

# Featurestrukturen in TAG

Vorlesung “Grammatikformalismen”  
Alexander Koller

28. April 2017

# Kongruenz mit kfGs

$S \rightarrow \text{NPsg VPsg}$

$\text{VPsg} \rightarrow \text{IVsg}$

$\text{VPsg} \rightarrow \text{TVsg NP}$

$\text{VPsg} \rightarrow \text{DVsg NP NP}$

$\text{IVsg} \rightarrow \text{schläft}$

$\text{TVsg} \rightarrow \text{isst}$

$\text{DVsg} \rightarrow \text{gibt}$

$S \rightarrow \text{NPpl VPpl}$

$\text{VPpl} \rightarrow \text{IVpl}$

$\text{VPpl} \rightarrow \text{TVpl NP}$

$\text{VPpl} \rightarrow \text{DVpl NP NP}$

$\text{IVpl} \rightarrow \text{schläft}$

$\text{TVpl} \rightarrow \text{isst}$

$\text{DVpl} \rightarrow \text{gibt}$

# 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

$S \rightarrow NP[\text{num:X}] VP[\text{num:X}]$

$VP[\text{num:X}] \rightarrow IV[\text{num:X}]$

$VP[\text{num:X}] \rightarrow TV[\text{num:X}] NP$

$VP[\text{num:X}] \rightarrow DV[\text{num:X}] NP NP$

$IV[\text{num:sg}] \rightarrow \text{schläft}$

$TV[\text{num:sg}] \rightarrow \text{isst}$

$DV[\text{num:sg}] \rightarrow \text{gibt}$

$IV[\text{num:pl}] \rightarrow \text{schläft}$

$TV[\text{num:pl}] \rightarrow \text{isst}$

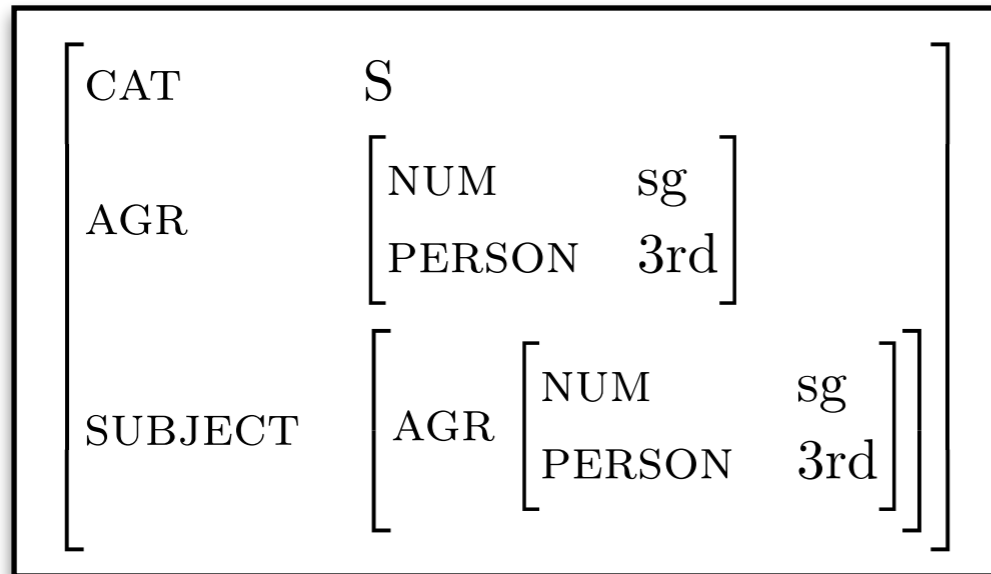
$DV[\text{num:pl}] \rightarrow \text{gibt}$

# Feature-Strukturen

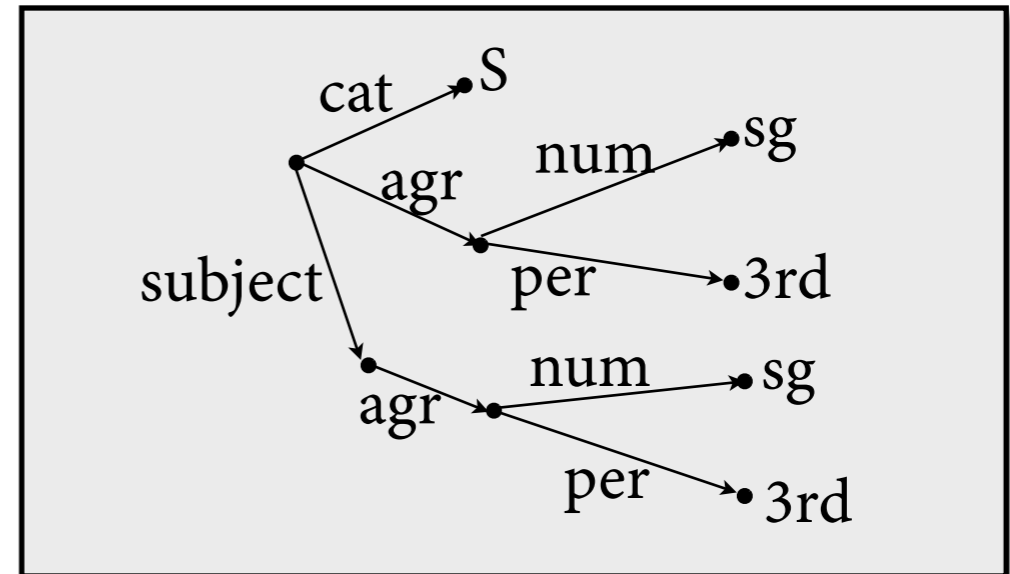
- Formalisierung mit *Feature-Strukturen (FSen)*:  
gruppiert mehrere Zuweisungen von Werten an Features.
- Verschiedene Featurestrukturen können  
verschiedene Features enthalten.
- Werte von Features können sein:
  - ▶ atomare Werte
  - ▶ Featurestrukturen



# Zwei Sichten auf FSen



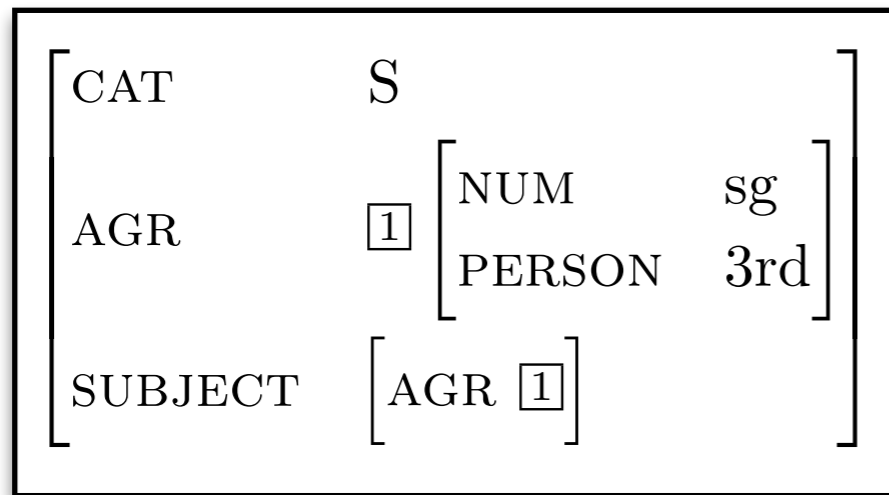
Attribut-Wert-Matrix  
(AVM)



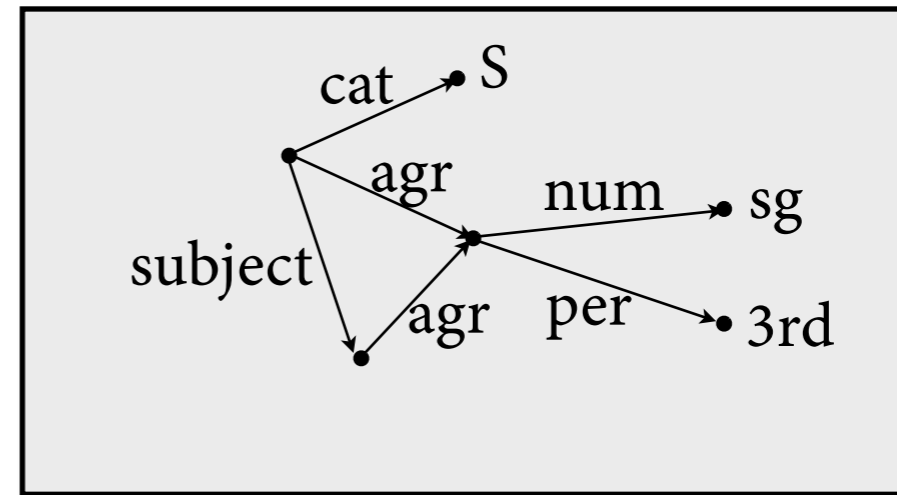
gerichteter azyklischer Graph  
(DAG)

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

# Reentrancy



Attribut-Wert-Matrix  
(AVM)



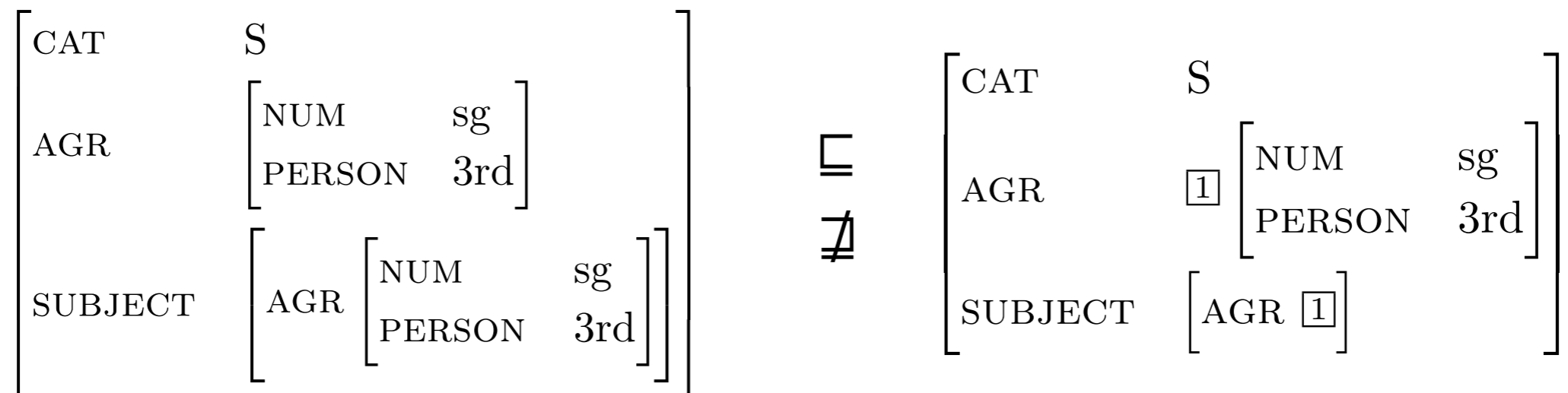
gerichteter azyklischer Graph  
(DAG)

Verschiedene Pfade im Graphen können zum gleichen Knoten führen.

Identität wird in der AVM durch *Koindizierung* erzwungen.

# 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  $F_2$



# 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:

$[num\ pl] \sqcup [num\ sg] \rightarrow fail$

- Unifikation sensibel für Reentrancy:

$$\left[ \begin{array}{l} agr \quad [pers\ 3rd] \\ subject \quad [agr \quad [pers\ 3rd]] \end{array} \right] \sqcup \left[ \begin{array}{l} agr \quad [num\ sg] \\ subject \quad [agr \quad [num\ pl]] \end{array} \right] = \left[ \begin{array}{l} agr \quad \left[ \begin{array}{l} pers\ 3rd \\ num\ sg \end{array} \right] \\ subject \quad \left[ agr \quad \left[ \begin{array}{l} pers\ 3rd \\ num\ pl \end{array} \right] \right] \end{array} \right]$$

$$\left[ \begin{array}{l} agr \quad \boxed{1} [pers\ 3rd] \\ subject \quad [agr \quad \boxed{1} [pers\ 3rd]] \end{array} \right] \sqcup \left[ \begin{array}{l} agr \quad [num\ sg] \\ subject \quad [agr \quad [num\ pl]] \end{array} \right] \rightarrow fail$$

# Featuregrammatiken

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

$$\begin{aligned} S &\rightarrow NP[\text{num } \boxed{1}] \quad VP[\text{num}\boxed{1}] \\ VP[\text{num } \boxed{1}] &\rightarrow IV[\text{num}\boxed{1}] \\ IV[\text{num: sg}] &\rightarrow \text{sleeps} \end{aligned}$$

# Ableitungen in FCFGs

- Regel  $A[F0] \rightarrow B[F1] C[F2]$  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  $G = F \sqcup [B \ G1] \sqcup [C \ G2]$  (darf nicht fehlschlagen)
  - ▶ und leite daraus  $A[G.A]$  ab.
- Sprache, Wort-, Parsingproblem wie für kfGs.

# Ein Beispiel

$S \rightarrow NP[\text{num } \boxed{1}] \quad VP[\text{num } \boxed{1}]$

$VP[\text{num } \boxed{1}] \rightarrow IV[\text{num } \boxed{1}]$

$IV[\text{num: sg}] \rightarrow \text{sleeps}$

$NP[\text{num: sg}] \rightarrow \text{John}$

$S \Rightarrow NP[\text{num: sg}] \quad VP[\text{num: sg}]$

$\Rightarrow NP[\text{num: sg}] \quad IV[\text{num: sg}]$

$\Rightarrow \text{John } IV[\text{num: sg}] \quad \Rightarrow \text{John sleeps}$

$$\begin{bmatrix} S \\ NP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \\ \text{num } \boxed{1} \end{array} \right] \\ VP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \\ \text{num } \boxed{1} \end{array} \right] \end{bmatrix} \sqcup [NP \quad [\text{num } \text{sg}]] \sqcup [VP \quad [\text{num } \text{sg}]] = \begin{bmatrix} S \\ NP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \text{ sg} \\ \text{num } \boxed{1} \text{ sg} \end{array} \right] \\ VP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \text{ sg} \\ \text{num } \boxed{1} \text{ sg} \end{array} \right] \end{bmatrix}$$

$$\begin{bmatrix} VP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \\ \text{num } \boxed{1} \end{array} \right] \\ IV \quad \left[ \begin{array}{l} \text{num } \boxed{1} \\ \text{num } \boxed{1} \end{array} \right] \end{bmatrix} \sqcup [IV \quad [\text{num } \text{sg}]] = \begin{bmatrix} VP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \text{ sg} \\ \text{num } \boxed{1} \text{ sg} \end{array} \right] \\ IV \quad \left[ \begin{array}{l} \text{num } \boxed{1} \text{ sg} \\ \text{num } \boxed{1} \text{ sg} \end{array} \right] \end{bmatrix}$$



# Ein Beispiel

$S \rightarrow NP[\text{num } \boxed{1}] \quad VP[\text{num } \boxed{1}]$

$VP[\text{num } \boxed{1}] \rightarrow IV[\text{num } \boxed{1}]$

$IV[\text{num: sg}] \rightarrow \text{sleeps}$

$NP[\text{num: sg}] \rightarrow \text{John}$

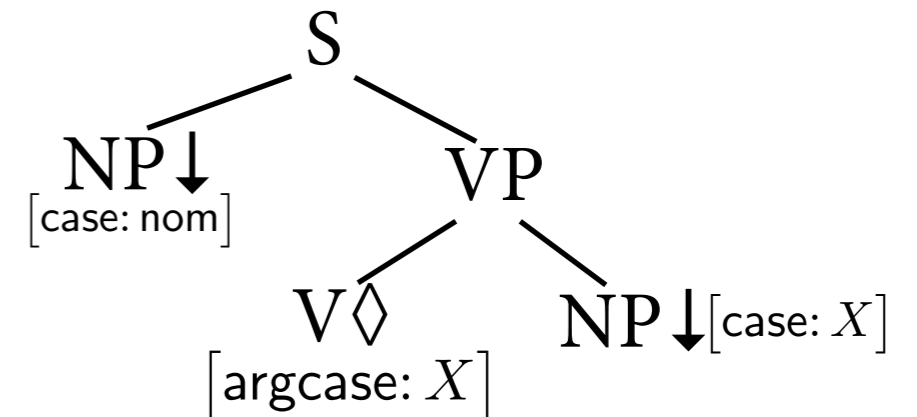
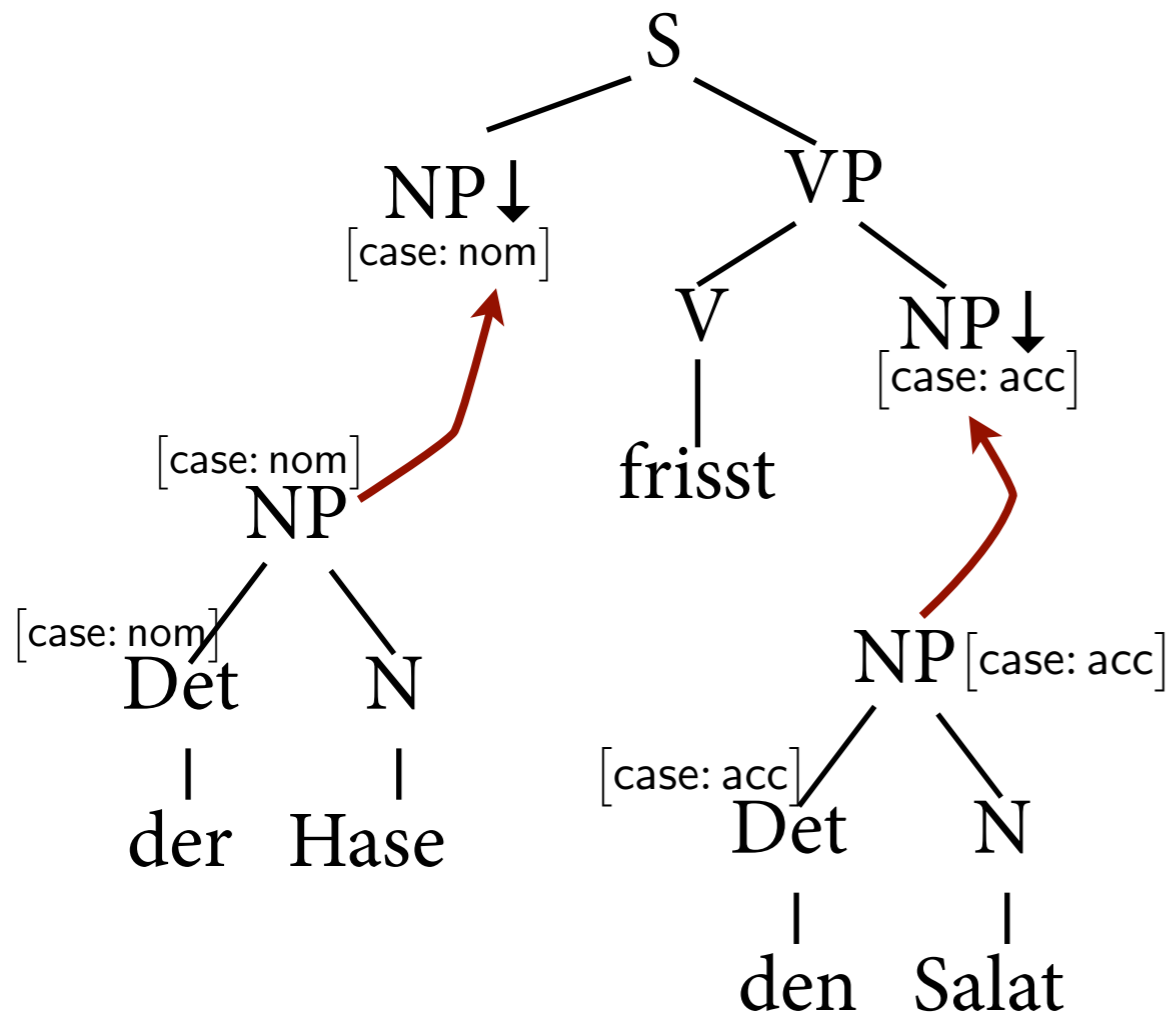
$S \not\Rightarrow NP[\text{num: sg}] \quad VP[\text{num: pl}]$

(also  $S \not\Rightarrow^* \text{John sleep}$ )

$\left[ \begin{array}{l} S \\ NP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \end{array} \right] \\ VP \quad \left[ \begin{array}{l} \text{num } \boxed{1} \end{array} \right] \end{array} \right] \sqcup [NP [\text{num sg}]] \sqcup [VP [\text{num pl}]] \rightarrow \text{fail}$

# TAG mit Features

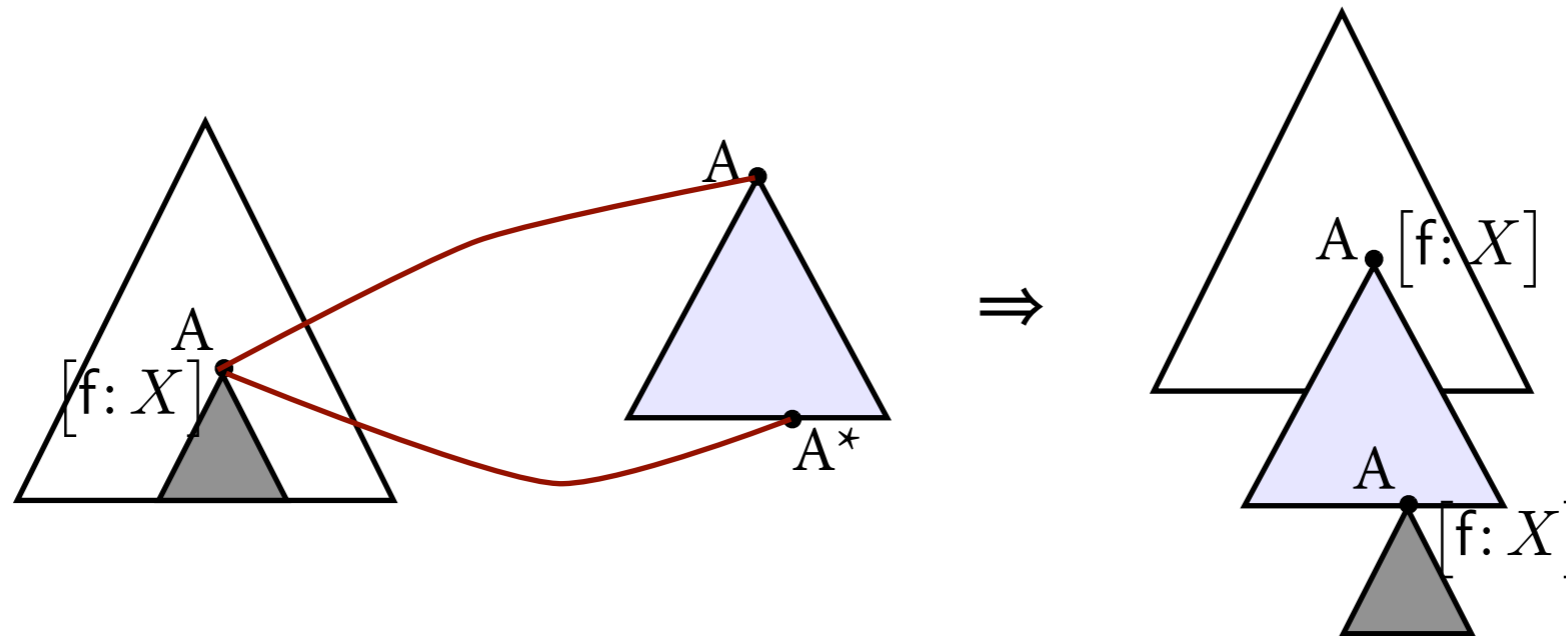
- Erste Idee: Knoten von Elementarbäumen mit Feature-Strukturen dekorieren.



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:  $FS(\text{Wurzel}) = FS(\text{Fußknoten})$ )



Bessere Intuition:

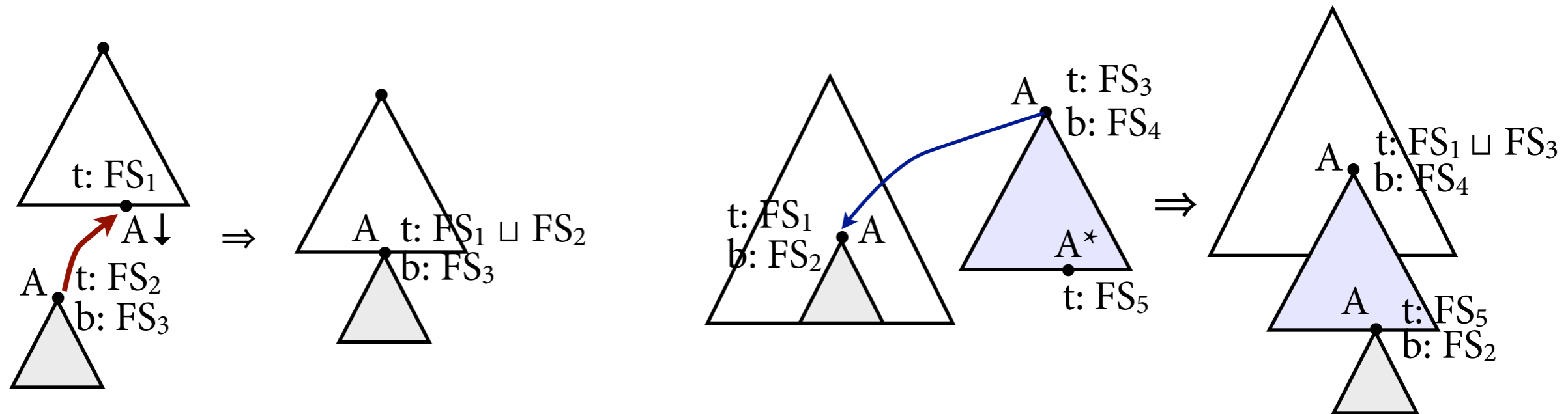
Adjunktion bricht einen Knoten in zwei Hälften auf.

# FTAG

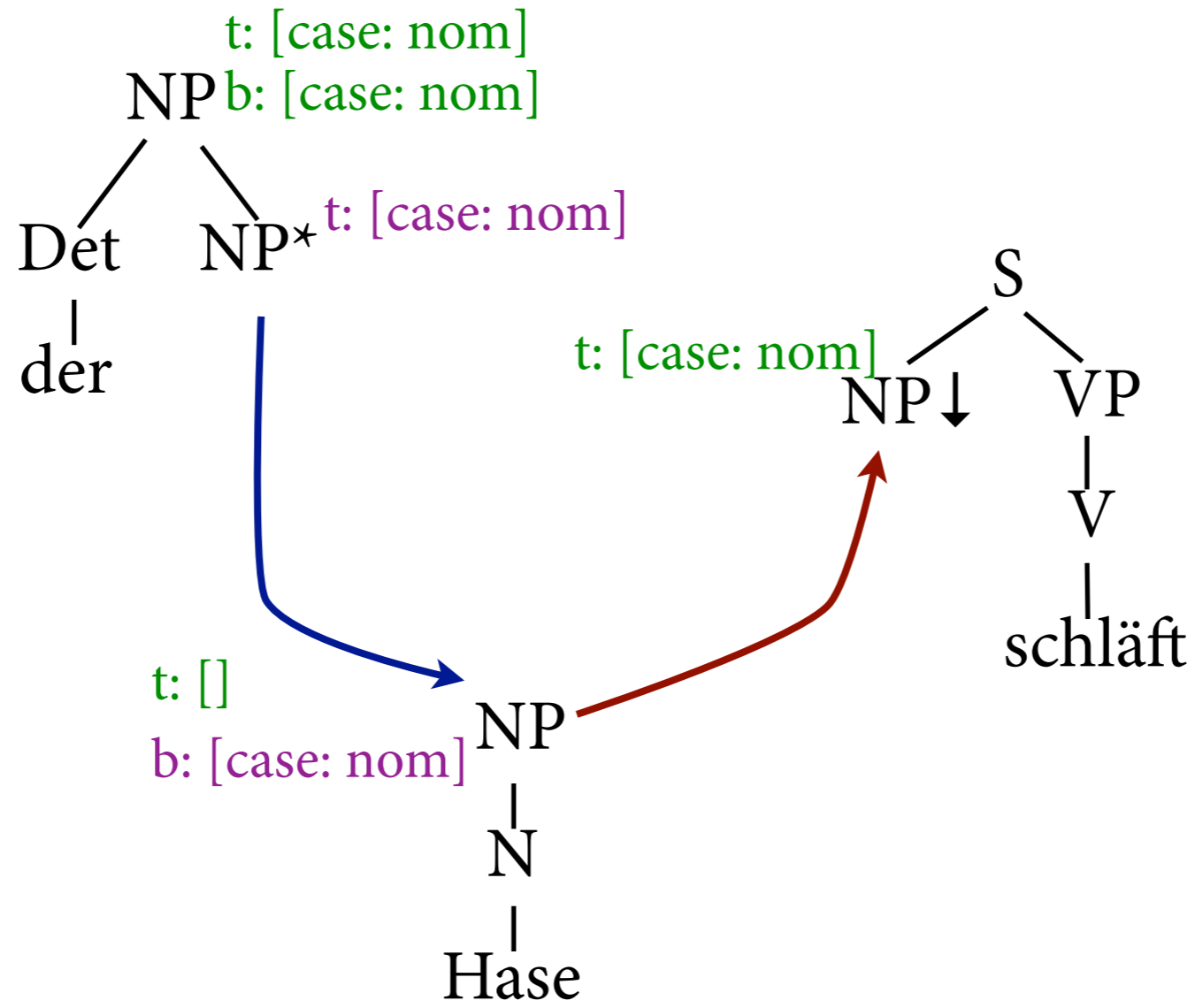
- In Feature-TAG trägt jeder Knoten *zwei* FSen: für obere und untere Hälfte des Knotens.
  - ▶ Wo man nicht adjungieren kann (z.B. Fußknoten), reicht eine FS
  - ▶ Obere FS von Wurzeln, untere FS von Blättern normalerweise leer
  - ▶ innere Knoten haben obere & untere FS
- Am Ende der Ableitung werden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

# Operationen von FTAG

- Substitution von  $t$  in  $u$ :
  - ▶ unifiziere obere FS von  $u$  mit FS der Wurzel von  $t$
- Adjunktion von  $t$  in  $u$ :
  - ▶ unifiziere obere FS von  $u$  mit FS der Wurzel von  $t$
  - ▶ unifizieren untere FS von  $u$  mit FS des Fußknotens von  $t$

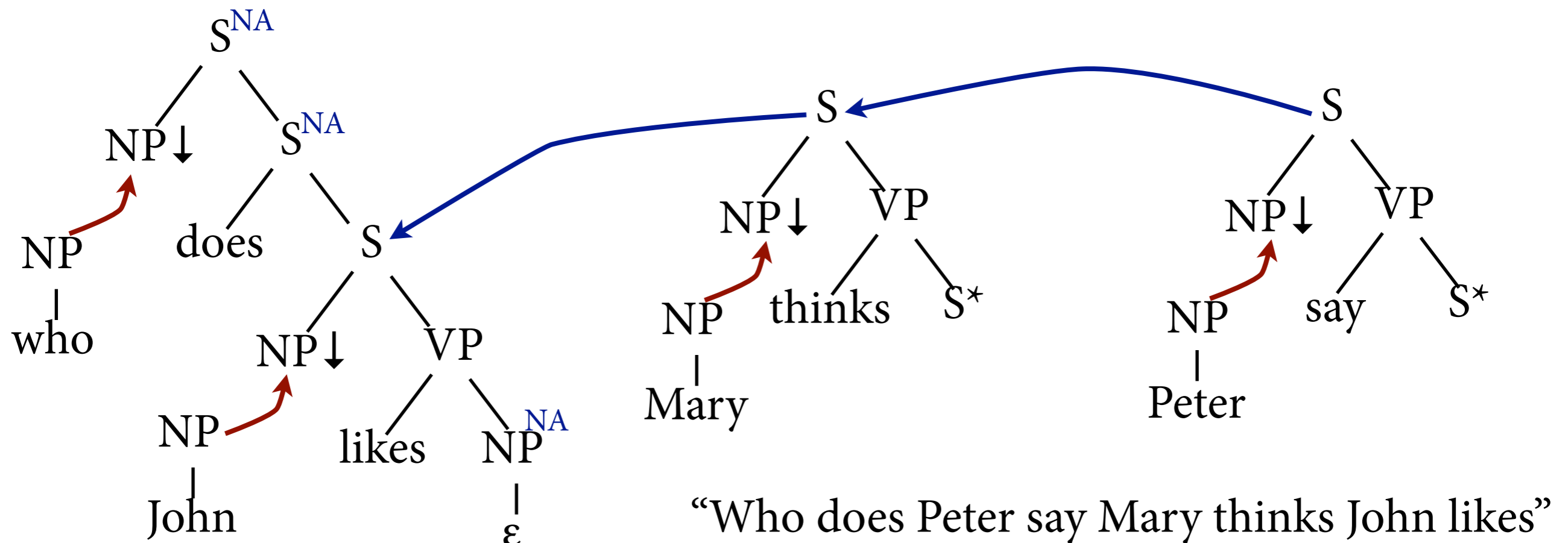


# Beispiel



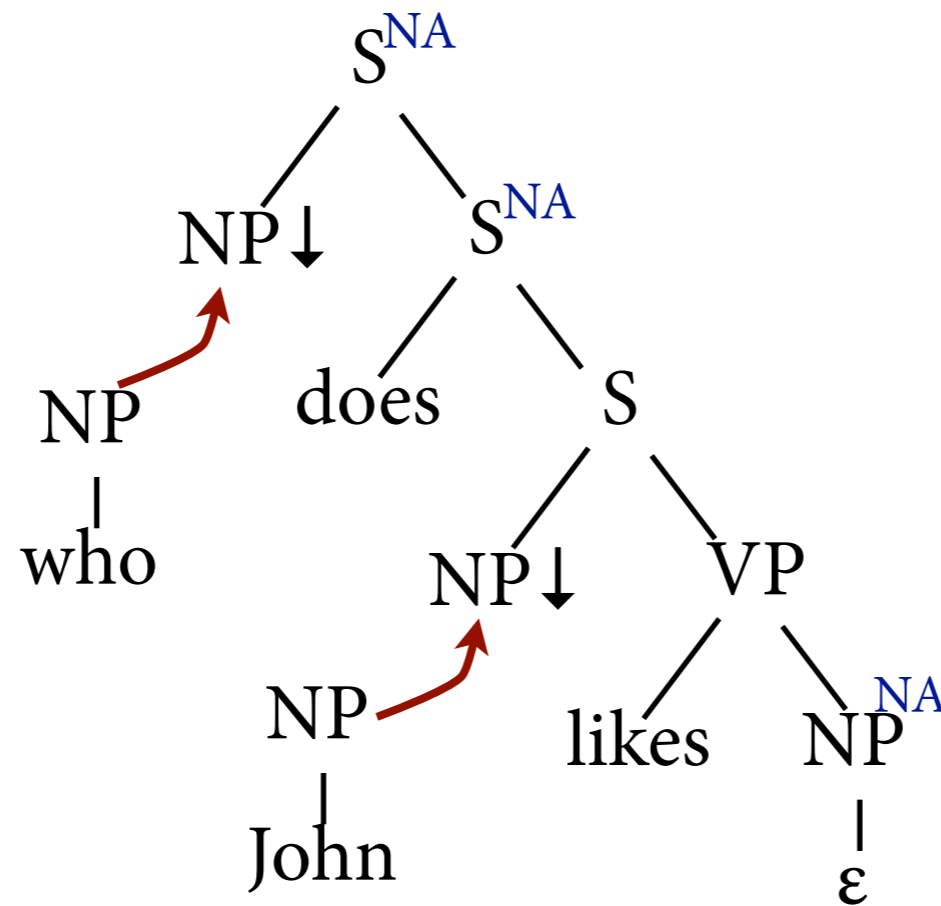
# Fernabhängigkeiten

- 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!



# Kontrolle von Fernabhängigkeiten

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

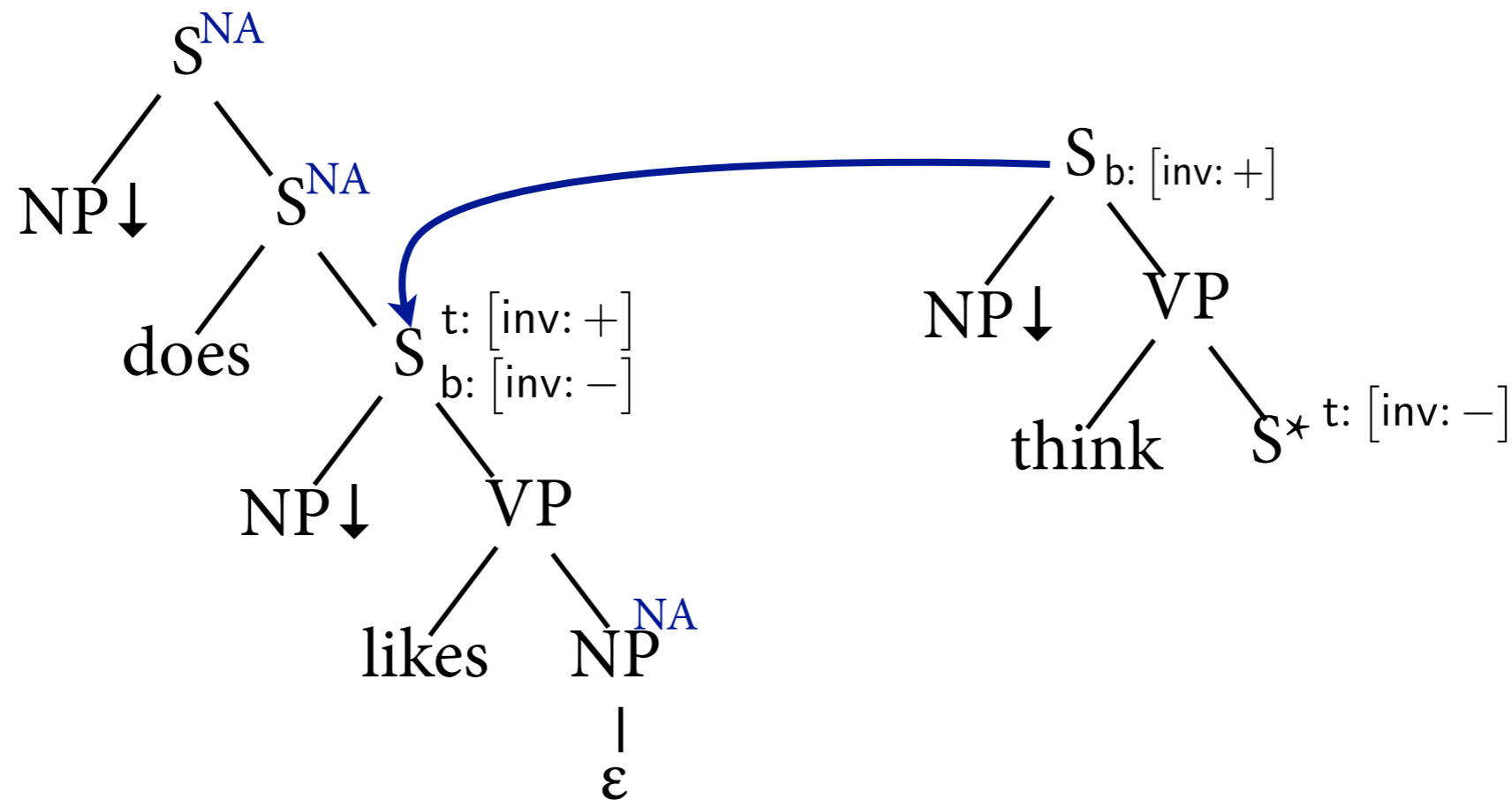


- Wie erzwingen wir, dass ein einbettendes Verb adjungiert werden *muss*?



# Kontrolle von Fernabhängigkeiten

- Lösung: inkompatible Features werden durch Adjunktion aufgelöst.



# Zusammenfassung

- Featuregrammatiken: Erlaubt 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.