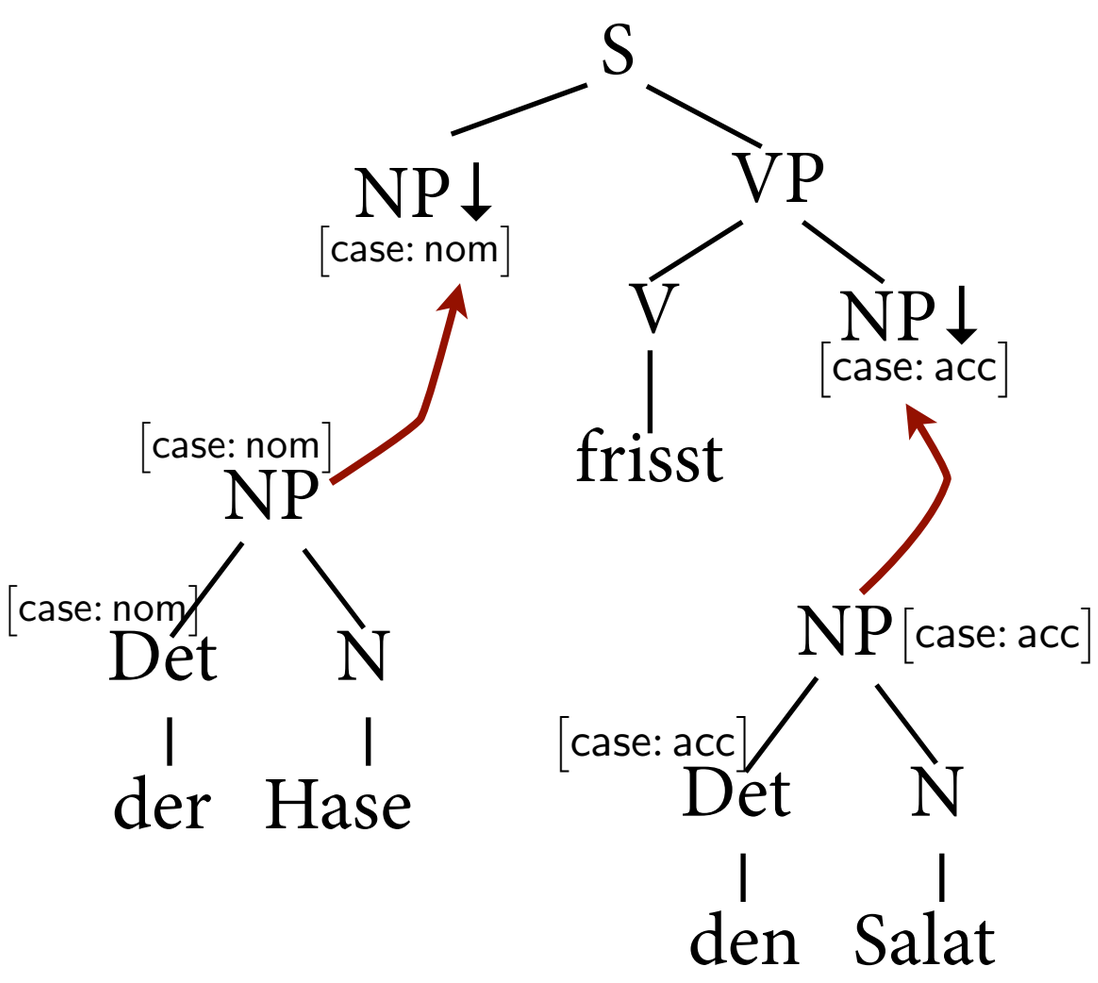
$$S \nRightarrow NP[num: sg]\ VP[num: pl]$$

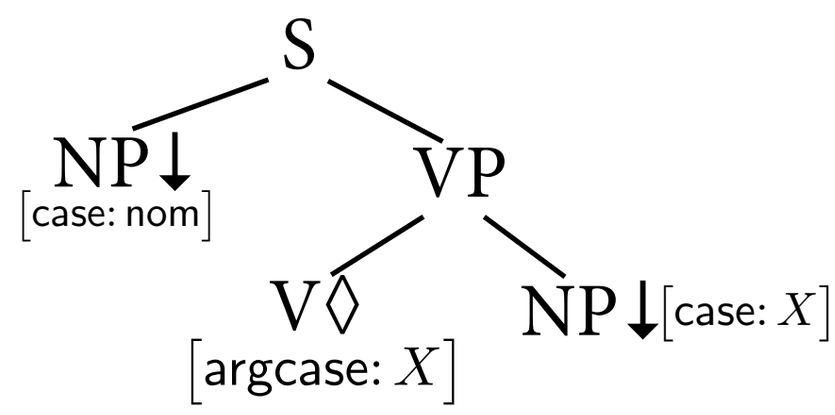
(also $S \nRightarrow^{*}$ John sleep)

$$\begin{bmatrix} S & \\ NP & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \\ VP & \begin{bmatrix} num & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup [NP\ [num\ sg]] \sqcup [VP\ [num\ pl]] \rightarrow fail$$

TAG mit Features

- Erste Idee: Knoten von Elementarbäumen mit Feature-Strukturen dekorieren.

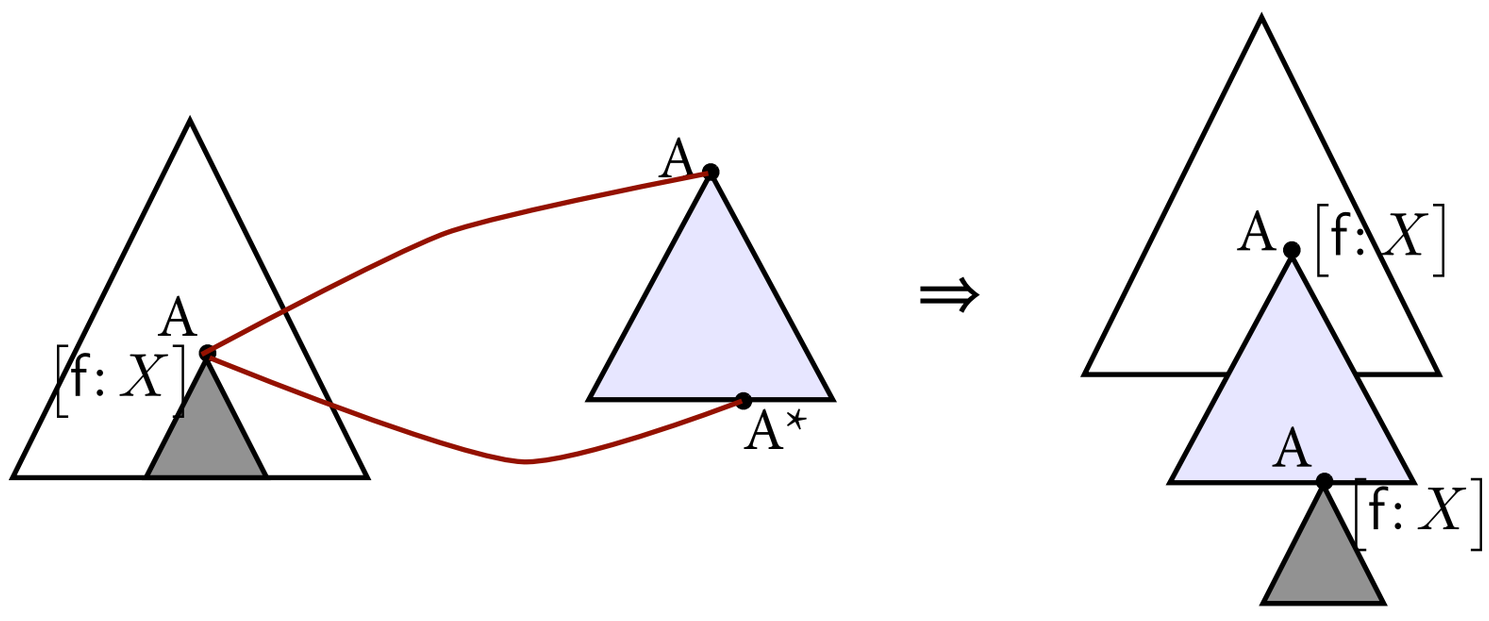




Im Lexikon spezifizieren:
frisst: V, argcase=acc
hilft: V, argcase=dat
usw.

Ein Problem

- Was passiert mit der Featurestruktur an einem Knoten, wenn man dort adjungiert?
  - X mit FS bei Wurzel unifizieren? Oder mit FS bei Fußknoten?
  - oder mit beiden? (Konsequenz: FS(Wurzel) = FS(Fußknoten))



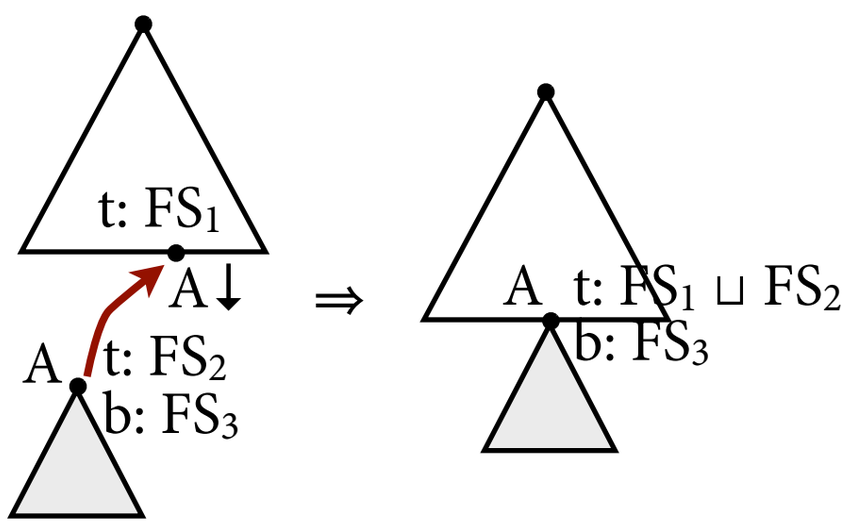
Bessere Intuition:
Adjunktion bricht einen Knoten in zwei Hälften auf.

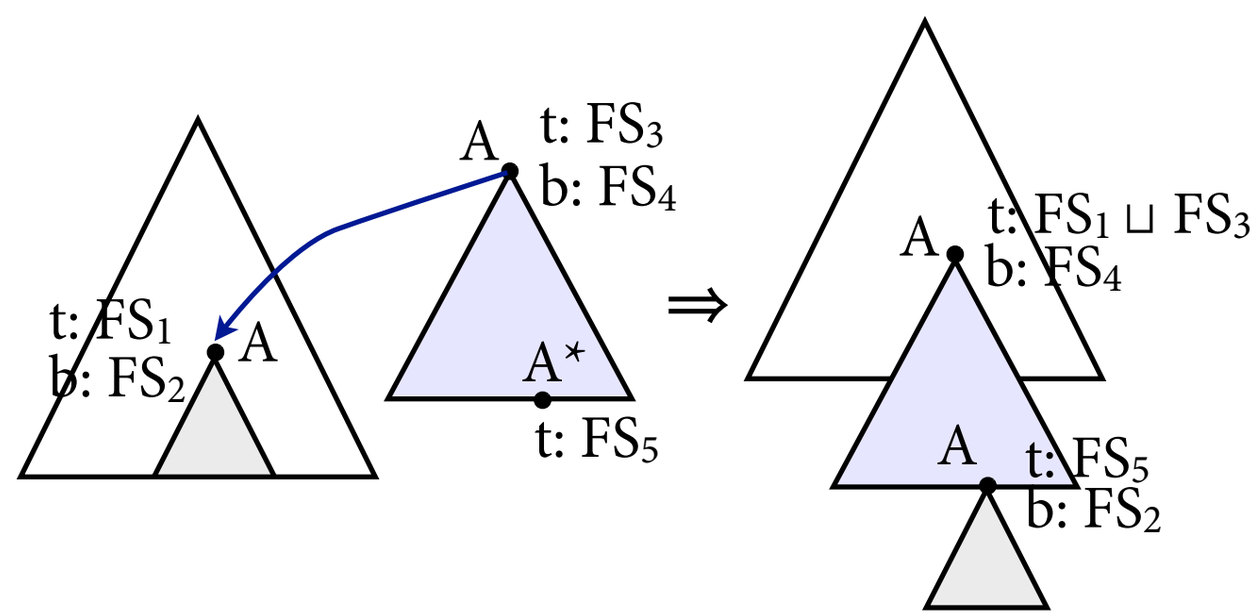
FTAG

- In Feature-TAG trägt jeder Knoten *zwei* FSen: für obere und untere Hälfte des Knotens.
  - Wo man nicht adjungieren kann (z.B. Fußknoten), reicht eine FS
  - Obere FS von Wurzeln, untere FS von Blättern normalerweise leer
  - innere Knoten haben obere & untere FS
- Am Ende der Ableitung werden obere und untere FS jedes Knotens unifiziert.
  - wenn das nicht geht, schlägt Ableitung fehl
  - zwischendurch dürfen FSen am gleichen Knoten nicht-unifizierbar sein

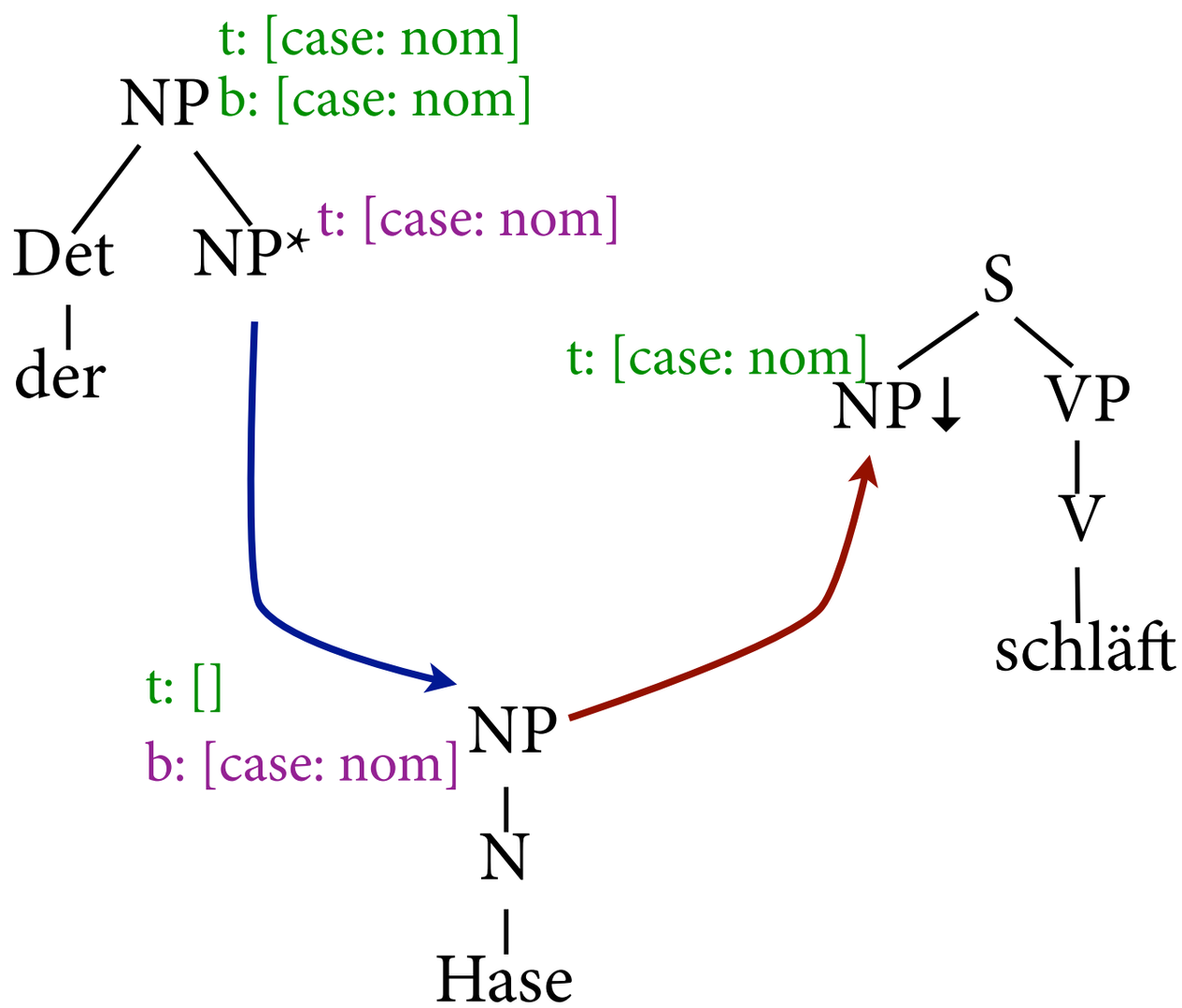
Operationen von FTAG

- Substitution von t in u:
  - unifiziere top(u) mit top(Wurzel von t)
- Adjunktion von t in u:
  - “obere Kopie” von u bekommt top(u) + FS(Wurzel von t)
  - “untere Kopie” von u bekommt bottom(u) + FS(Fußknoten von t)



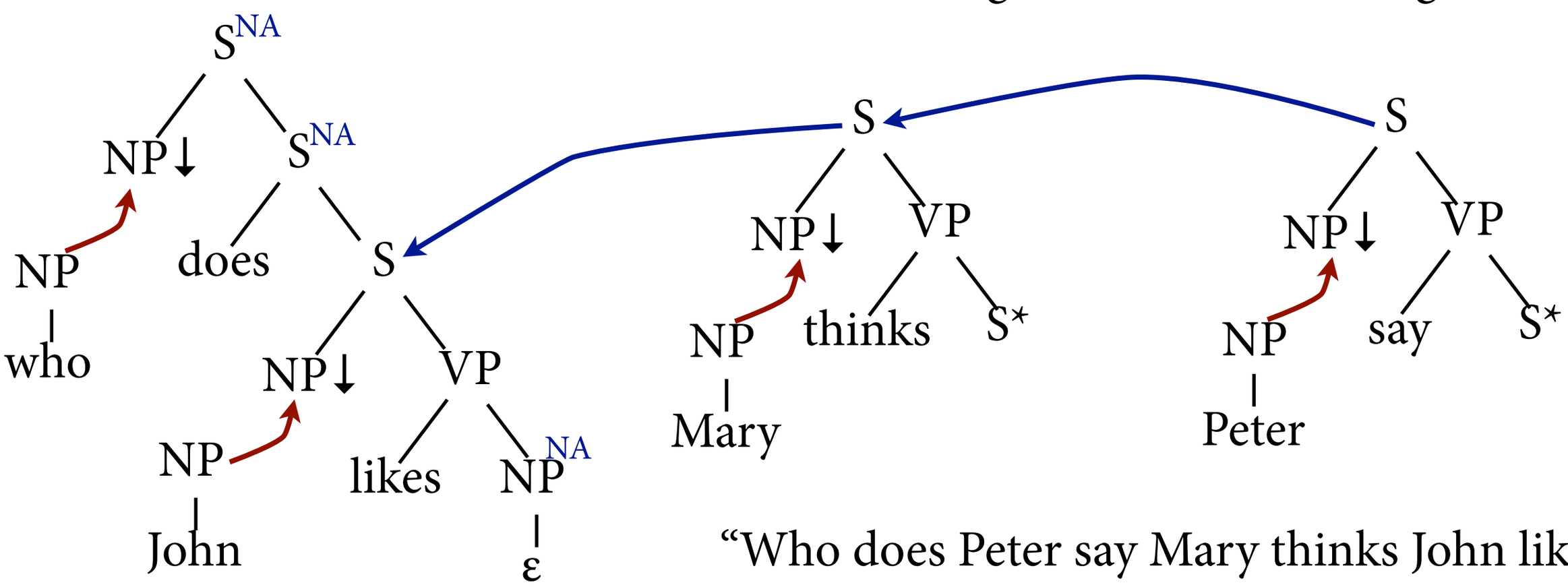


Beispiel


t: [case: nom]
NP
b: [case: nom]
Det
NP*
t: [case: nom]
der
S
t: [case: nom]
NP↓
VP
V
schläft
t: []
b: [case: nom]
NP
N
Hase

Fernabhängigkeiten

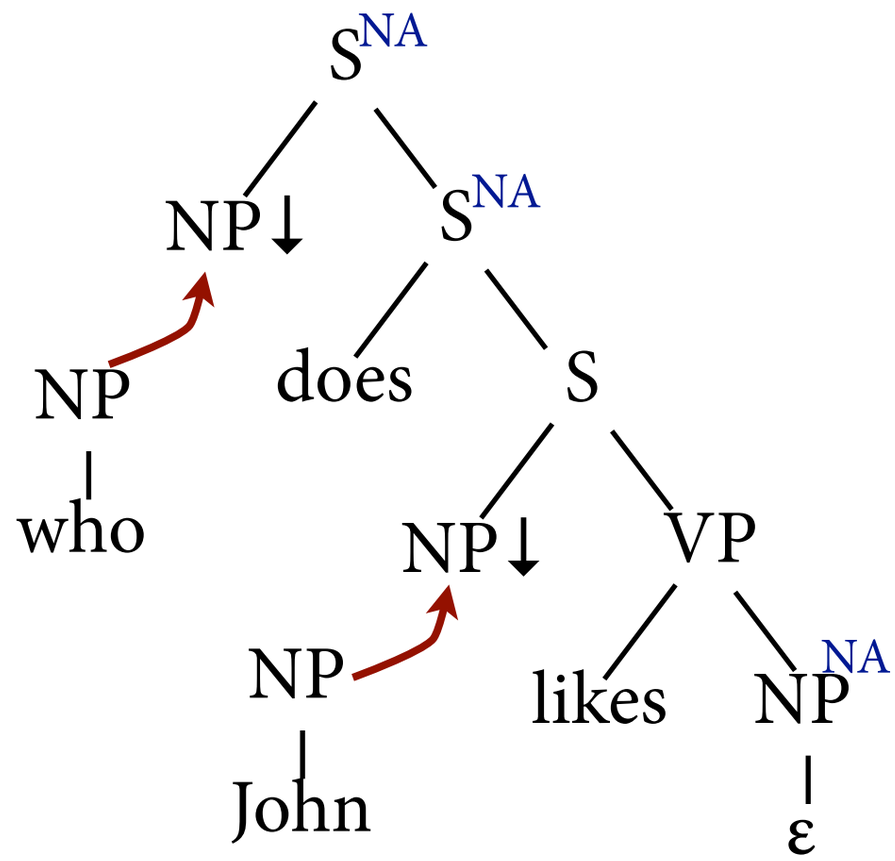
- Fernabhängigkeiten (z.B. wh-Bewegung):
  - “Bewegung” innerhalb eines Elementarbaums
  - “bewegtes” wh-Wort wird durch Adjunktion immer weiter vom Verb weggedrückt
  - NB: Bewegung ist nur Motivation für konkreten Elementarbaum; in TAG-Ableitungen wird nichts bewegt!



“Who does Peter say Mary thinks John likes”

Kontrolle von Fernabhängigkeiten

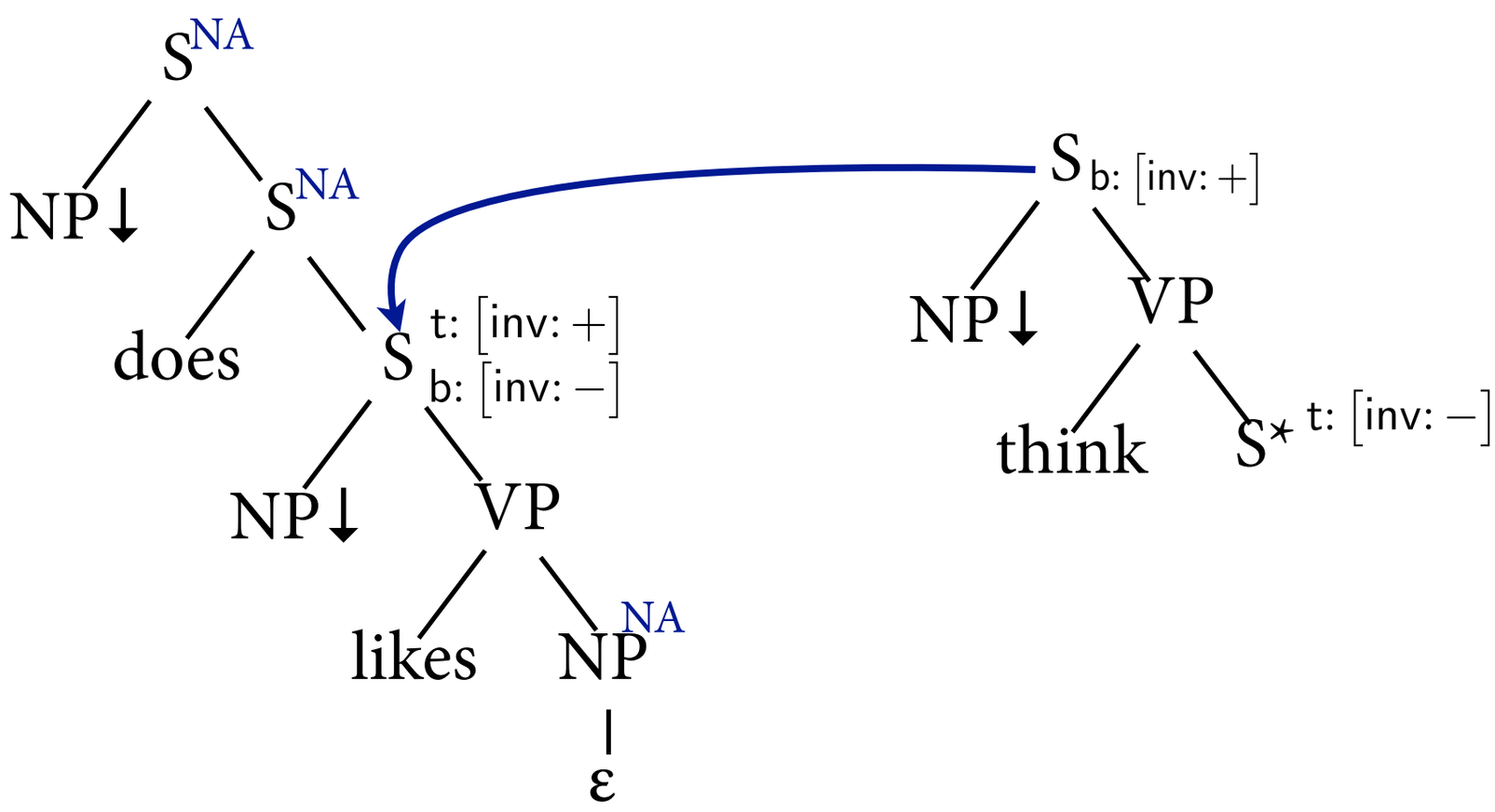
- Problem in unserer Analyse: “who does John likes” wird als grammatisch akzeptiert.



- Wie erzwingen wir, dass ein einbettendes Verb adjungiert werden *muss*?

Kontrolle von Fernabhängigkeiten

- Lösung: inkompatible Features werden durch Adjunktion aufgelöst.



Zusammenfassung

- Featuregrammatiken: Erlauben kompakte Repräsentation von grammatischen Informationen.
- Unifikation: Fasst Information aus verschiedenen Featurestrukturen zusammen.
- FTAG: Teile jeden Knoten in obere und untere Hälfte auf; jede Hälfte bekommt eine FS.