

Reguläre Abhängigkeitsgrammatiken

Vorlesung “Grammatikformalismen”
Alexander Koller

28. Mai 2019

Reguläre Baumgrammatiken

- Reguläre Baumgrammatik (RTG) ist ein Tupel $G = (\Sigma, N, S, P)$, wobei
 - ▶ Σ eine Signatur (= Terminalsymbole)
 - ▶ N eine endliche Menge von Nichtterminalsymbolen
 - ▶ $S \in N$ das Startsymbol
 - ▶ P eine Menge von Produktionsregeln von der Form $A \rightarrow f(A_1, \dots, A_n)$, wobei $f \in \Sigma$ und $A, A_1, \dots, A_n \in N$.
- Eine RTG G definiert eine *Baumsprache* $L(G) \subseteq T_\Sigma$.

Ableitungen von RTGs

- Ableitungsprozess:
 - ▶ mit Startsymbol anfangen
 - ▶ in jedem Schritt ein Nichtterminalsymbol durch Baum auf der rechten Seite einer Regel ersetzen
 - ▶ wenn der Baum nur noch Terminalsymbole enthält, kommt er in die Sprache.

$S \rightarrow f(A,S)$

$S \rightarrow c$

$A \rightarrow a$

$A \rightarrow b$

RTG G

Ableitungen von RTGs

- Ableitungsprozess:
 - ▶ mit Startsymbol anfangen
 - ▶ in jedem Schritt ein Nichtterminalsymbol durch Baum auf der rechten Seite einer Regel ersetzen
 - ▶ wenn der Baum nur noch Terminalsymbole enthält, kommt er in die Sprache.

$S \rightarrow f(A,S)$

$S \rightarrow c$

$A \rightarrow a$

$A \rightarrow b$

$\bullet S \longrightarrow \bullet c$

RTG G

Ableitungen von RTGs

- Ableitungsprozess:
 - ▶ mit Startsymbol anfangen
 - ▶ in jedem Schritt ein Nichtterminalsymbol durch Baum auf der rechten Seite einer Regel ersetzen
 - ▶ wenn der Baum nur noch Terminalsymbole enthält, kommt er in die Sprache.

$S \rightarrow f(A,S)$

$S \rightarrow c$

$A \rightarrow a$

$A \rightarrow b$

$\bullet S \longrightarrow \boxed{\bullet c}$

RTG G

Ableitungen von RTGs

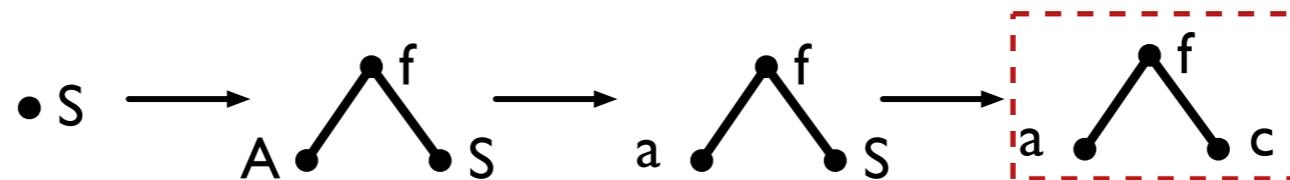
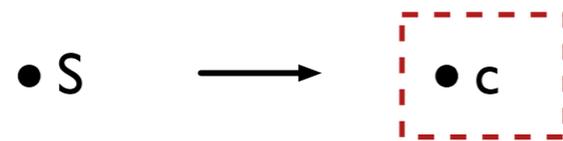
- Ableitungsprozess:
 - ▶ mit Startsymbol anfangen
 - ▶ in jedem Schritt ein Nichtterminalsymbol durch Baum auf der rechten Seite einer Regel ersetzen
 - ▶ wenn der Baum nur noch Terminalsymbole enthält, kommt er in die Sprache.

$S \rightarrow f(A,S)$

$S \rightarrow c$

$A \rightarrow a$

$A \rightarrow b$



RTG G

Ableitungen von RTGs

- Ableitungsprozess:

- ▶ mit Startsymbol anfangen
- ▶ in jedem Schritt ein Nichtterminalsymbol durch Baum auf der rechten Seite einer Regel ersetzen
- ▶ wenn der Baum nur noch Terminalsymbole enthält, kommt er in die Sprache.

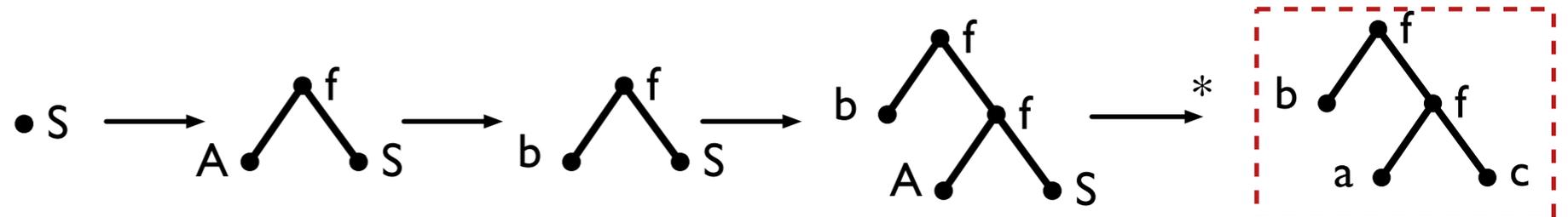
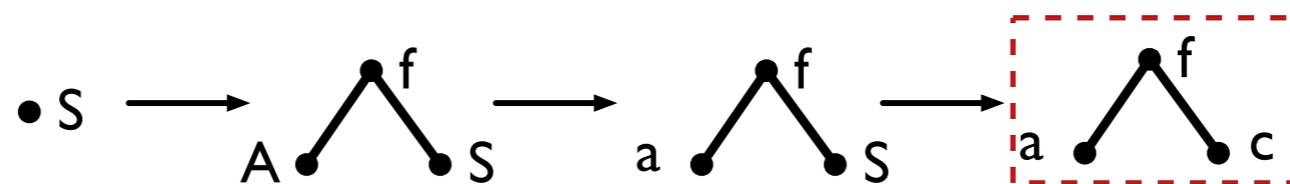
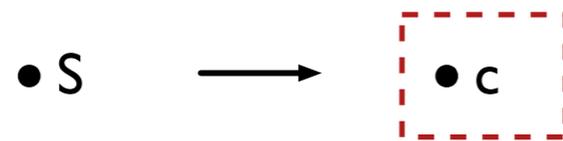
$S \rightarrow f(A,S)$

$S \rightarrow c$

$A \rightarrow a$

$A \rightarrow b$

RTG G



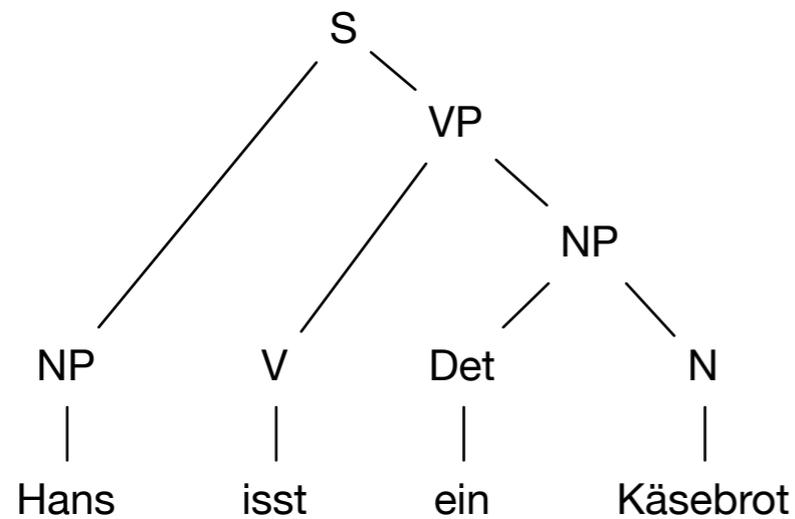
Reguläre Baumsprachen

- Eine Sprache L von Bäumen heißt *regulär*, wenn es eine RTG G gibt mit $L = L(G)$.
- Man kann reguläre Baumsprachen (RTLs) auch über *Baumautomaten* definieren.
- RTLs sind abgeschlossen unter Vereinigung, Schnitt, Komplement, usw.
- Sprache der Parsebäume einer kfG ist immer regulär.
Sprache der Erträge einer RTL ist immer kontextfrei.

kfG \leftrightarrow RTG

$S \rightarrow NP VP$ $V \rightarrow isst$
 $NP \rightarrow Det N$ $NP \rightarrow Hans$
 $VP \rightarrow V NP$ $Det \rightarrow ein$
 $N \rightarrow Käsebrot$

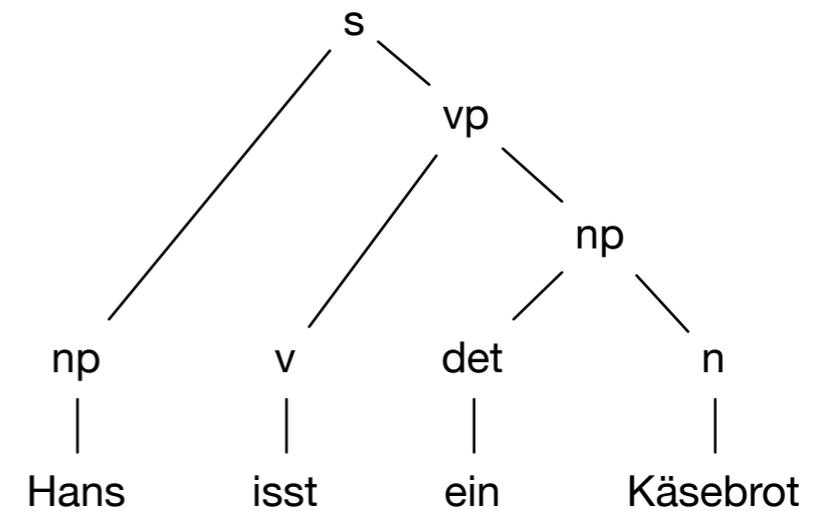
kontextfreie Grammatik G_1



“Hans isst ein Käsebrot”
 $\in L(G_1)$
(Sprache enthält Strings)

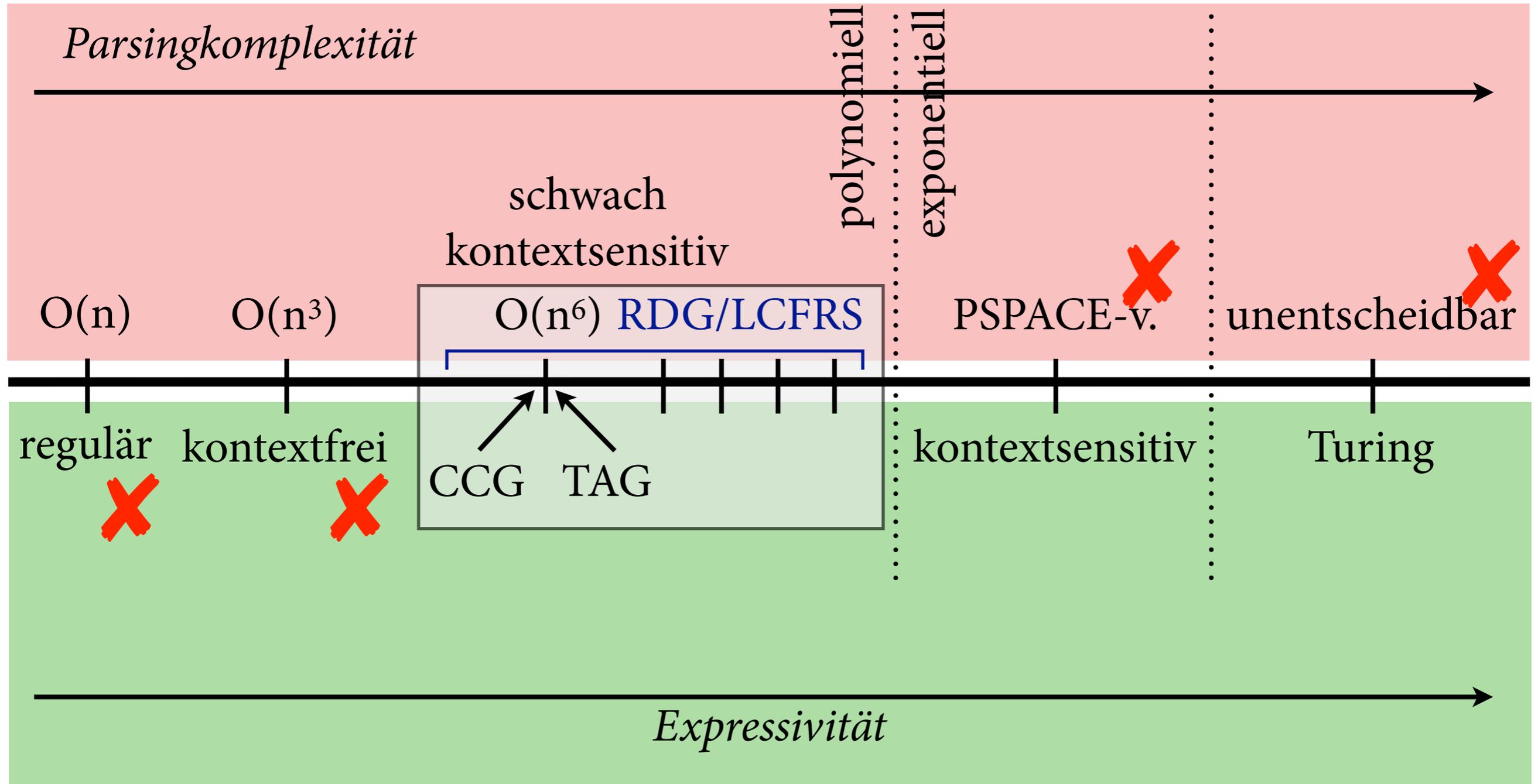
$S \rightarrow s(NP, VP)$ $V \rightarrow v(isst)$
 $NP \rightarrow np(Det, N)$ $NP \rightarrow np(Hans)$
 $VP \rightarrow vp(V, NP)$ $Det \rightarrow det(ein)$
 $N \rightarrow n(Käsebrot)$

RTG G_2



$\in L(G_2)$
(Sprache enthält Bäume)

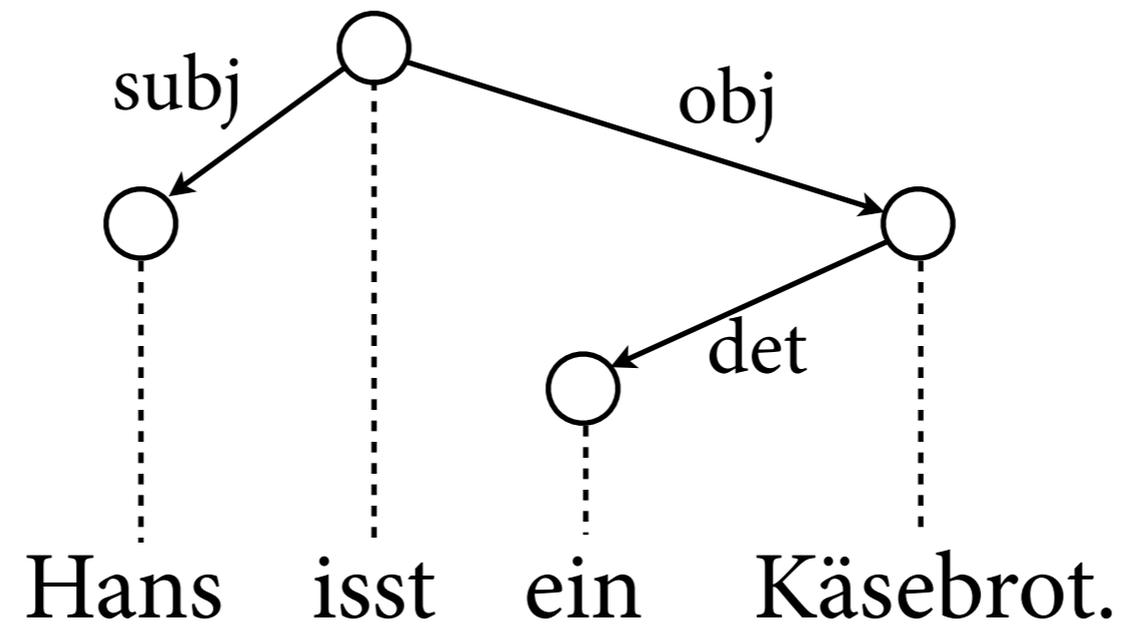
Eine allgemeinere Perspektive



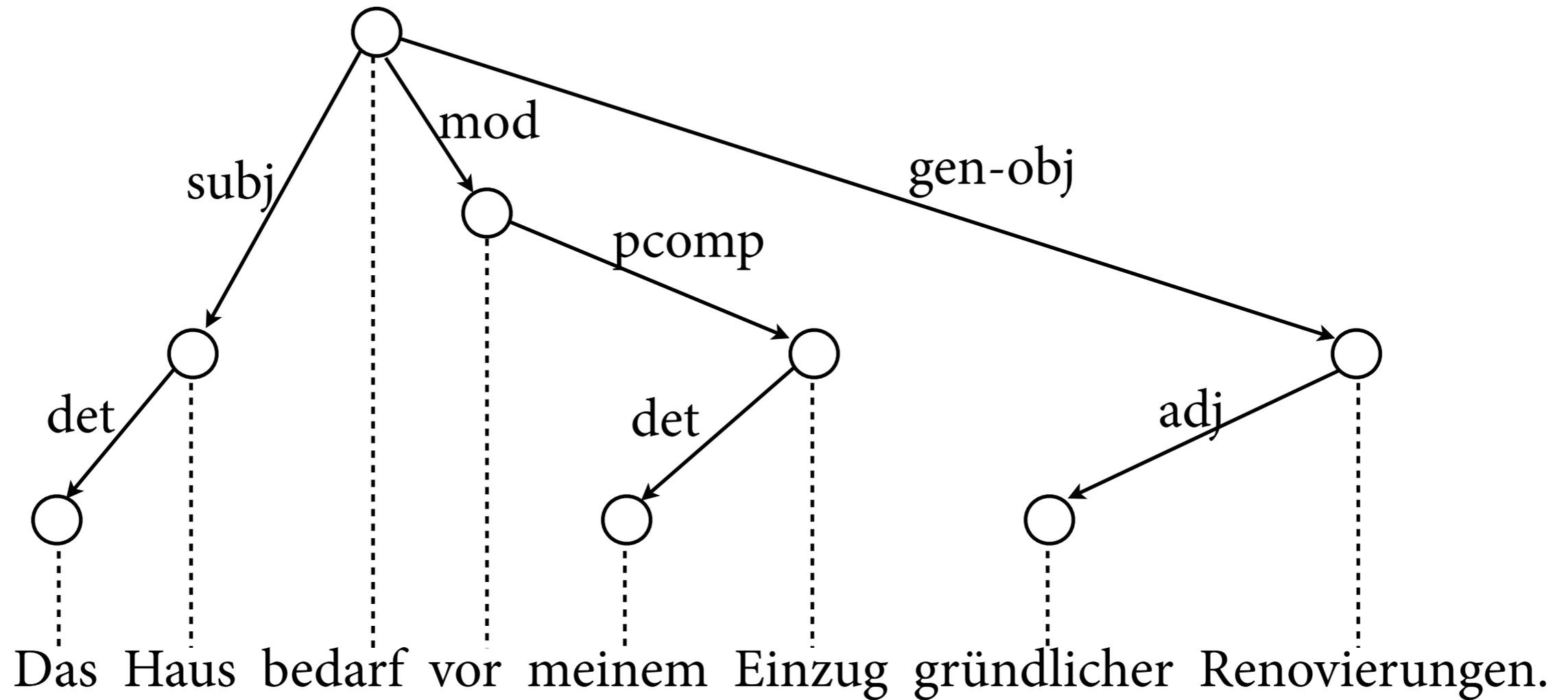
Abhängigkeitsgrammatiken

- Grammatiktheorie: Syntaktische Strukturen = *Abhängigkeiten* zwischen Wörtern.
- Abhängigkeitsbaum:
 - ▶ genau ein Knoten für jedes Wort
 - ▶ Kante von u nach v : v füllt eine Valenz von u
 - ▶ manchmal gibt es *Kantenlabels*, die grammatische Rolle angeben

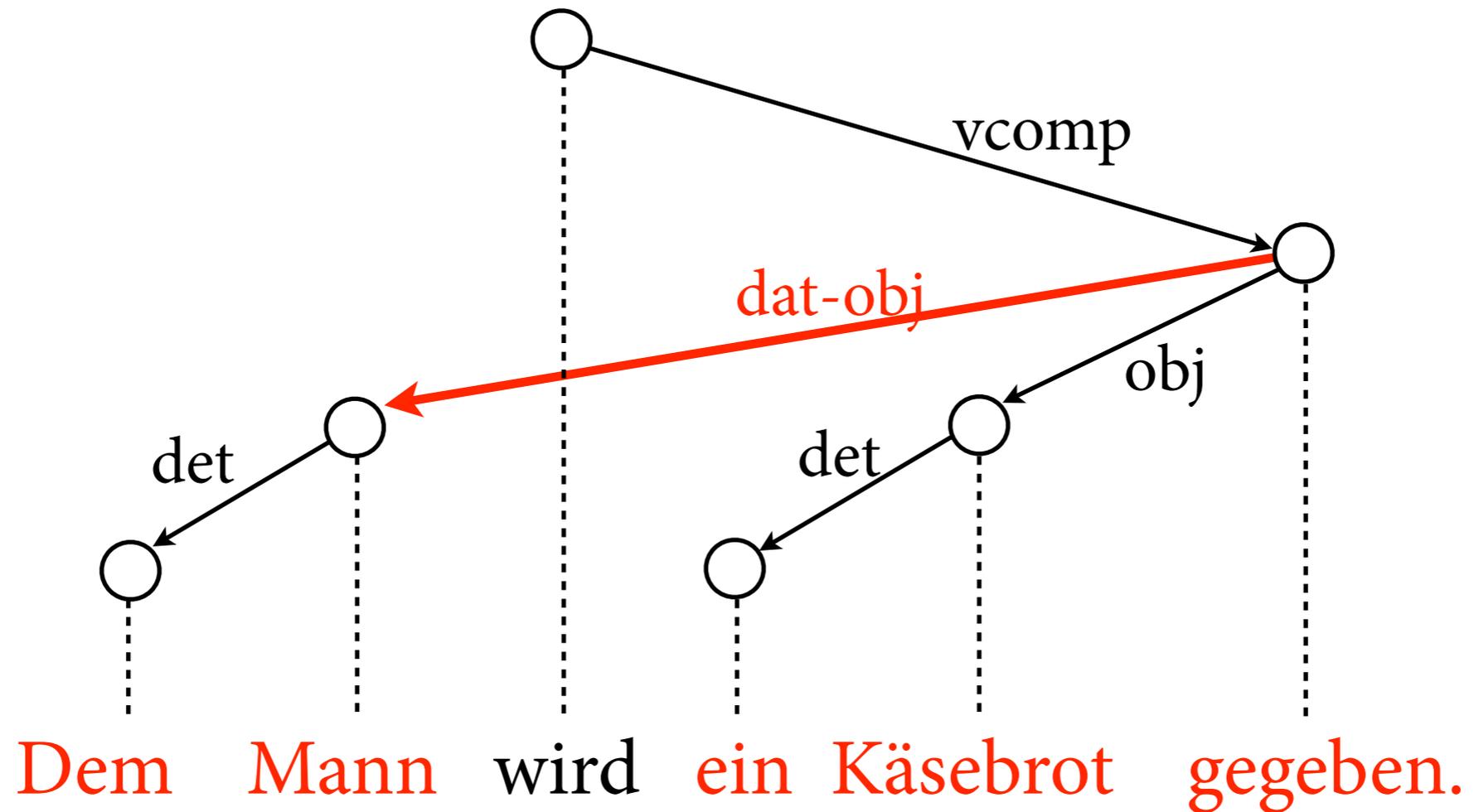
Abhängenkbäume: Beispiele



Dependenzbäume: Beispiele

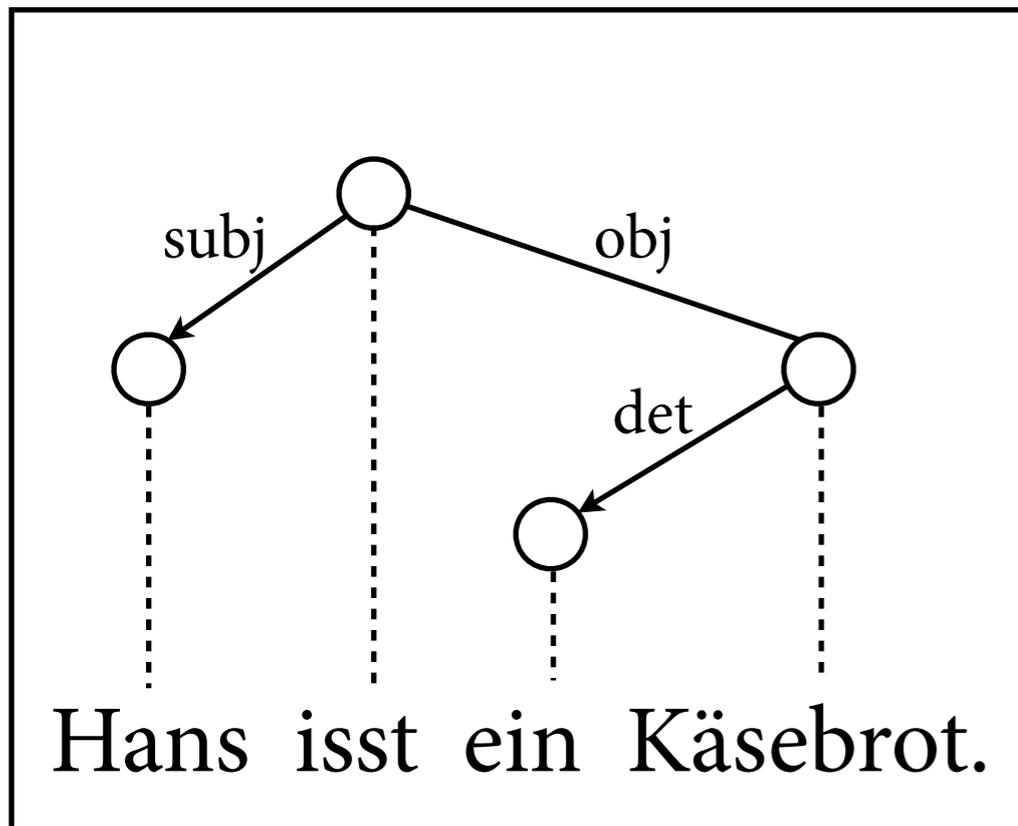


Abhängenkbäume: Beispiele

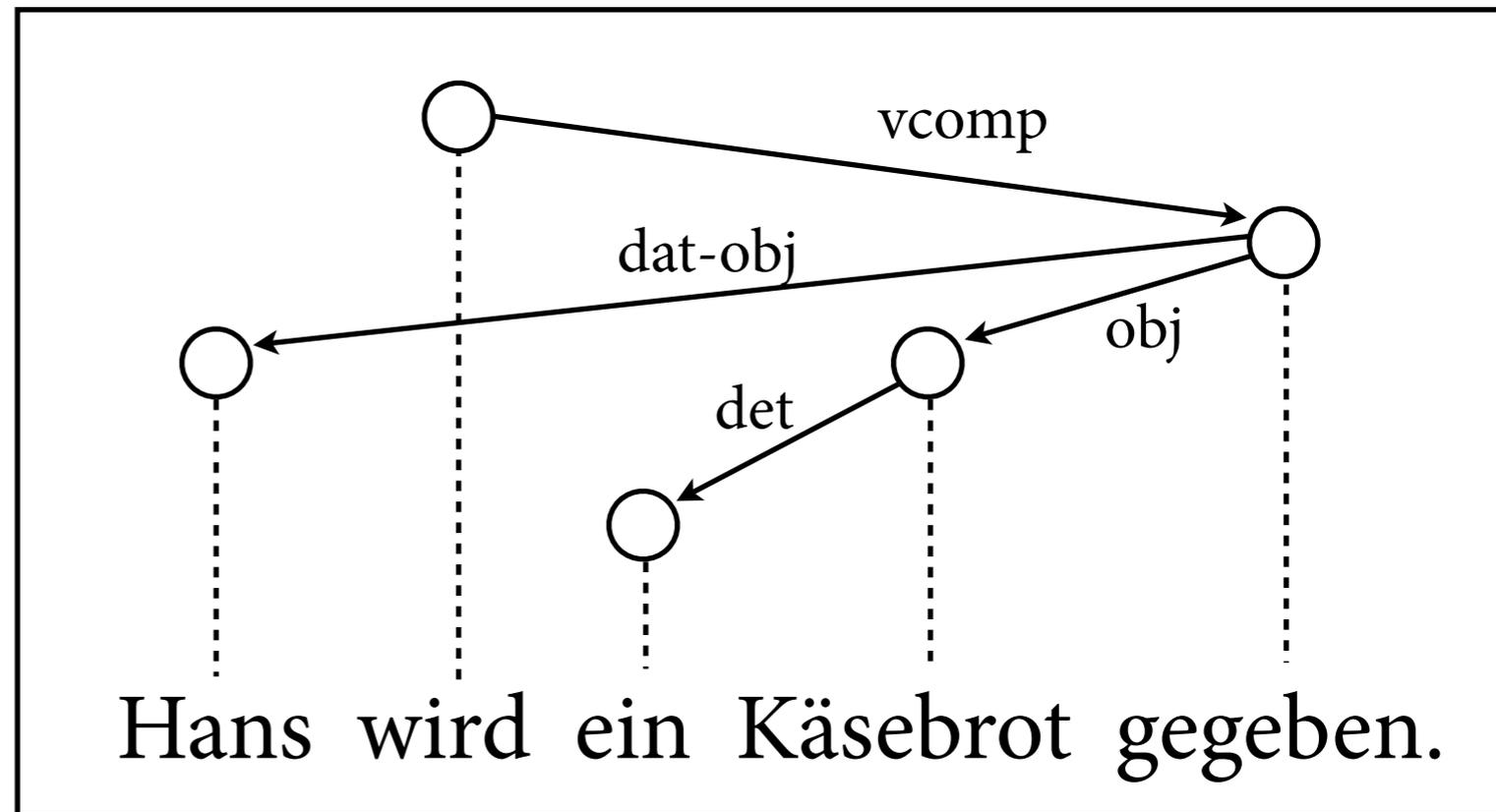


Projektivität

- Ein Dependenzbaum heißt *projektiv*, wenn sich keine Kante mit einer Projektionslinie schneidet.



projektiv



nichtprojektiv

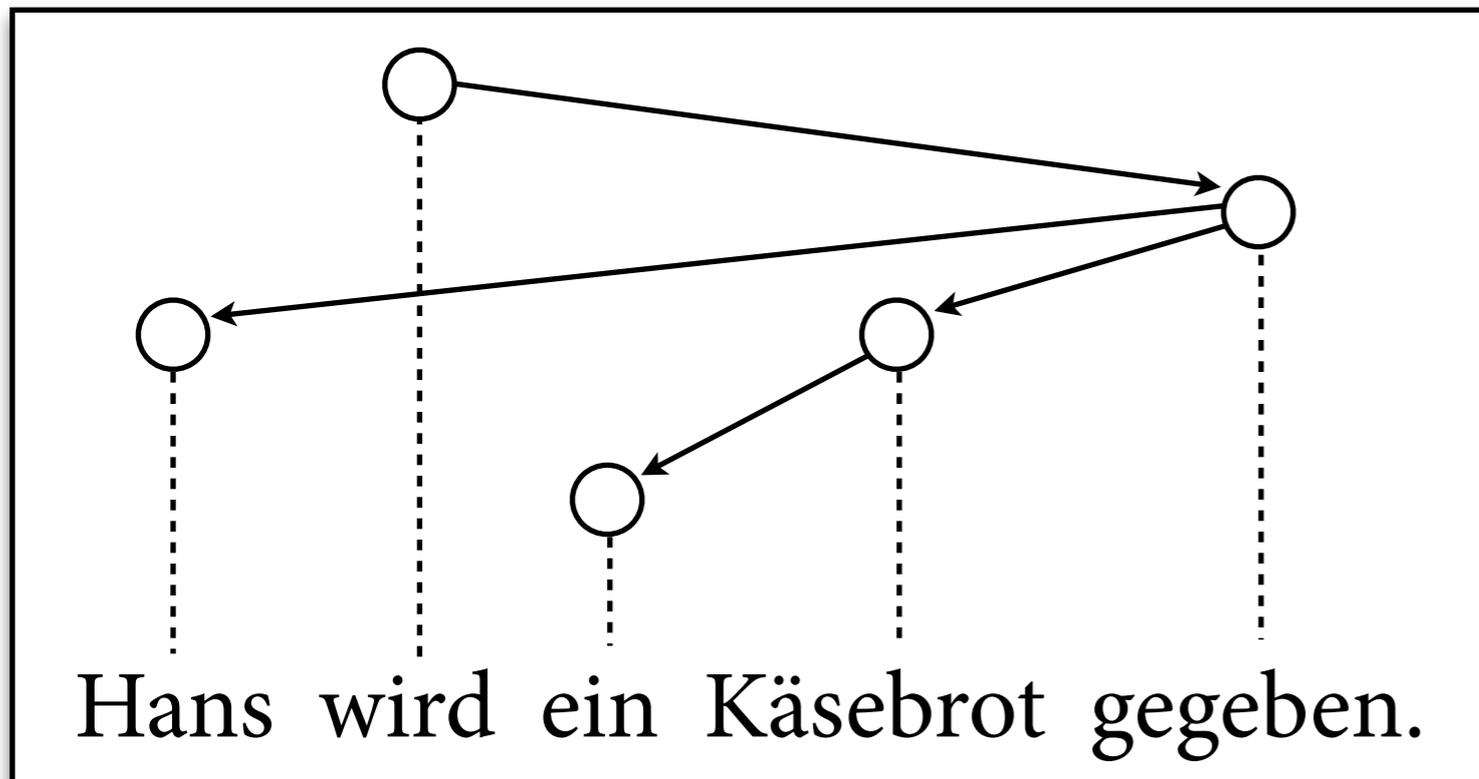
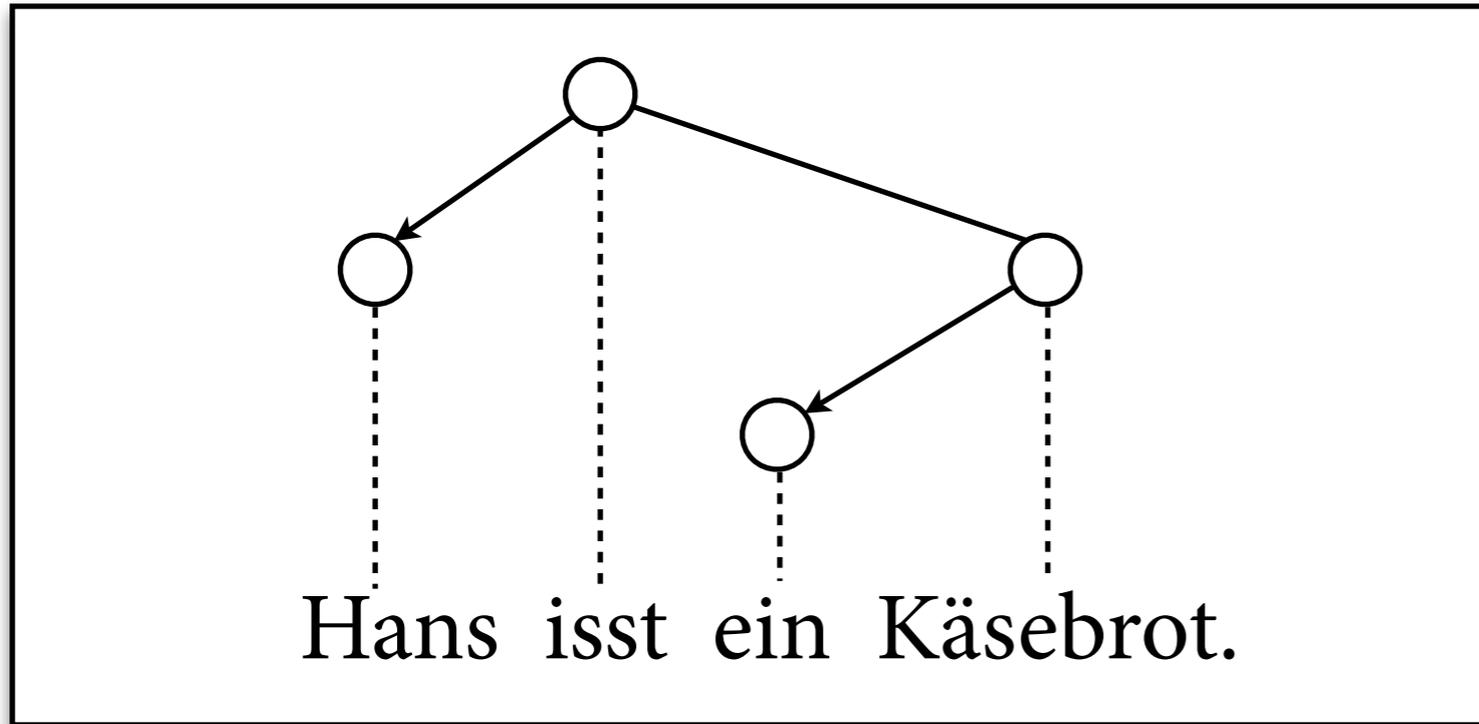
Grammatikformalismus

- Um einen Grammatik*formalismus* für Dependenzgrammatiken zu entwickeln, müssen wir einige Fragen beantworten:
 - ▶ Was genau ist formal ein Dependenzbaum? Sollte saubere Definition von Projektivität erlauben.
 - ▶ Wie sieht eine Grammatik aus, die Sprachen von Dependenzbäumen beschreiben kann?

Dependenzstruktur

- Def. *Dependenzstruktur* = Paar von *Baum* und *linearer Ordnung* auf den Knoten.
- Valenz wird beschrieben durch *Dominanzrelation* auf dem Baum.
- Wortstellung wird beschrieben durch *Präzedenzrelation* in der linearen Ordnung.

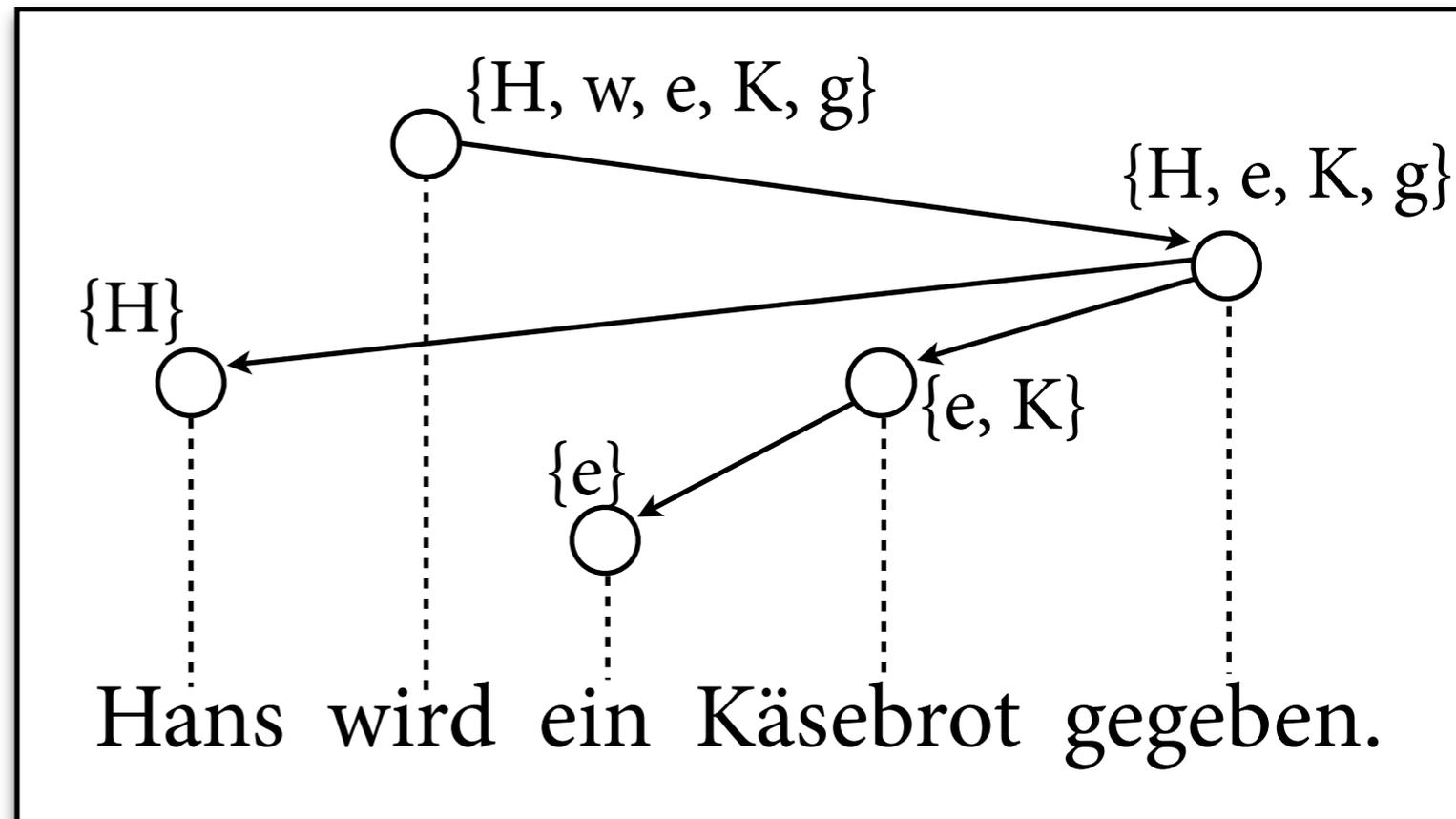
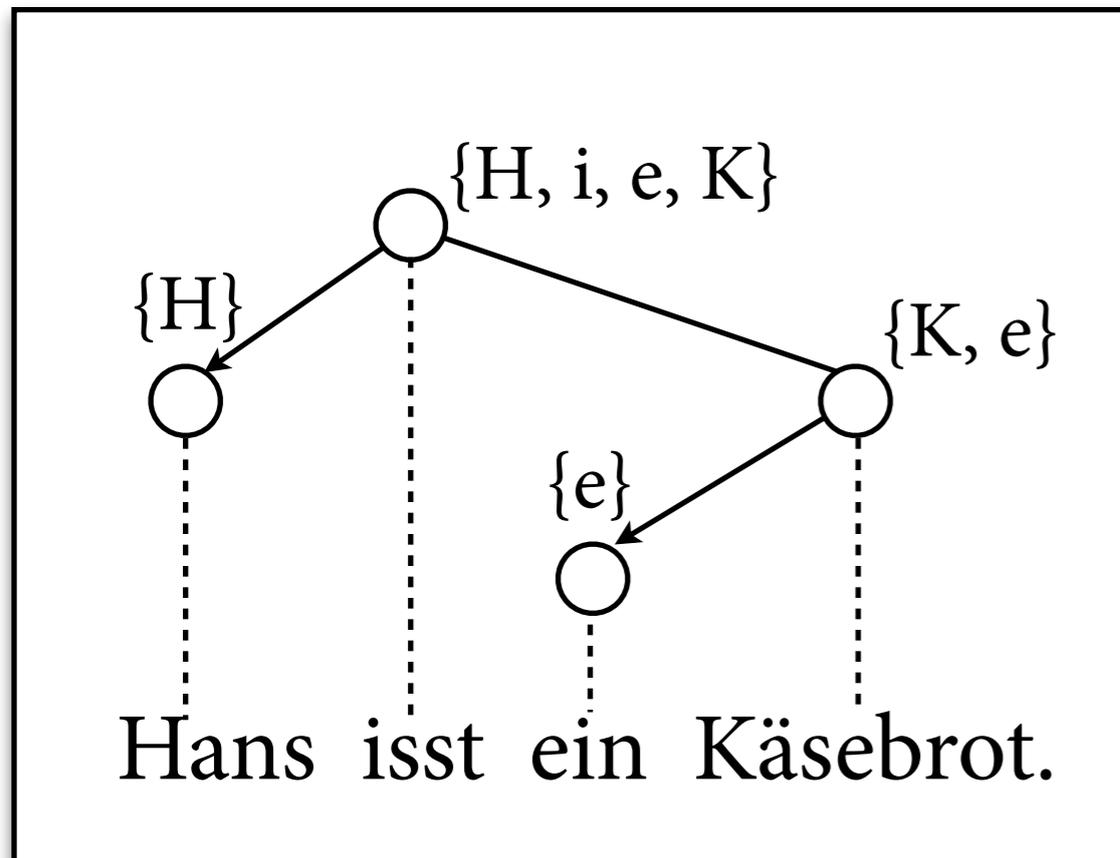
Beispiel



Dominanz	Präzedenz
$H \triangleleft^* H$ $i \triangleleft^* H, i, e, K$ $e \triangleleft^* e$ $K \triangleleft^* e, K$	$H < i, e, K$ $i < e, K$ $e < K$
$H \triangleleft^* H$ $w \triangleleft^* H, w, e, K, g$ $e \triangleleft^* e$ $K \triangleleft^* e, K$ $g \triangleleft^* e, K, g$	$H < w, e, K, g$ $w < e, K, g$ $e < K, g$ $K < g$

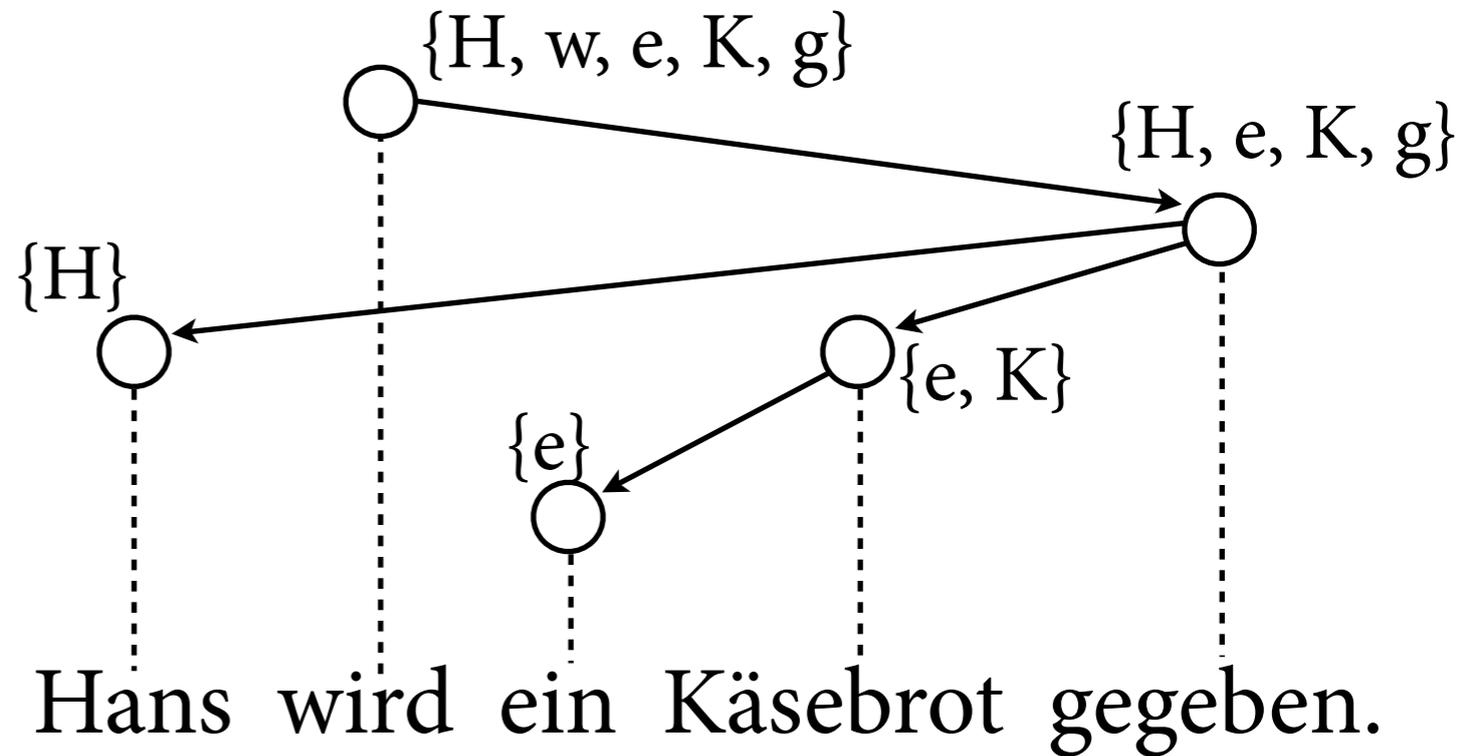
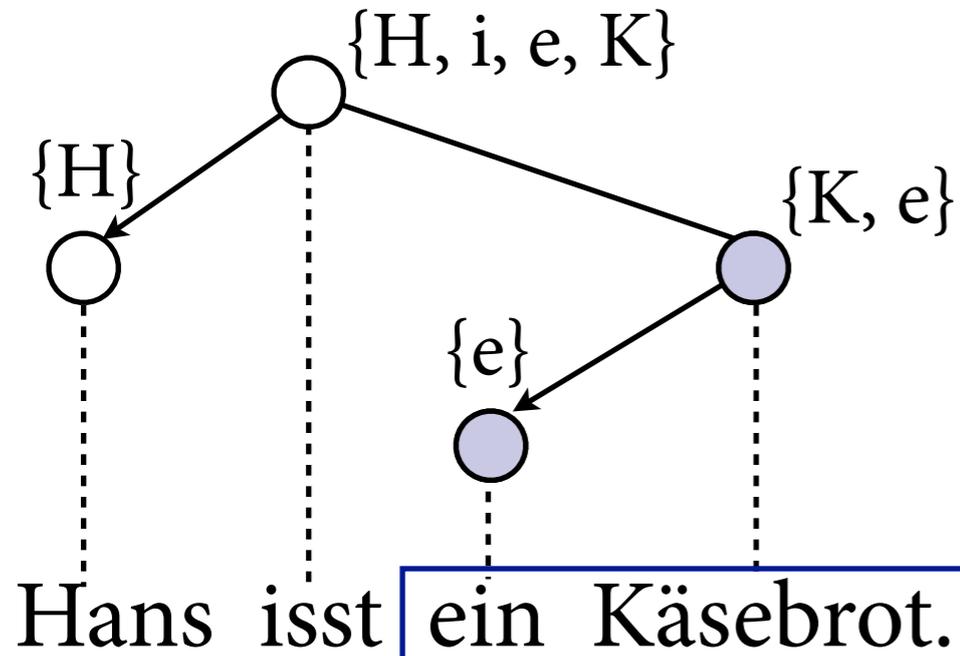
Ertrag

- *Ertrag* des Knotens u = Menge aller Knoten, die von u dominiert werden, in der Reihenfolge der Präzedenzrelation.



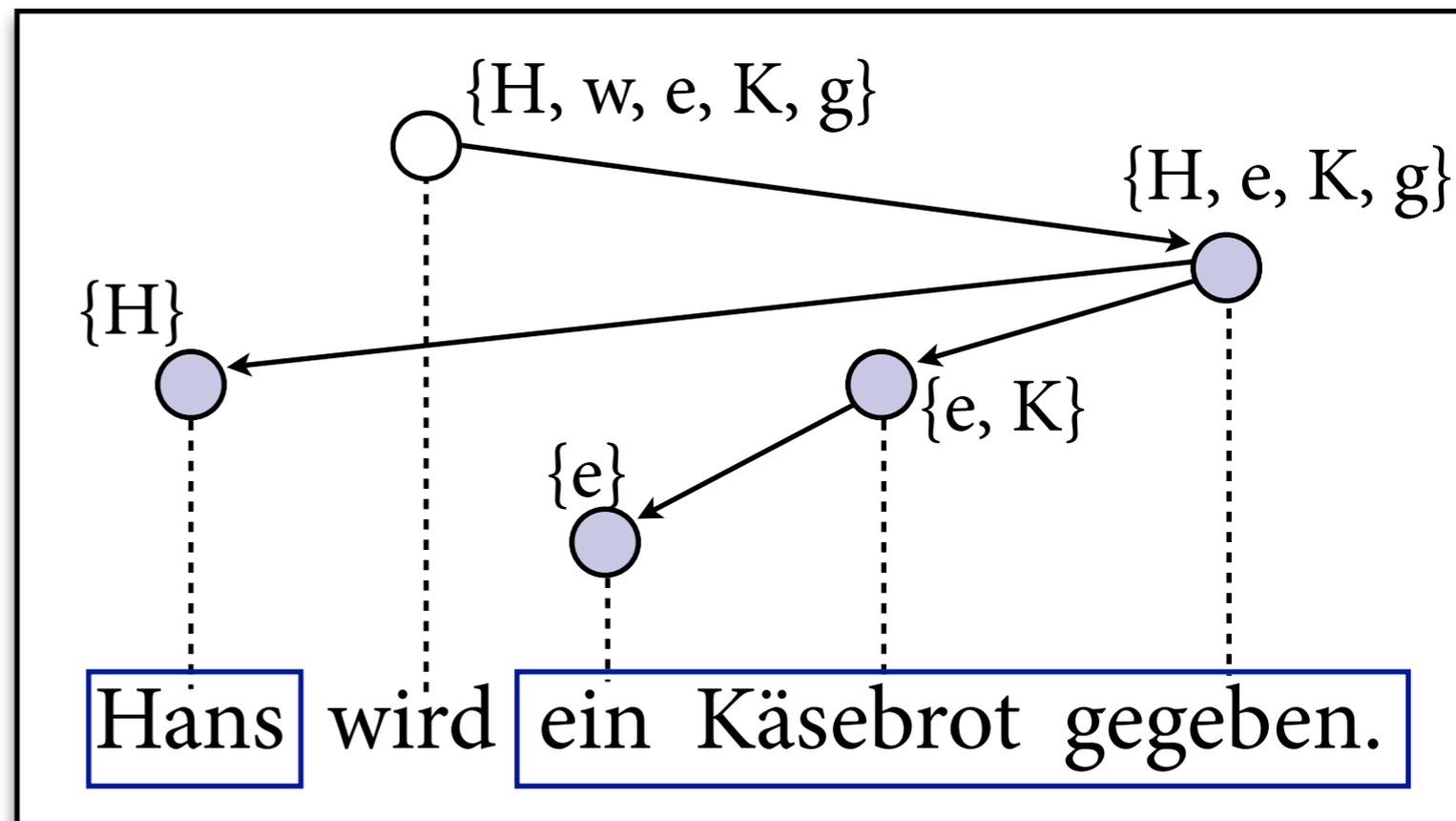
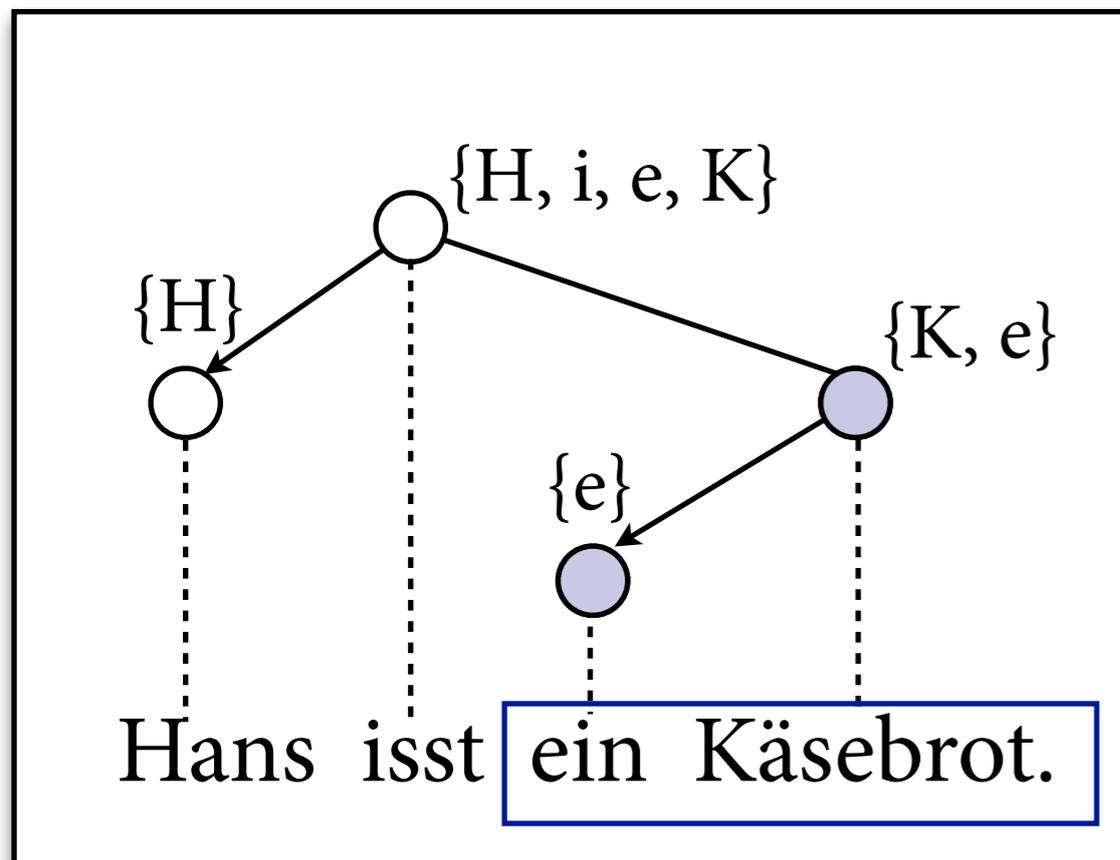
Ertrag

- *Ertrag* des Knotens u = Menge aller Knoten, die von u dominiert werden, in der Reihenfolge der Präzedenzrelation.



Ertrag

- *Ertrag* des Knotens u = Menge aller Knoten, die von u dominiert werden, in der Reihenfolge der Präzedenzrelation.



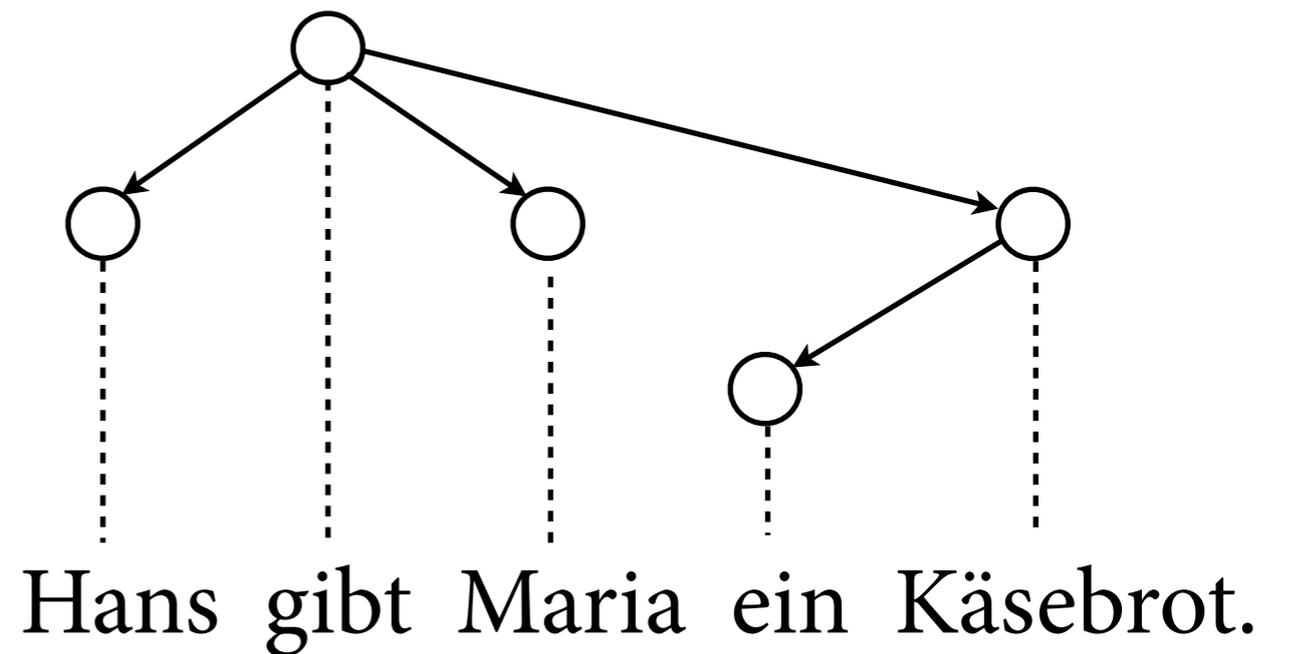
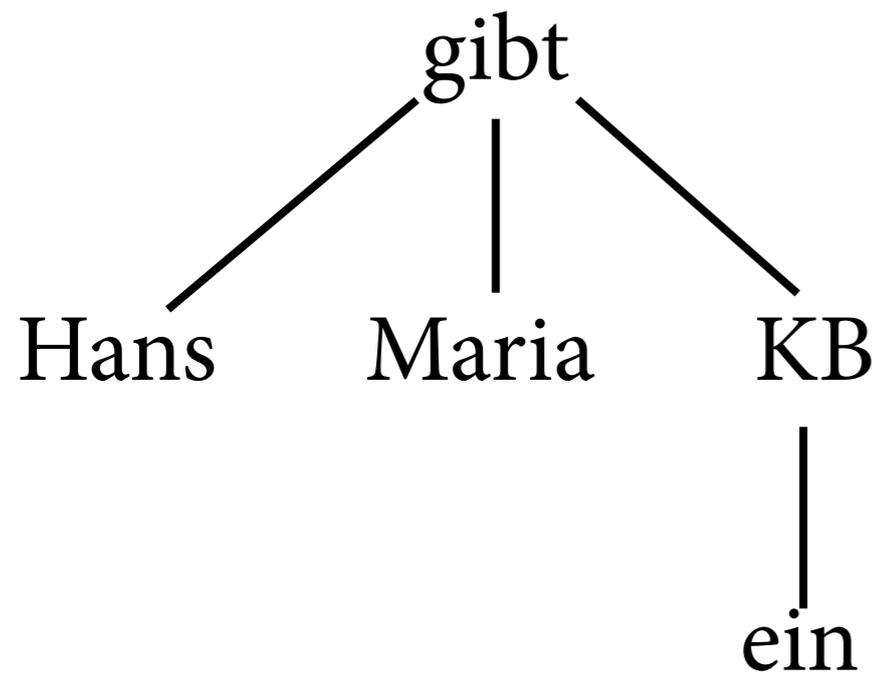
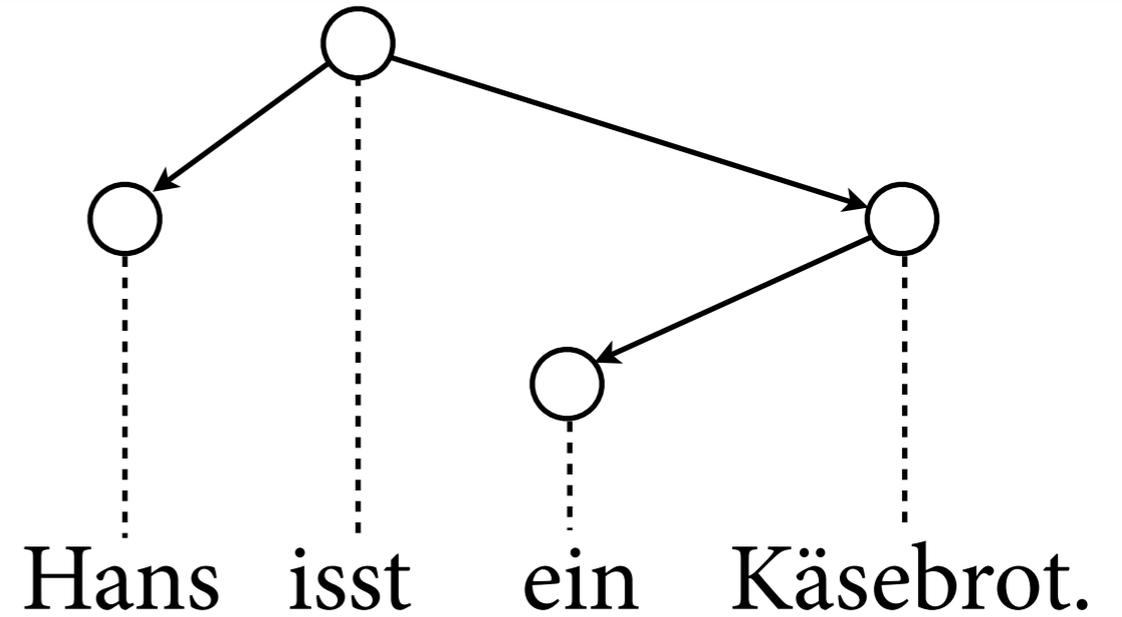
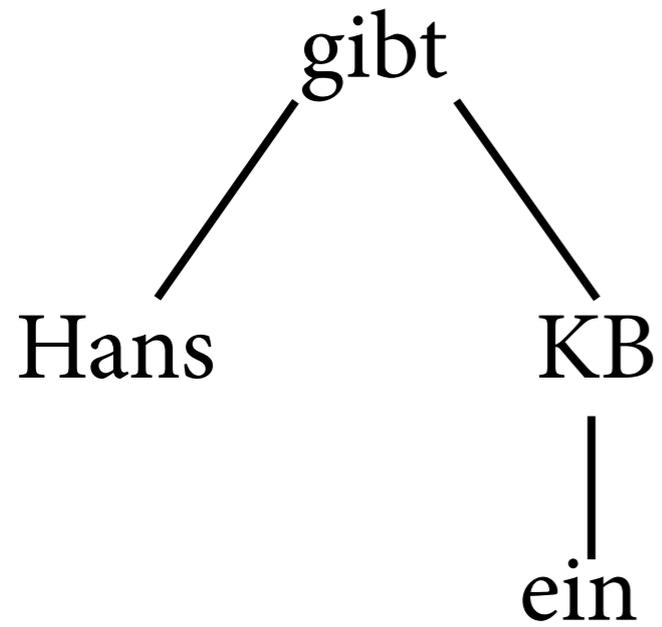
Projektivität

- *Block*: maximale Sequenz von Knoten, die in der Präzedenzrelation keine Lücke haben.
- *Blockgrad* eines Knotens u = Anzahl der zusammenhängenden Blöcke im Ertrag von u .
- Blockgrad eines Baums = maximaler Blockgrad der Knoten.
- Baum ist *projektiv* = hat Blockgrad 1.

Abhängigkeitsprachen

- Wir haben schon einen Grammatikformalismus, der Sprachen von Bäumen beschreibt: reguläre Baumgrammatiken.
- Abhängigkeitsstrukturen sind aber keine Bäume, sondern enthalten *globale Ordnungsinformationen*.
- Wie stellt man diese Ordnungsinformation am besten dar?

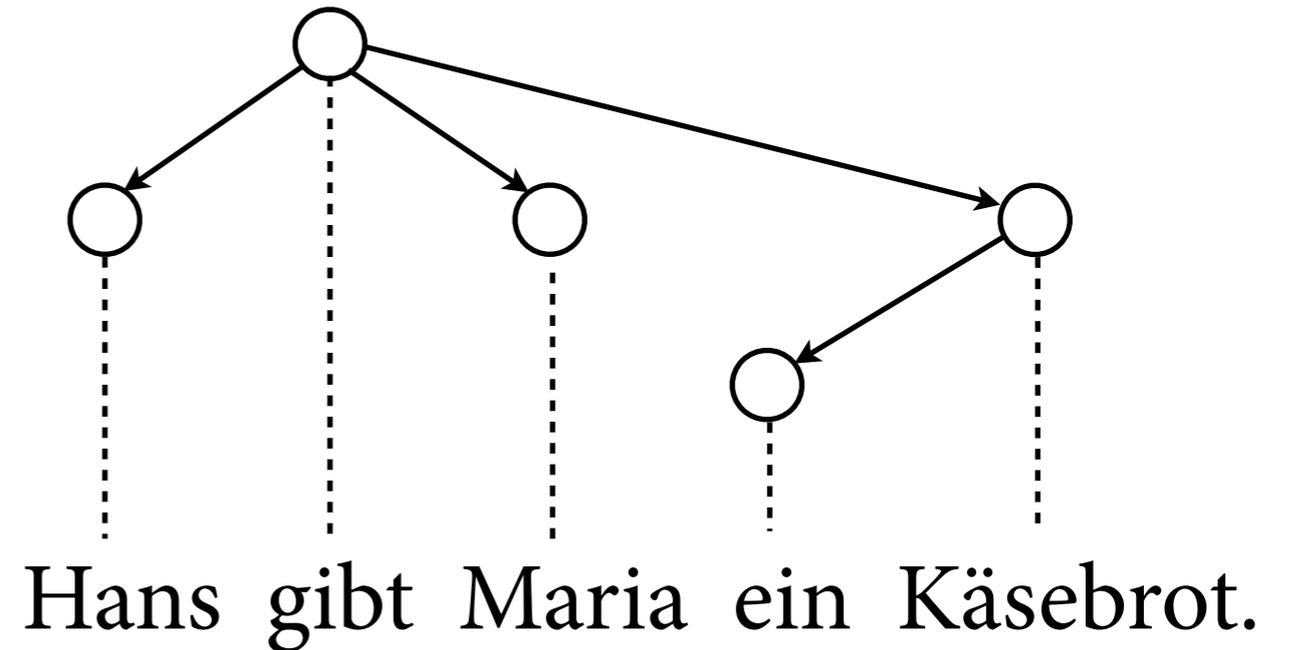
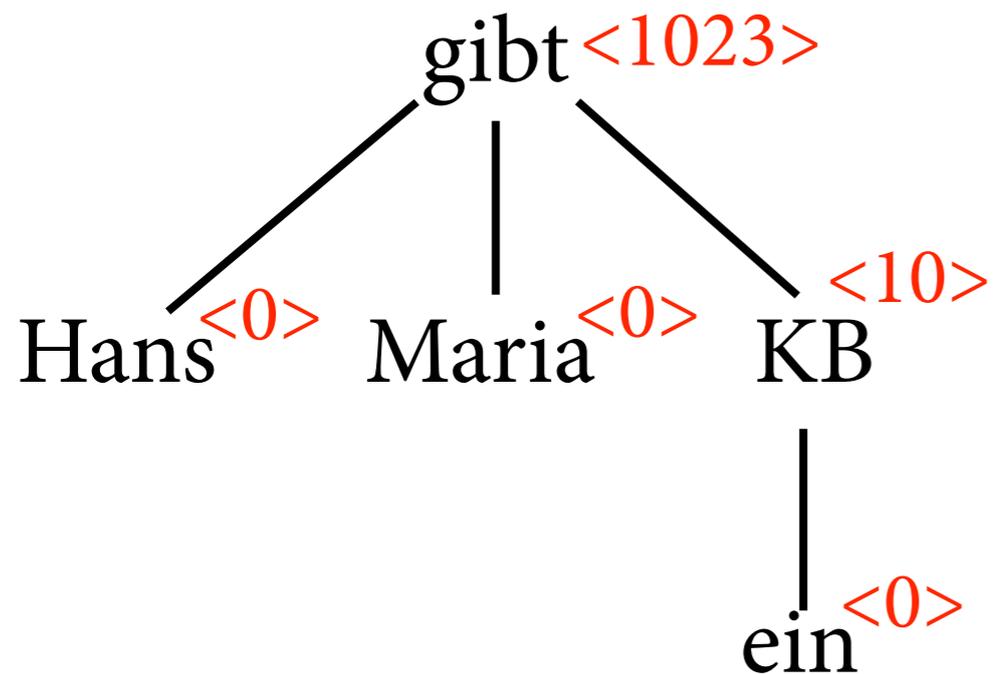
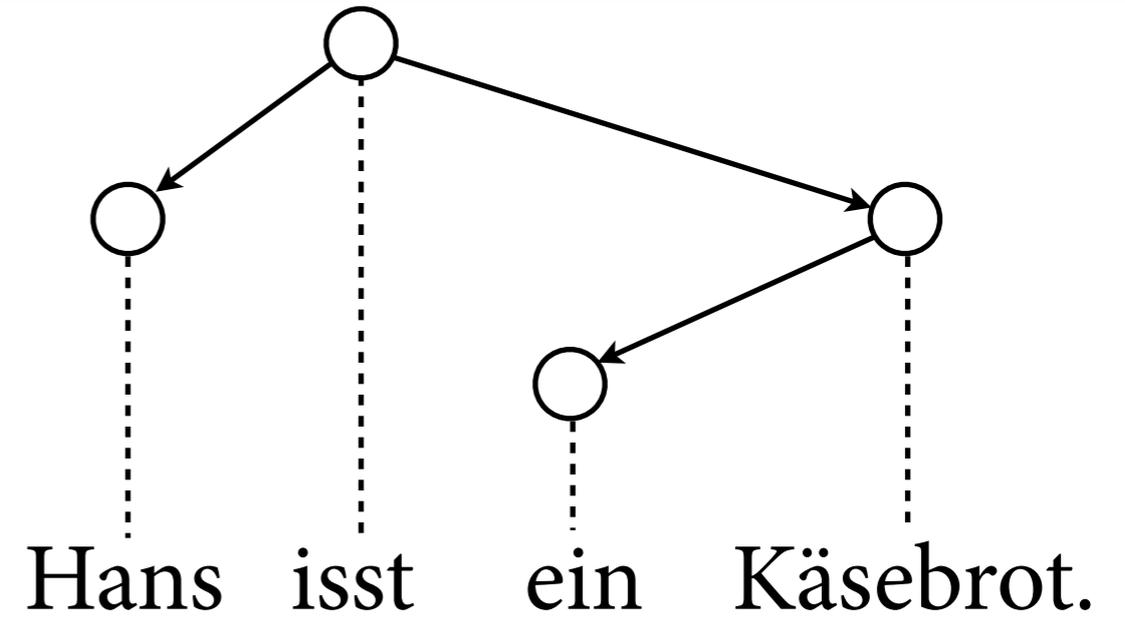
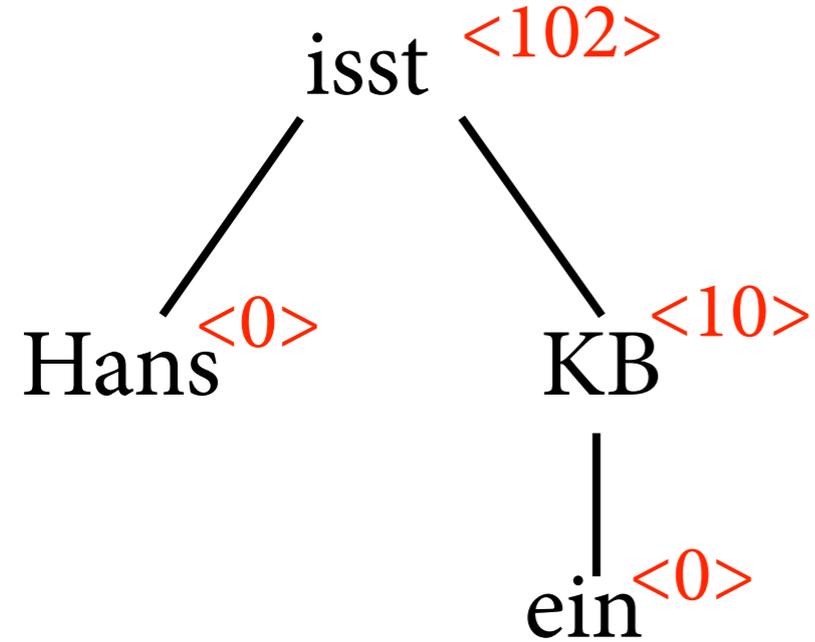
Global vs. lokal



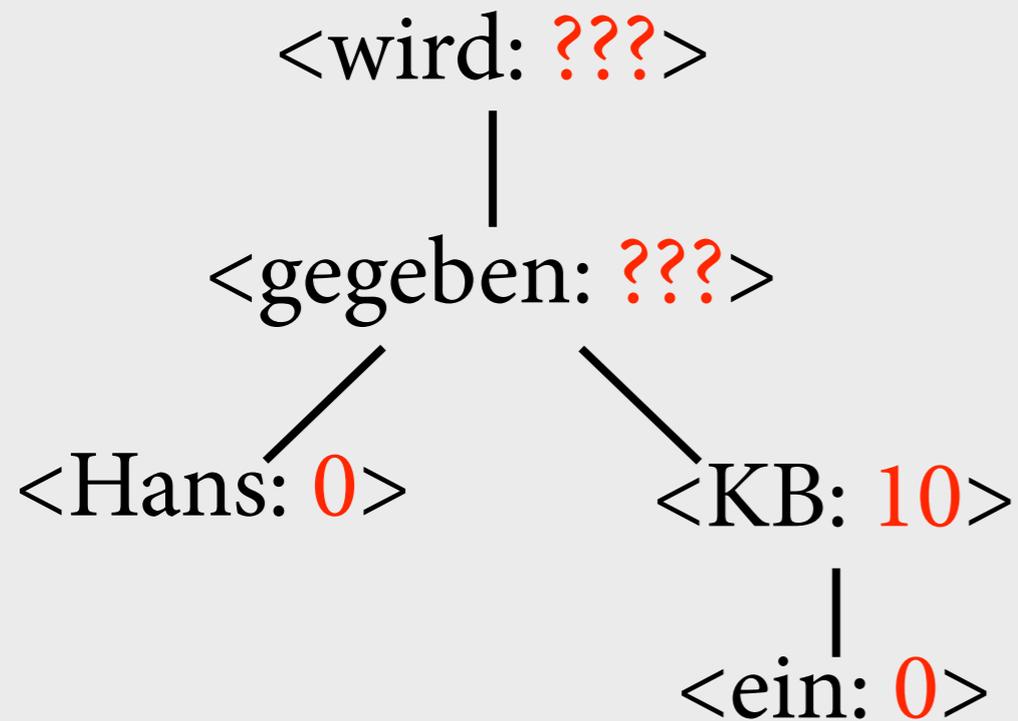
Ordnungsannotationen

- Annotiere jeden Knoten im Baum mit einer *Ordnungsannotation (OA)*, die angibt, wie Ertrag des Knotens sich aus Erträgen seiner Kinder zusammensetzt.
- Ordnungsannotation = String von Zahlen
 - ▶ 0: Knoten selbst
 - ▶ 1: Ertrag des ersten Kindes
 - ▶ 2: Ertrag des zweiten Kindes, usw.

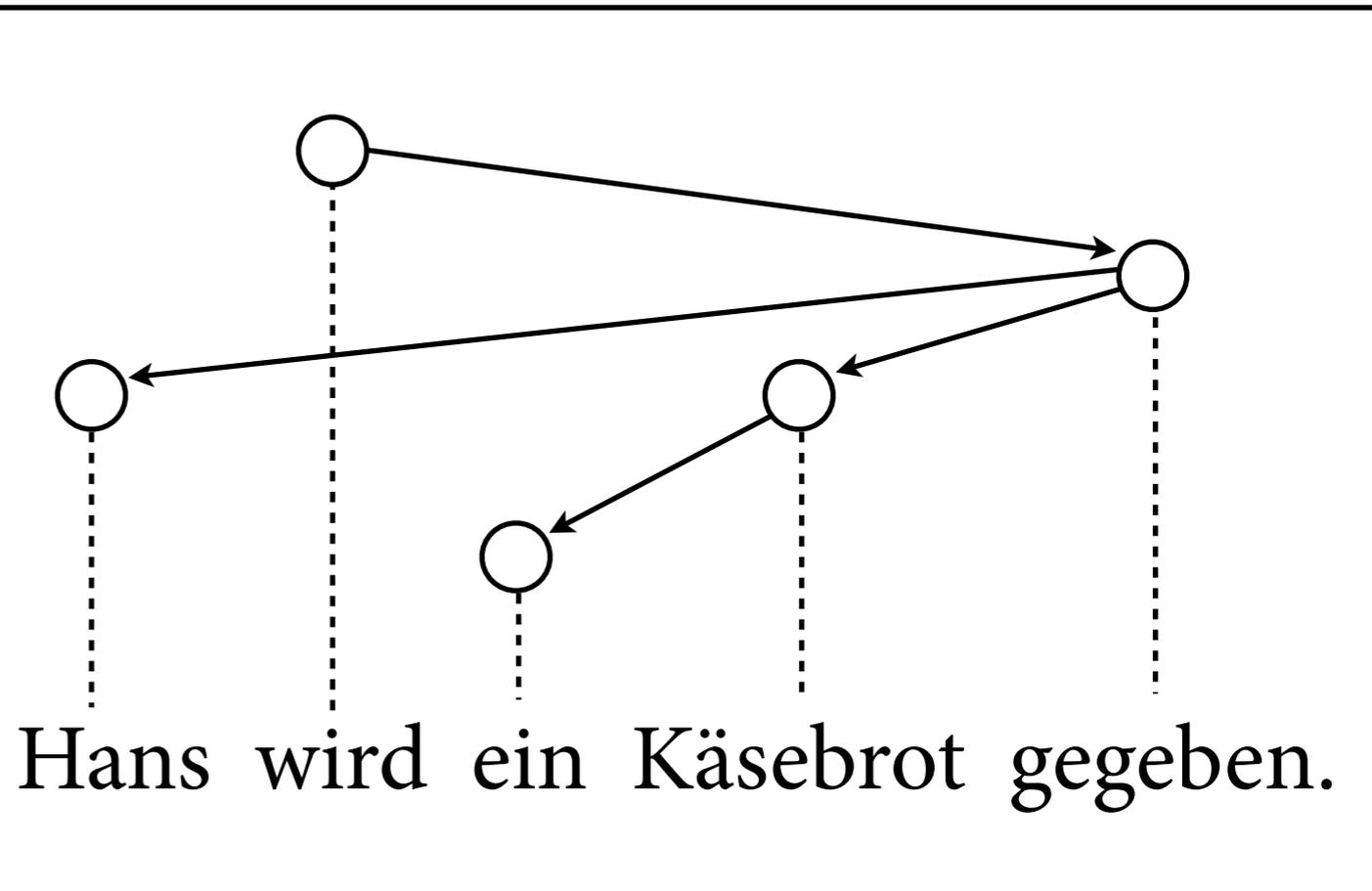
Global vs. lokal



Nichtprojektive Dep.strukturen



*Baum mit
Ordnungsannotationen*

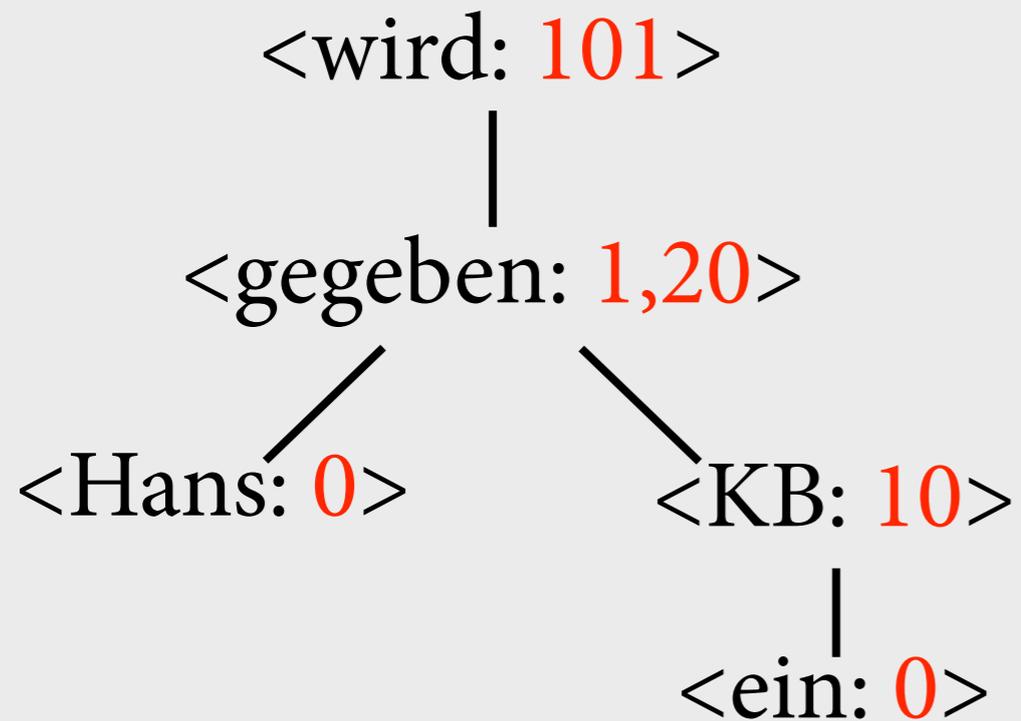


Abhängigkeitsstruktur

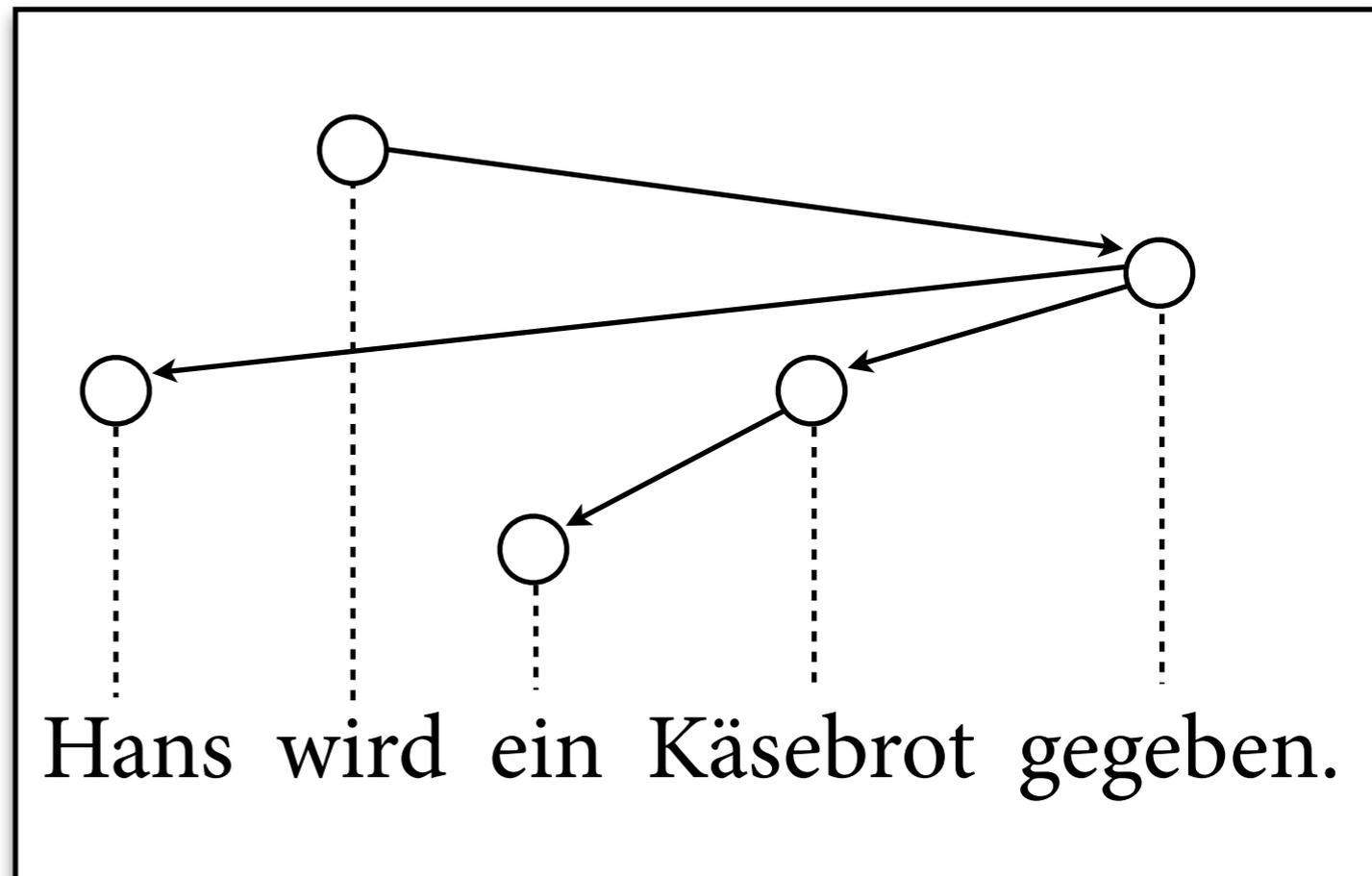
Nichtprojektive OAen

- In nichtprojektiven Dep.strukturen ist Ertrag eines Knotens nicht immer String, sondern Liste von mehreren *Blöcken*.
- Ordnungsannotationen erweitern:
 - ▶ $\langle 01,23 \rangle$: zwei Blöcke; erster besteht aus 01, der zweite aus 23
 - ▶ $\langle 1201 \rangle$: erstes Kind hat zwei Blöcke; String besteht aus erstem Block des ersten Kindes, dann einzigem Block des zweiten Kindes, dann 0, dann zweitem Block des ersten Kindes

Nichtprojektive OAen



*Baum mit
Ordnungsannotationen*



Abhängigkeitsstruktur

OA-Codierung von Dep.strukturen

- Sei OA die Menge aller Ordnungsannotationen.
- Für jede Dependenzstruktur über Signatur Σ ist die OA-Codierung ein Baum über Signatur $\Sigma \times \text{OA}$.
- Für jede Dependenzstruktur gibt es eine eindeutige OA-Codierung.
- Aus einer OA-Codierung kann man eindeutig die Dependenzstruktur rekonstruieren.

Reguläre Dependenzgrammatiken

- Eine *reguläre Dependenzgrammatik* (RDG) G ist eine RTG über einer Signatur $\Sigma \times \text{OA}$.
- Lexikalisiert: In jeder Regel genau ein Wort aus Σ .
- Sprachen:
 - ▶ zunächst: Bäume über der Signatur $\Sigma \times \text{OA}$
 - ▶ lese als Sprache $L(G)$ von Dependenzstrukturen
 - ▶ Stringsprache von $G = \text{Erträge der Bäume in } L(G)$

Beispiel

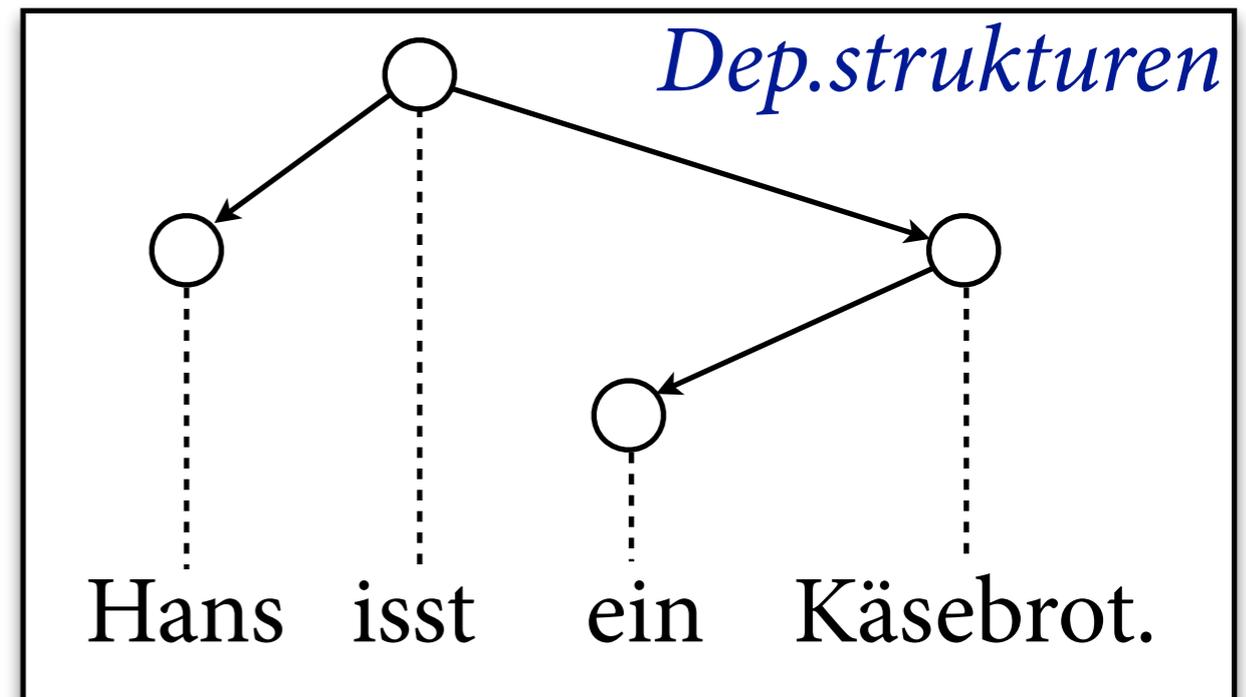
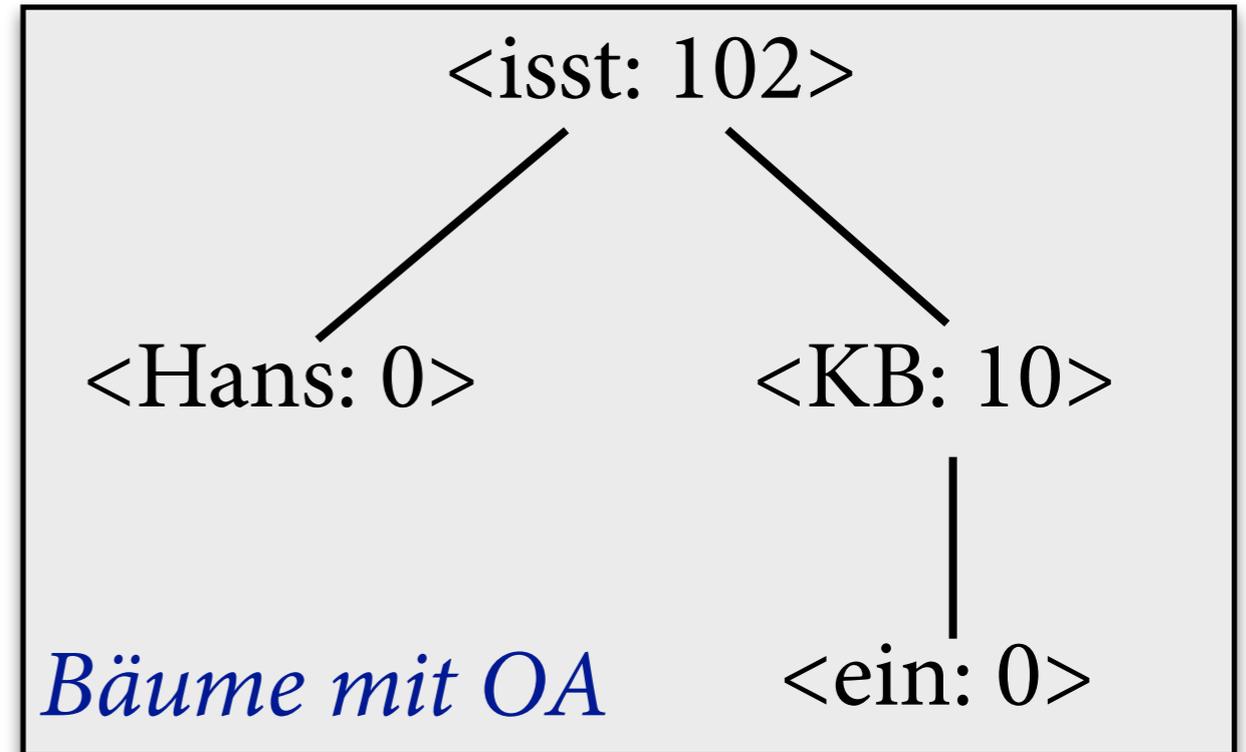
$S \rightarrow \langle \text{isst: 102} \rangle (\text{NP}, \text{VP})$

$\text{NP} \rightarrow \langle \text{Hans: 0} \rangle$

$\text{NP} \rightarrow \langle \text{KB: 10} \rangle (\text{Det})$

$\text{Det} \rightarrow \langle \text{ein: 0} \rangle$

RDG-Grammatik



Zusammenfassung

- Reguläre Abhängigkeitsgrammatiken: Ein Formalismus für die Grammatiktheorie “Abhängigkeitsgrammatik”.
- Abhängigkeitsstruktur = Baum + lin. Ordnung
 - ▶ codiert als normaler Baum mit Ordnungsannotationen
- Reguläre Abhängigkeitsgrammatik: RTG über erweiterter Signatur mit Ordnungsannotationen