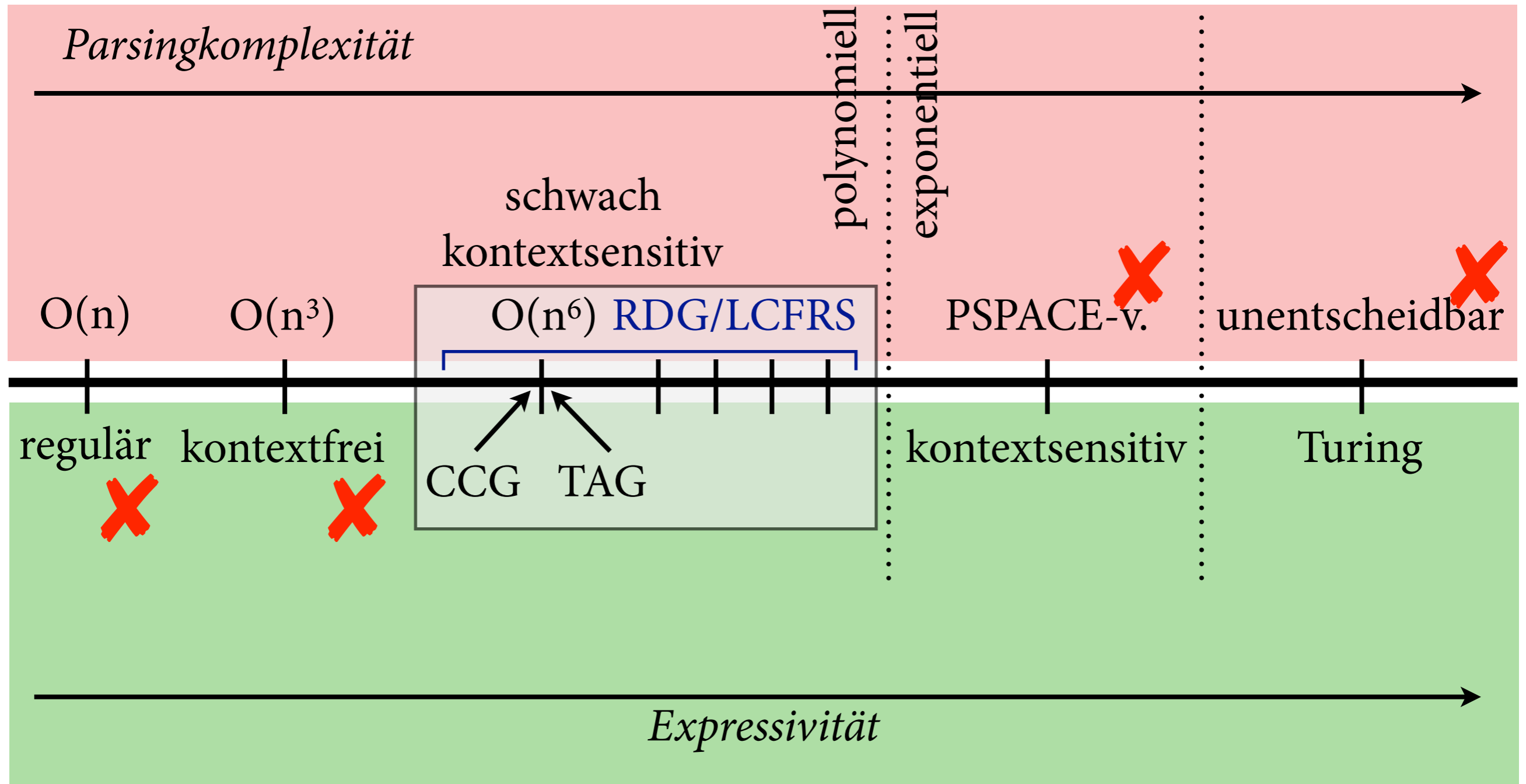


Reguläre Abhängigkeitsgrammatiken

Vorlesung “Grammatikformalismen”
Alexander Koller

6. Juni 2017

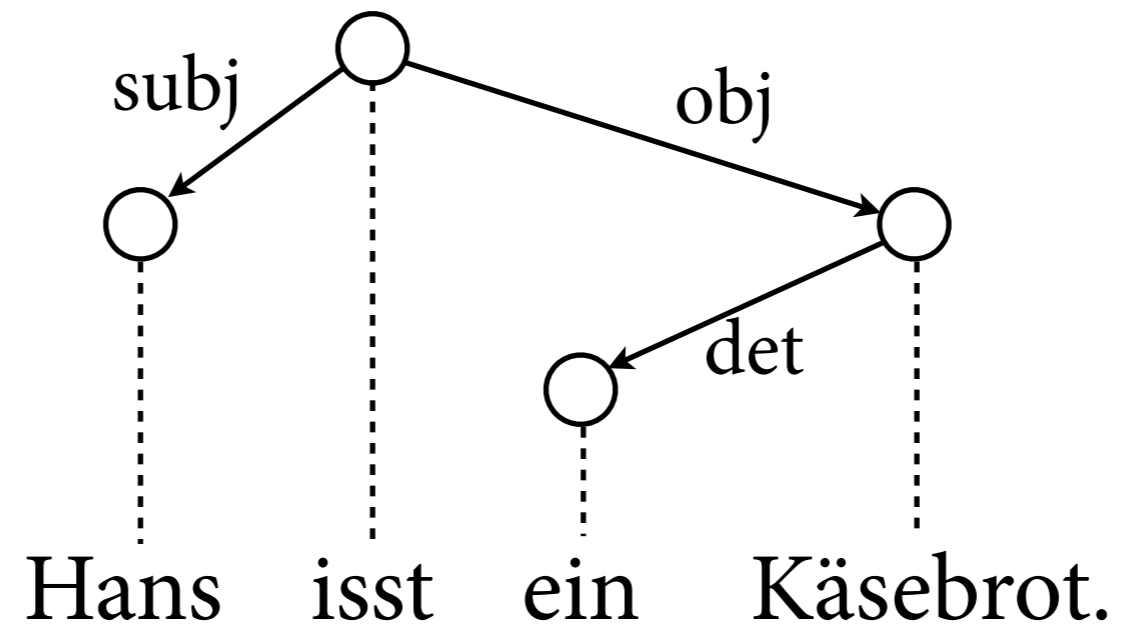
Eine allgemeinere Perspektive



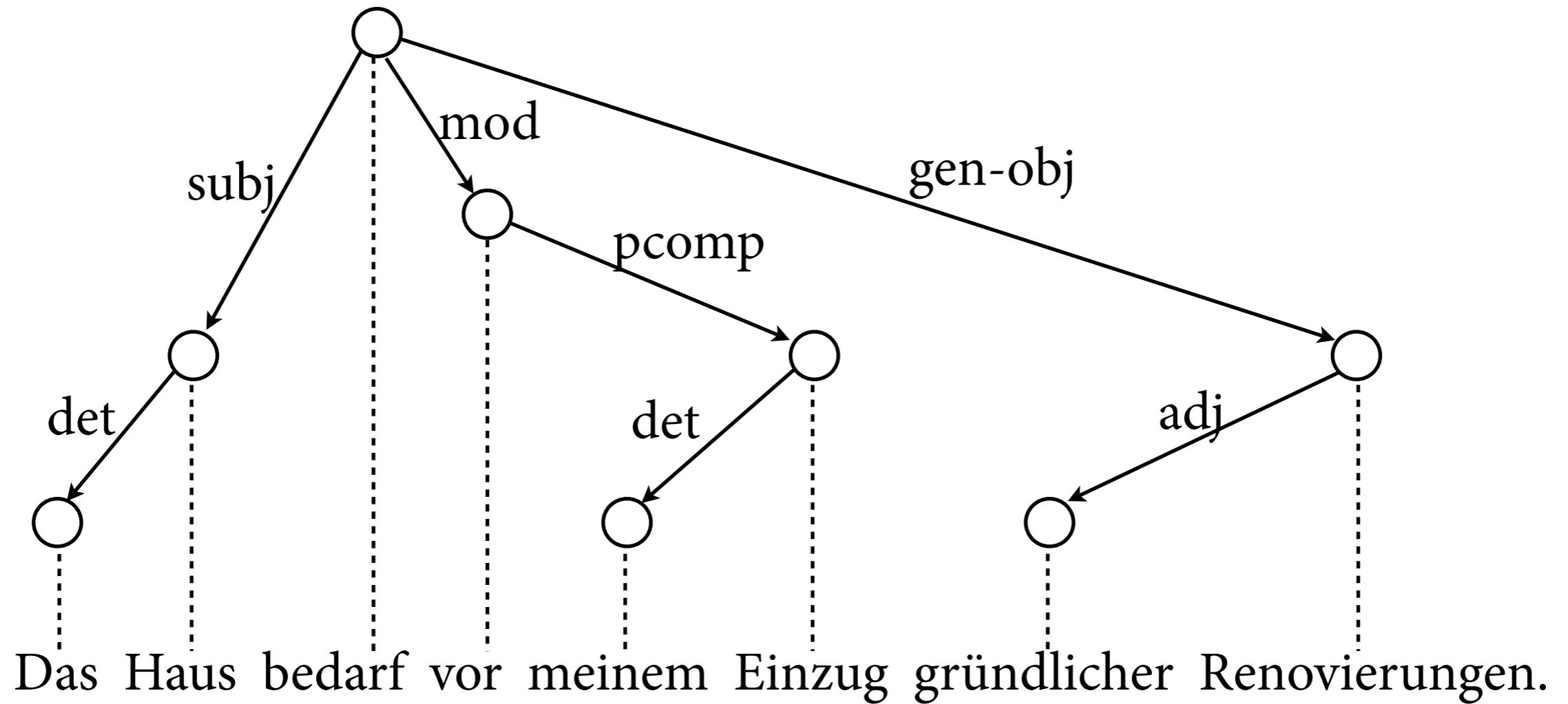
Dependenzgrammatiken

- Grammatiktheorie: Syntaktische Strukturen = *Dependenzen* zwischen Wörtern.
- Dependenzbaum:
 - ▶ 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

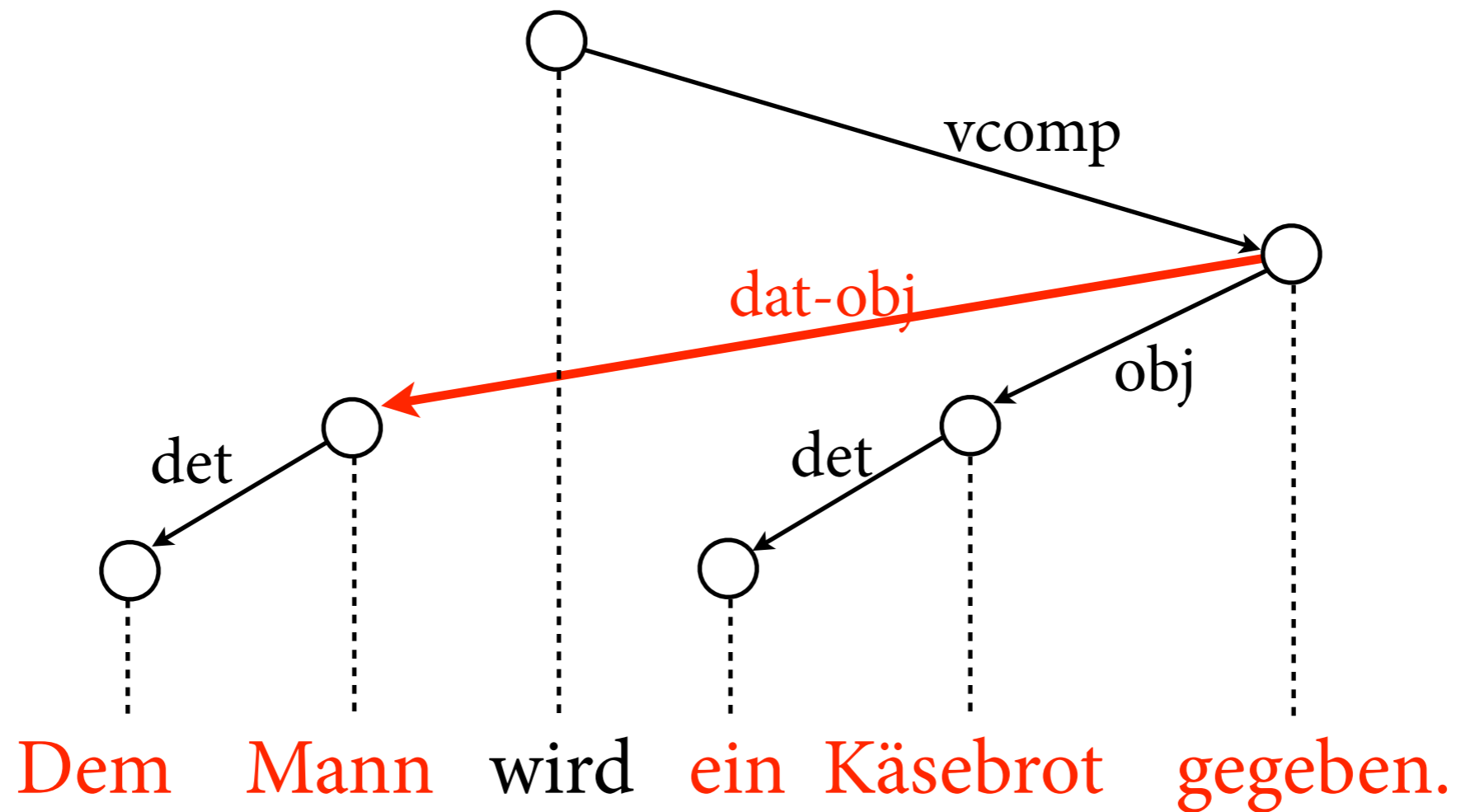
Dependenzbäume: Beispiele



Dependenzbäume: Beispiele

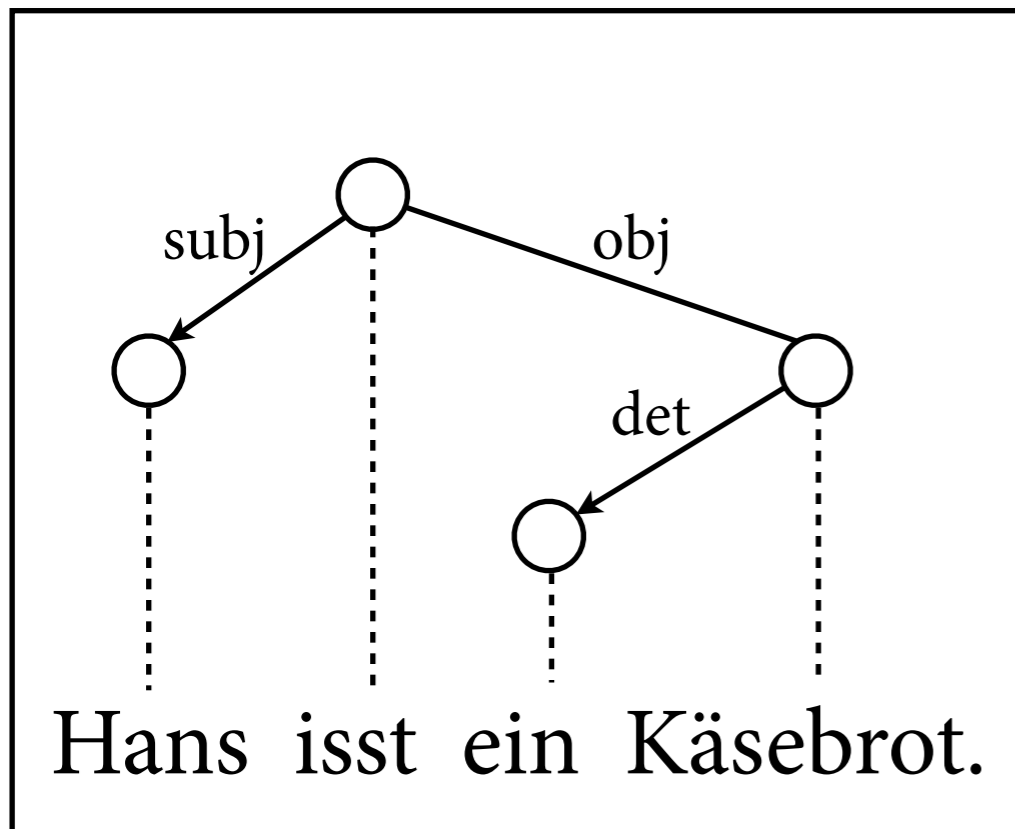


Dependenzbäume: Beispiele

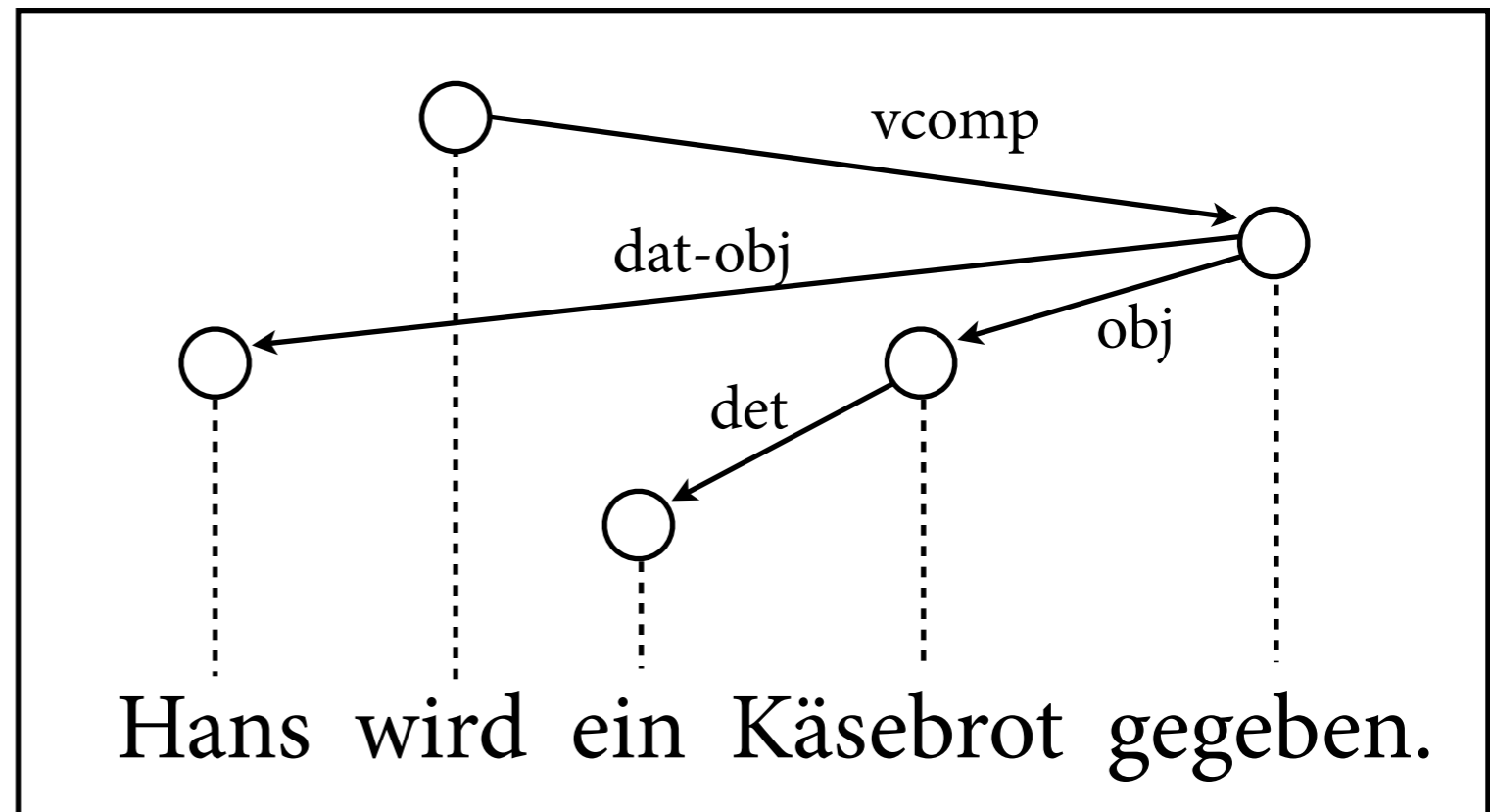


Projektivität

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



projektiv



nichtprojektiv

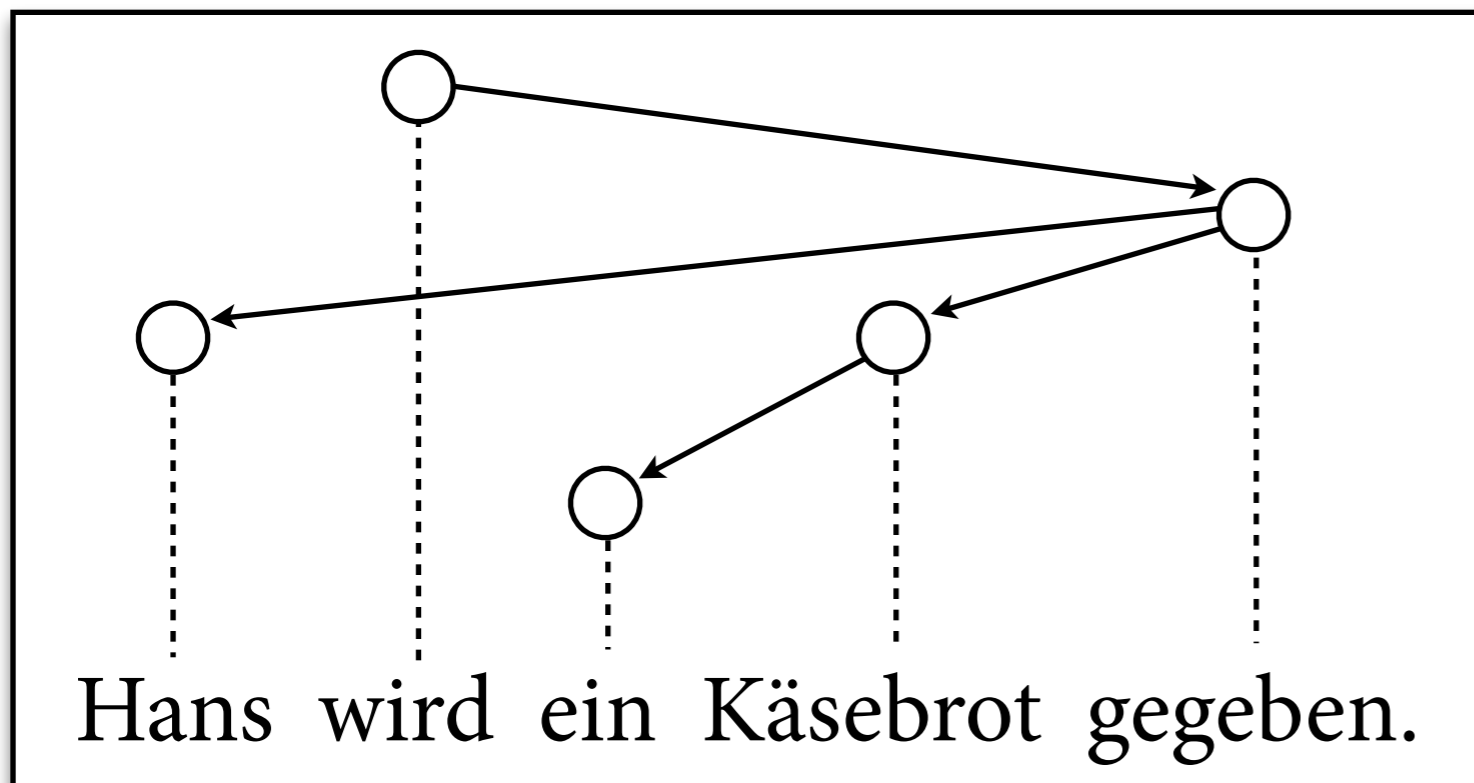
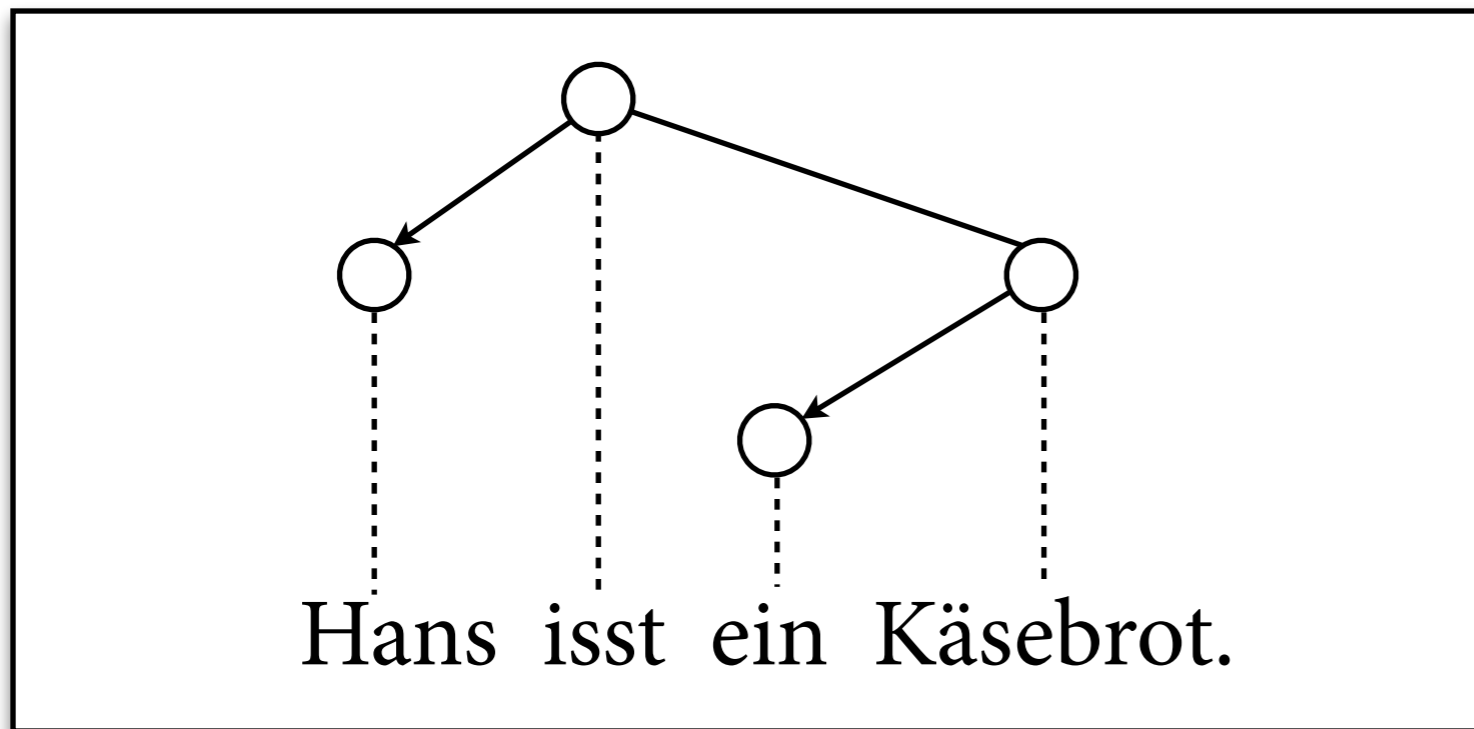
Grammatikformalismus

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

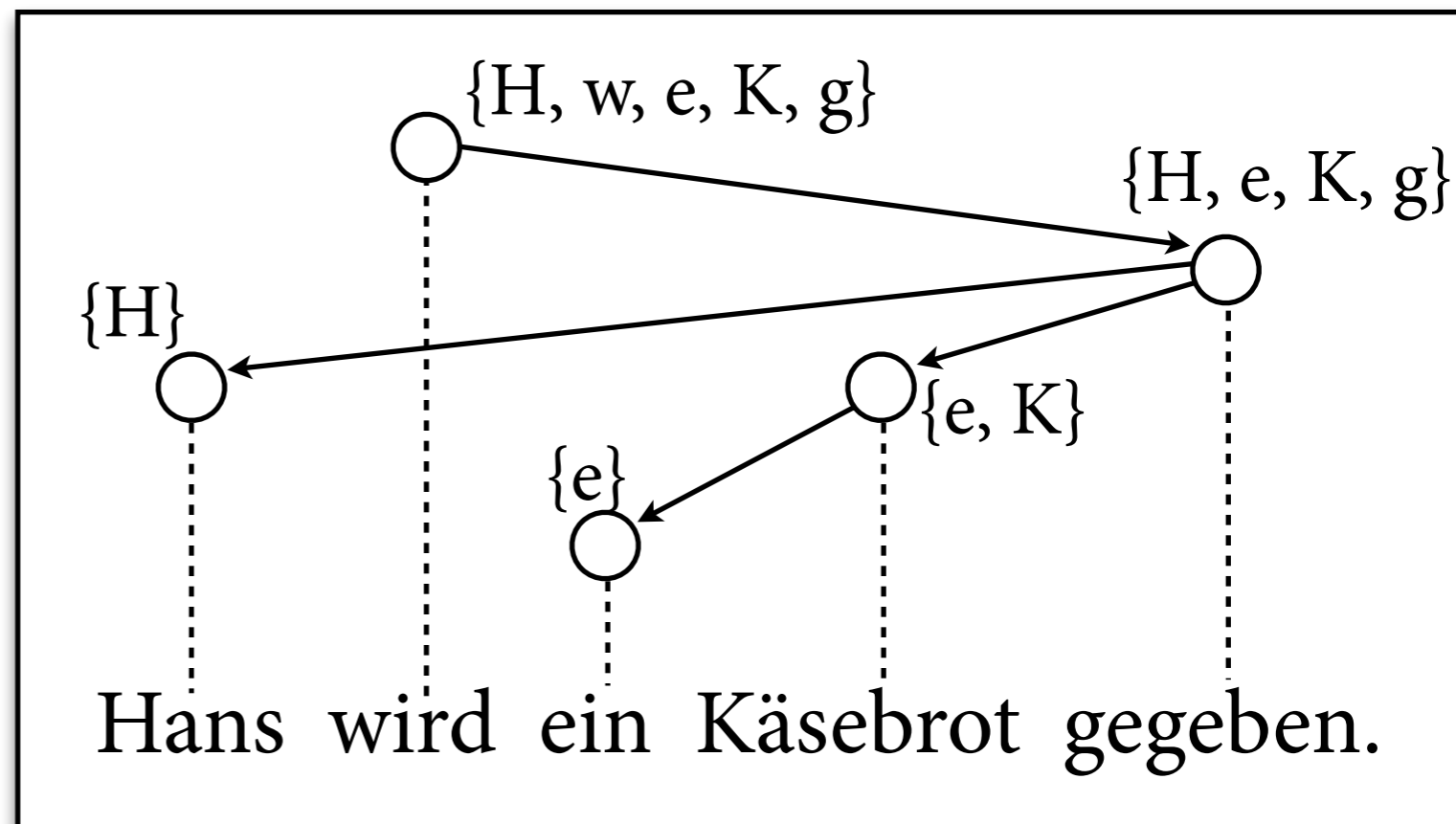
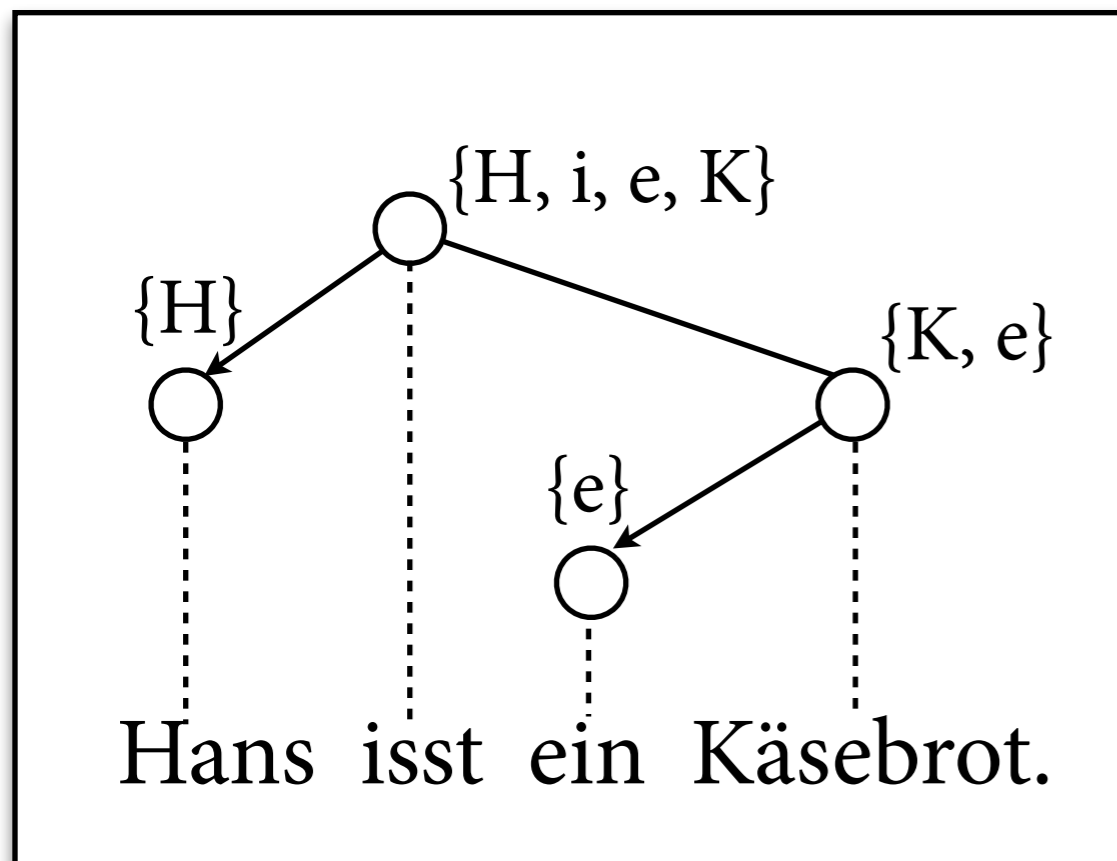
Beispiel



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

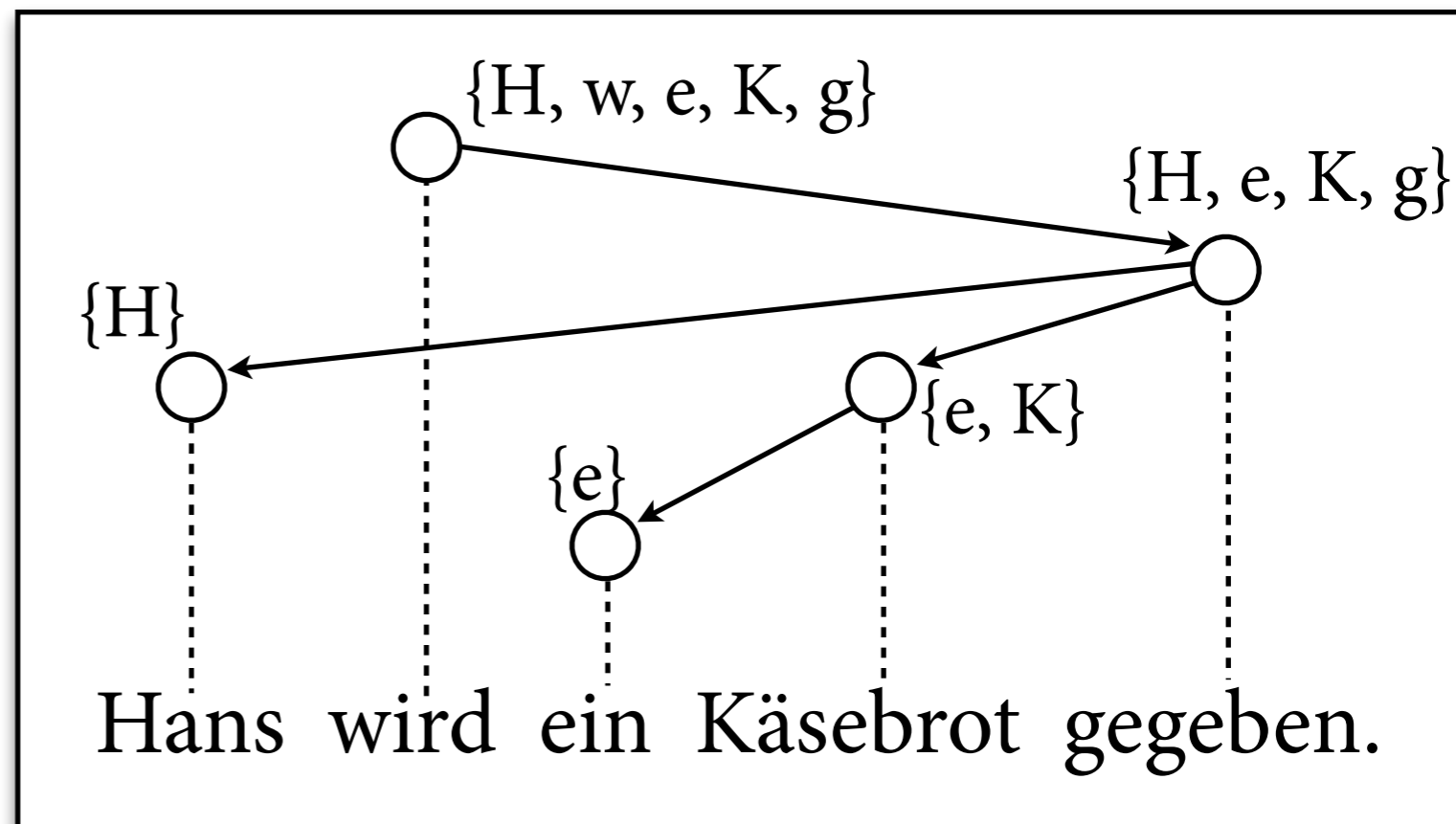
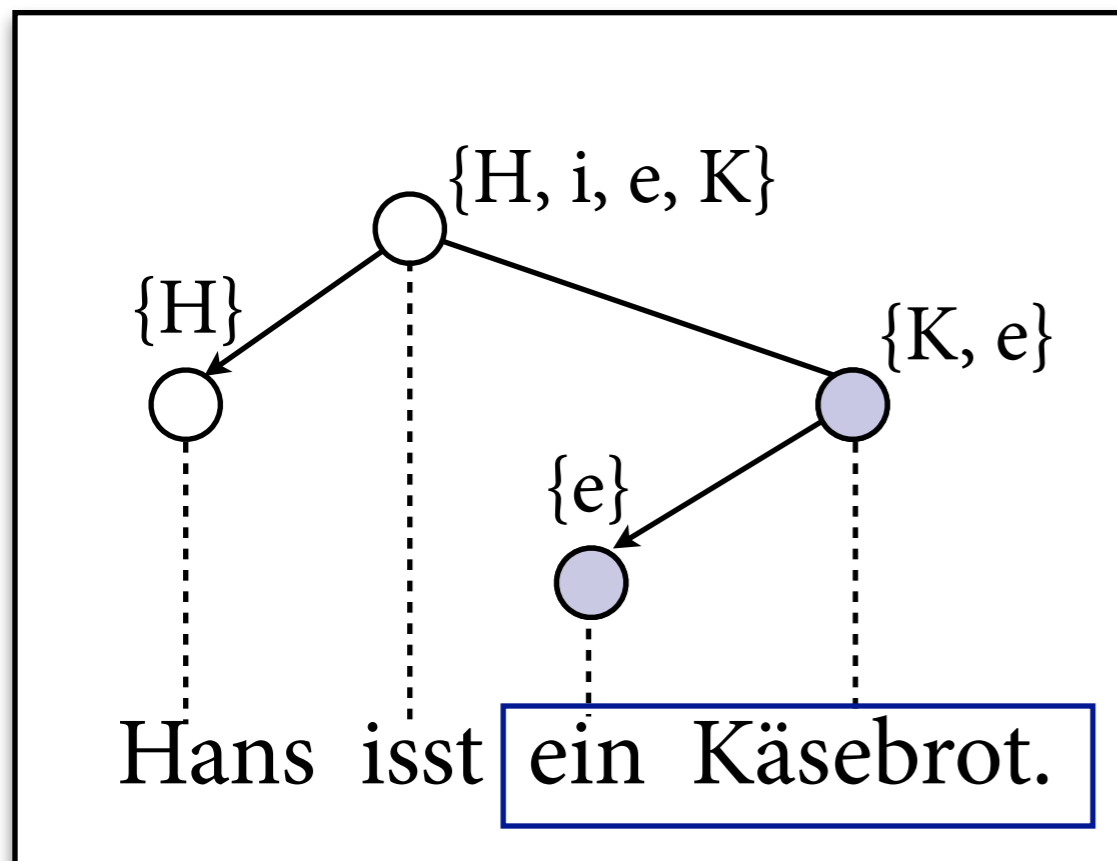
Ertrag

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



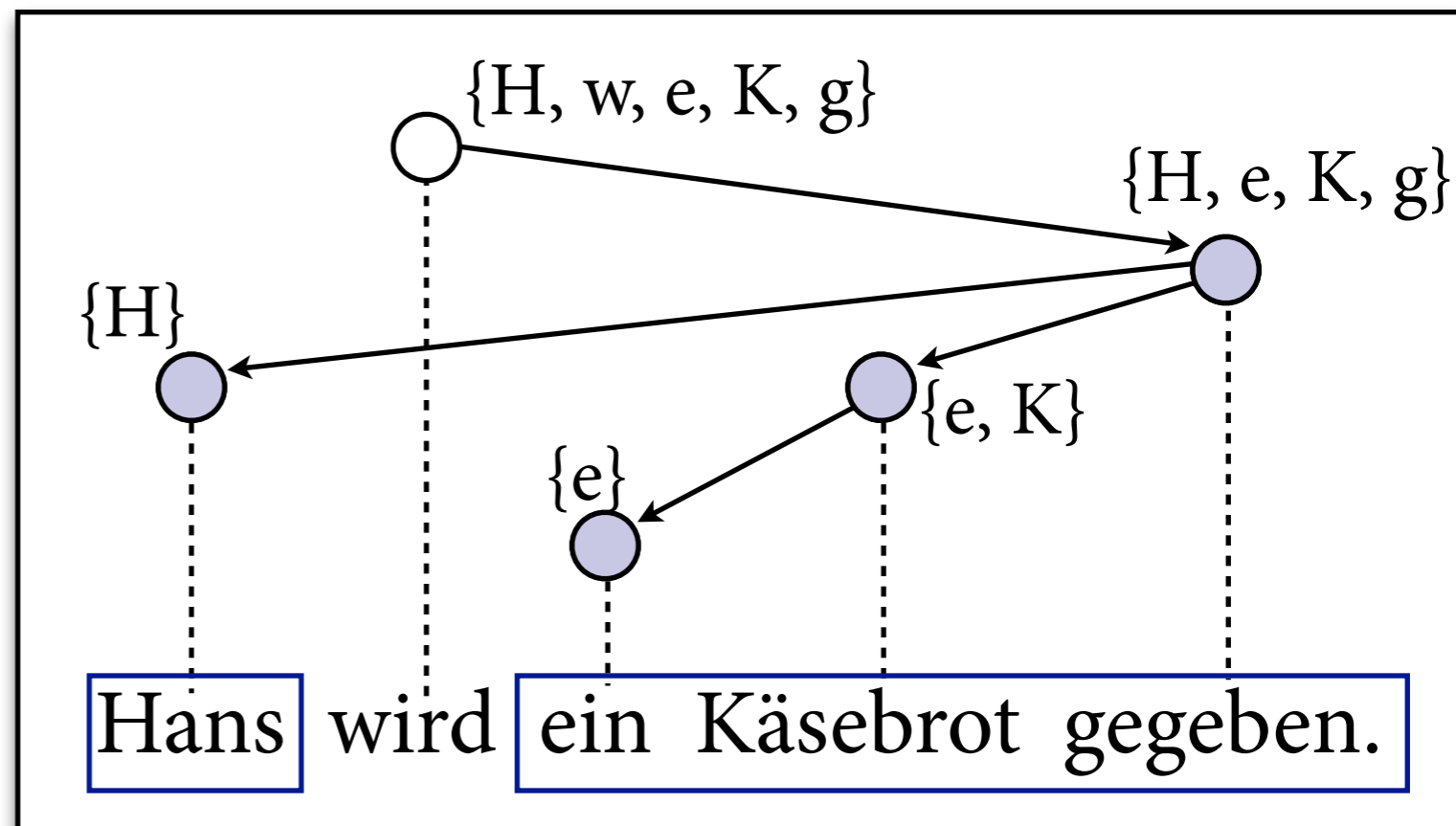
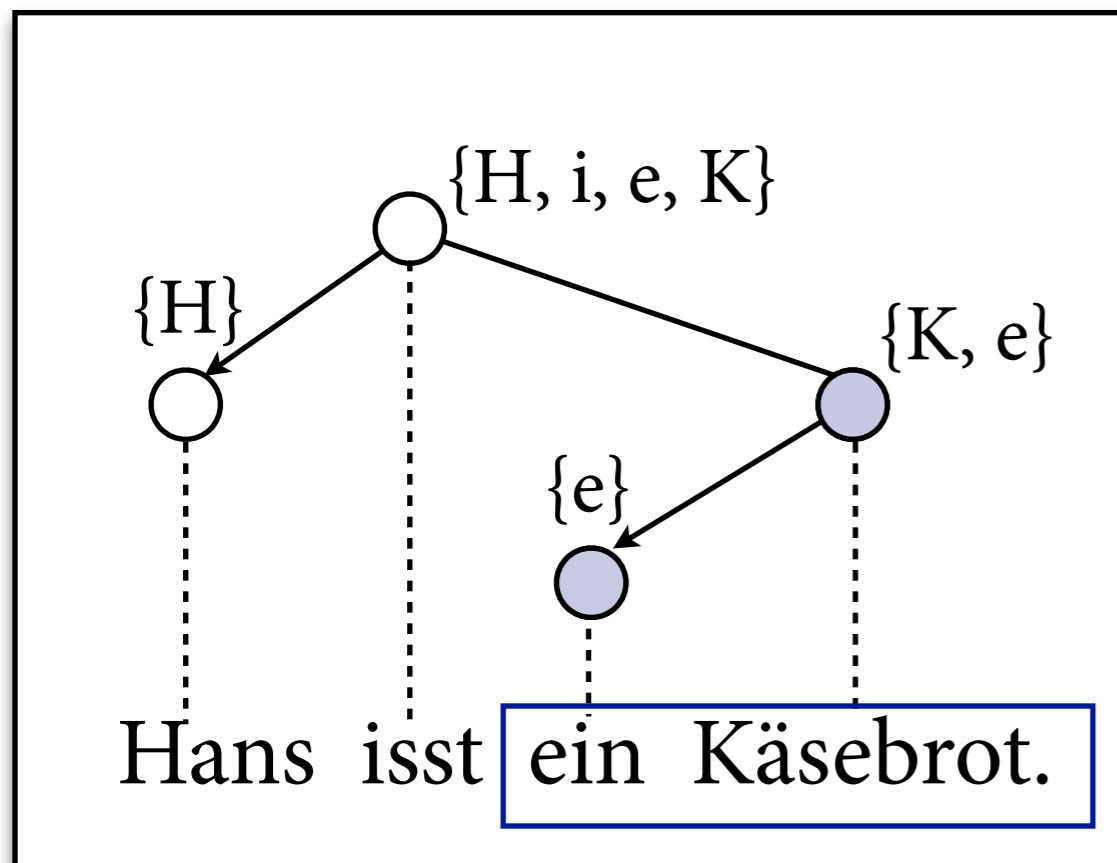
Ertrag

- *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.



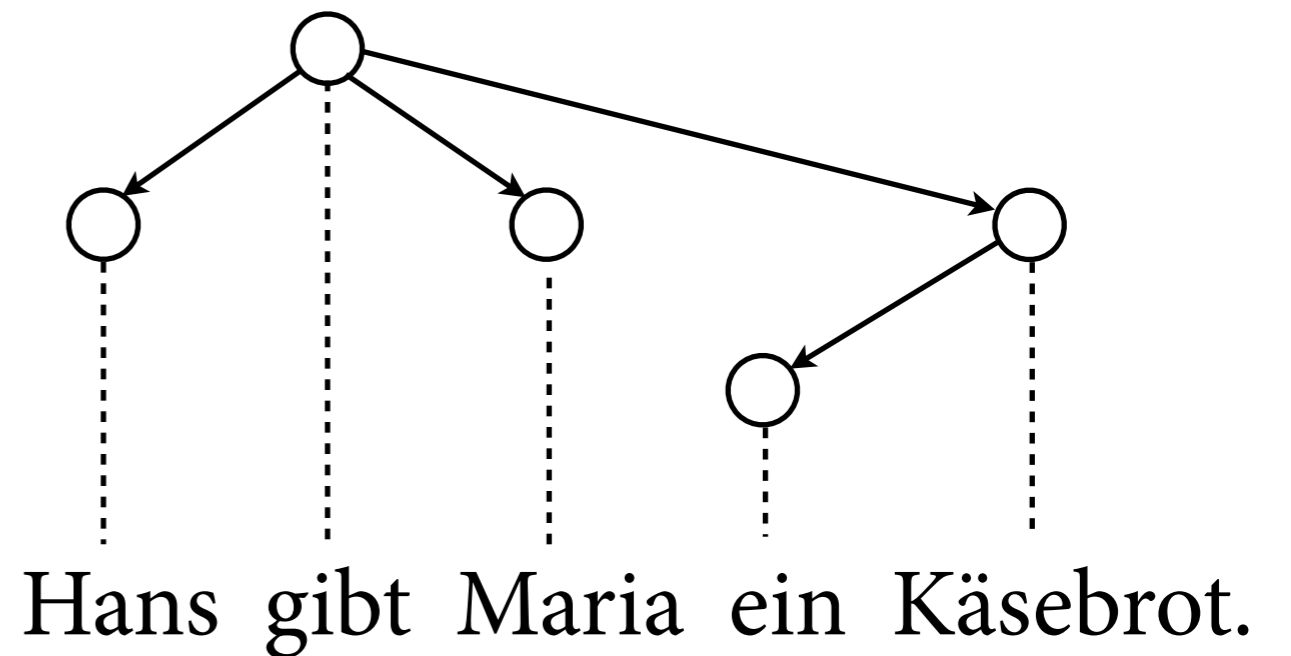
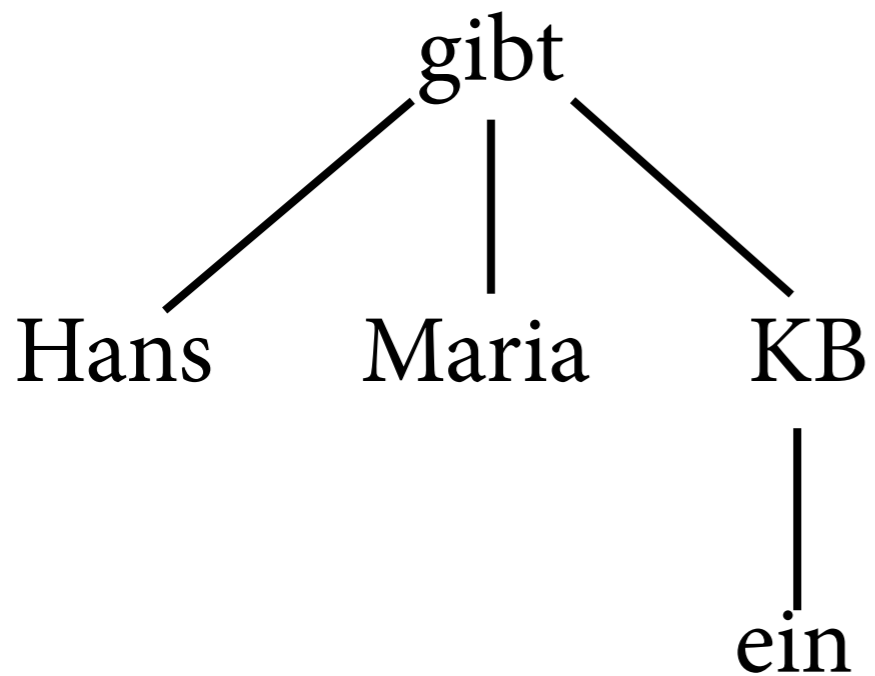
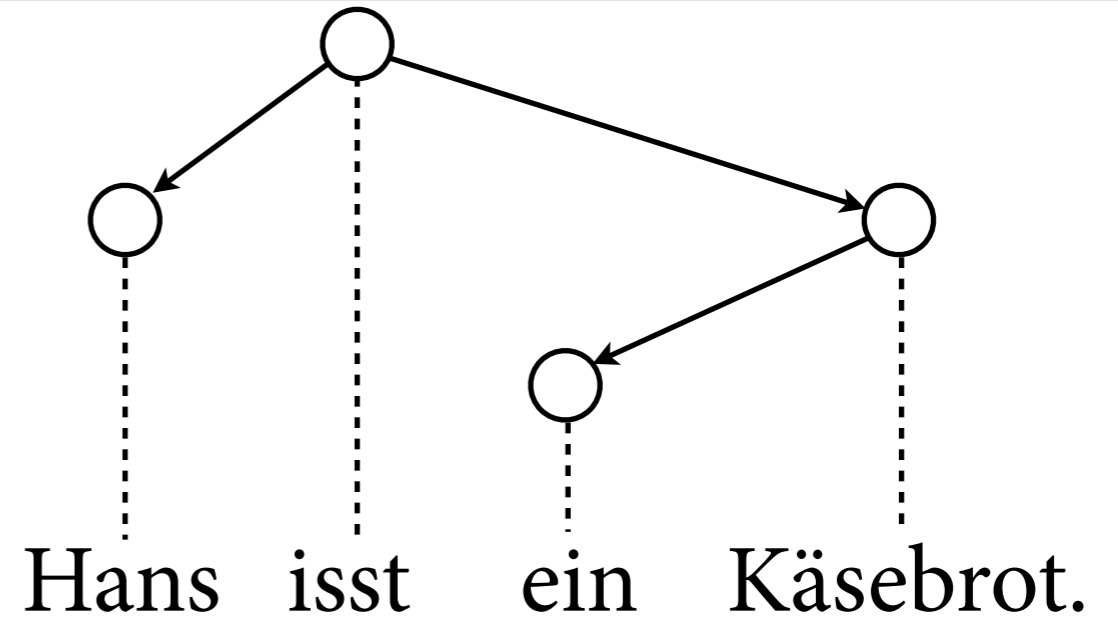
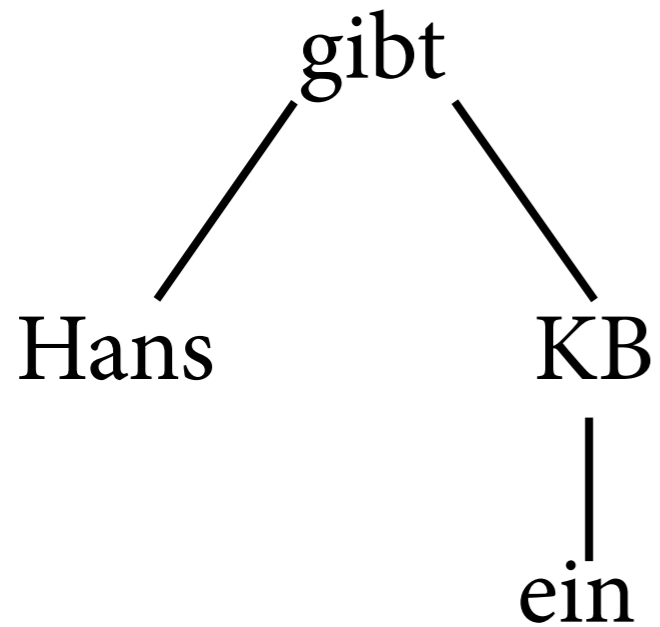
Projektivität

- *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?

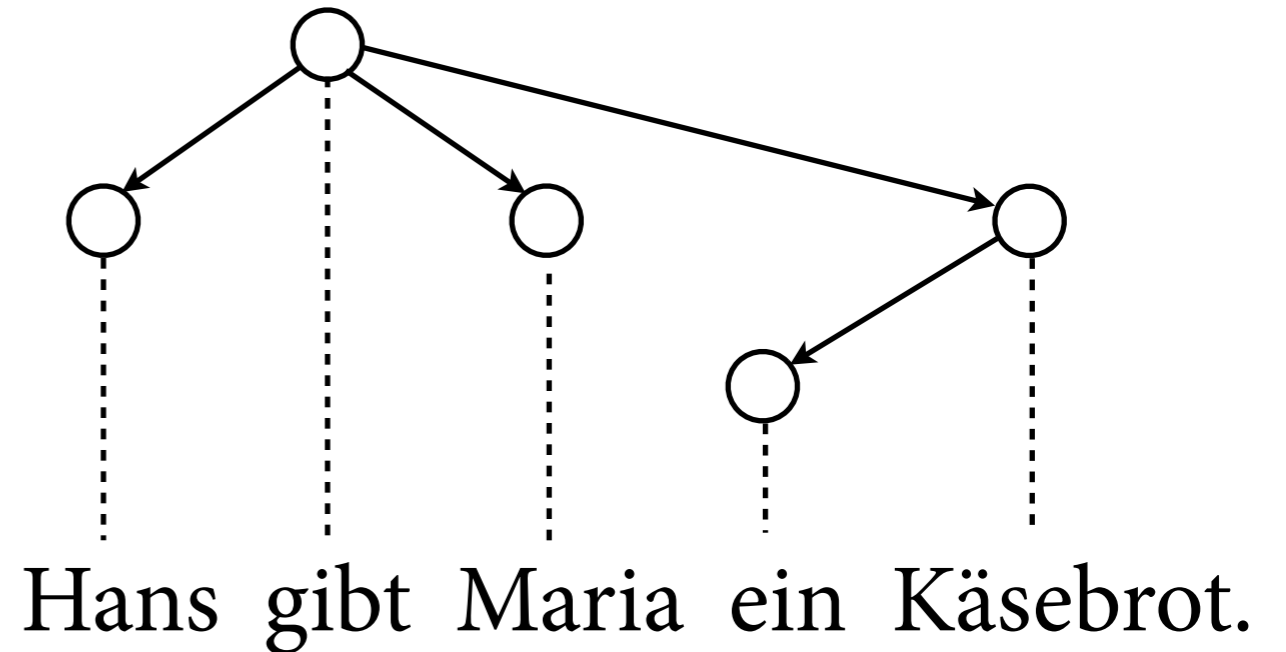
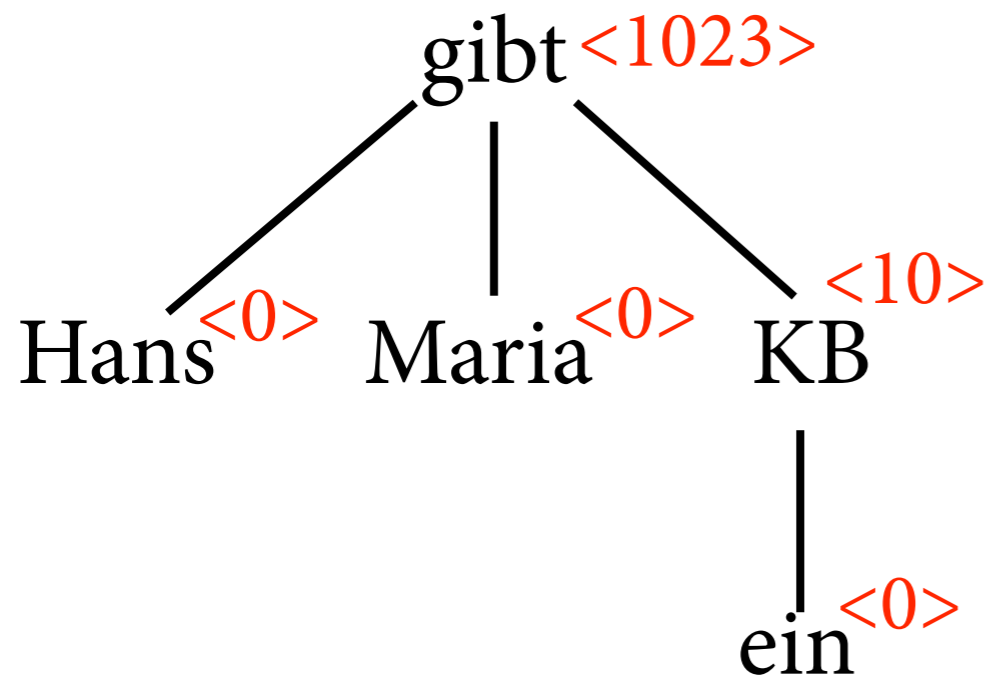
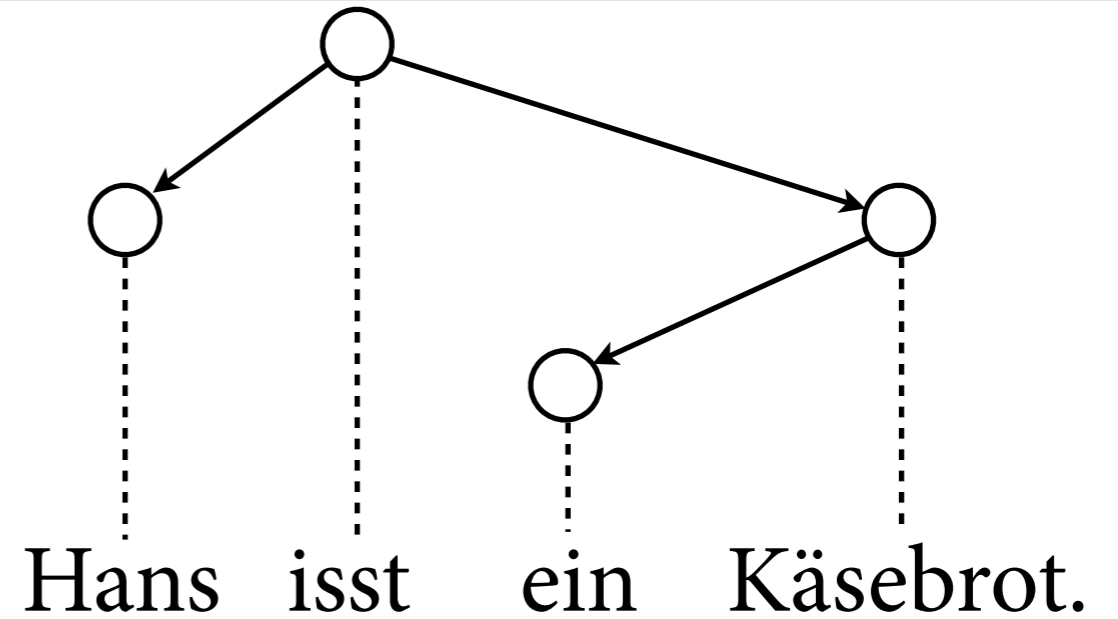
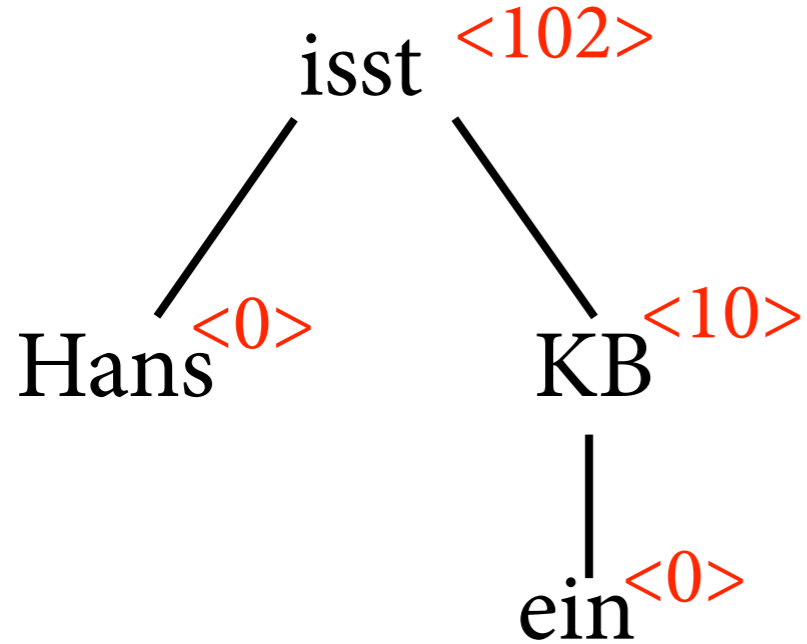
Global vs. lokal



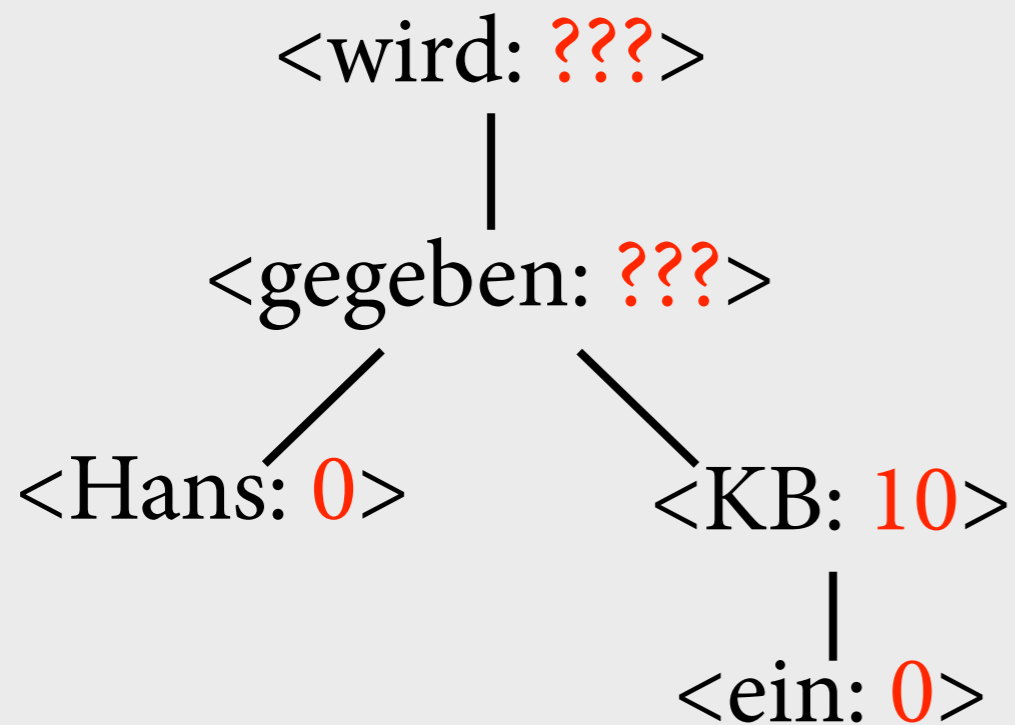
Ordnungsannotationen

- 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.

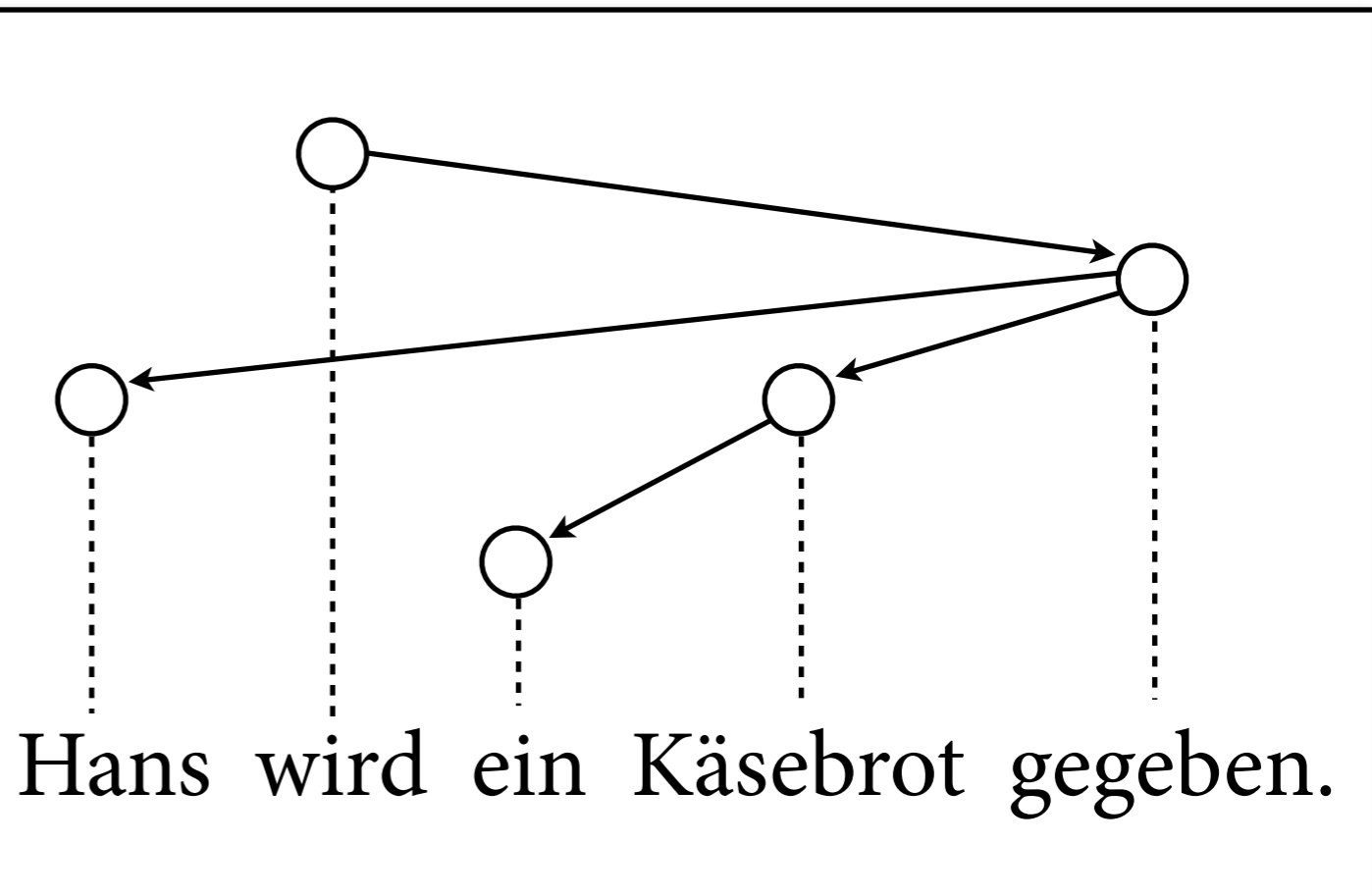
Global vs. lokal



Nichtprojektive Dep.strukturen



*Baum mit
Ordnungsannotationen*

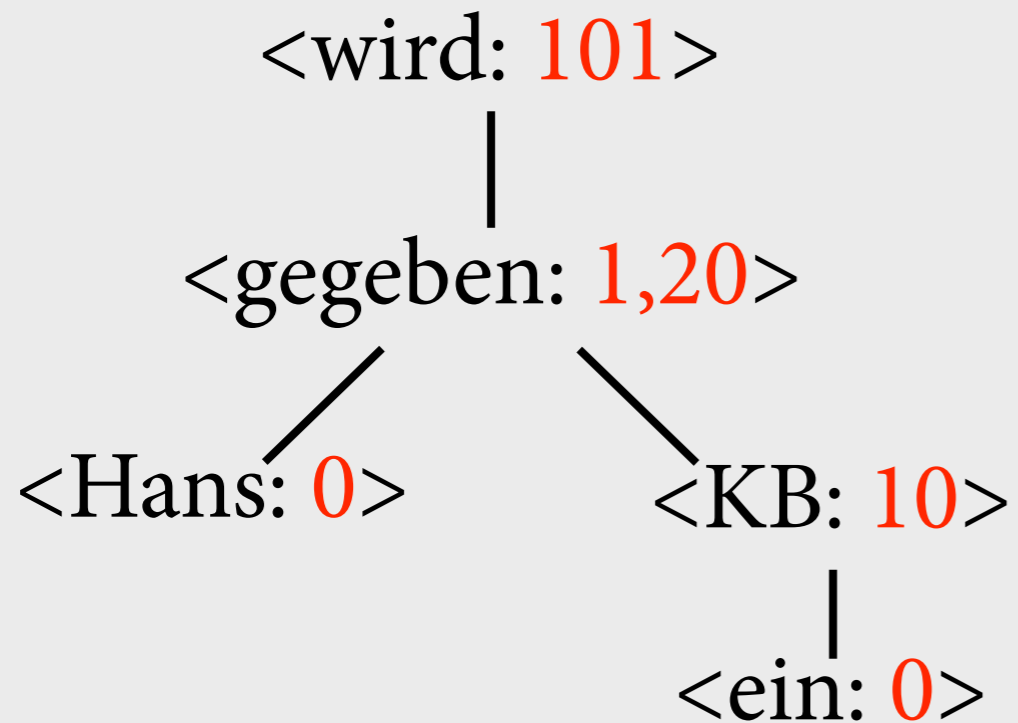


Abhängenzstruktur

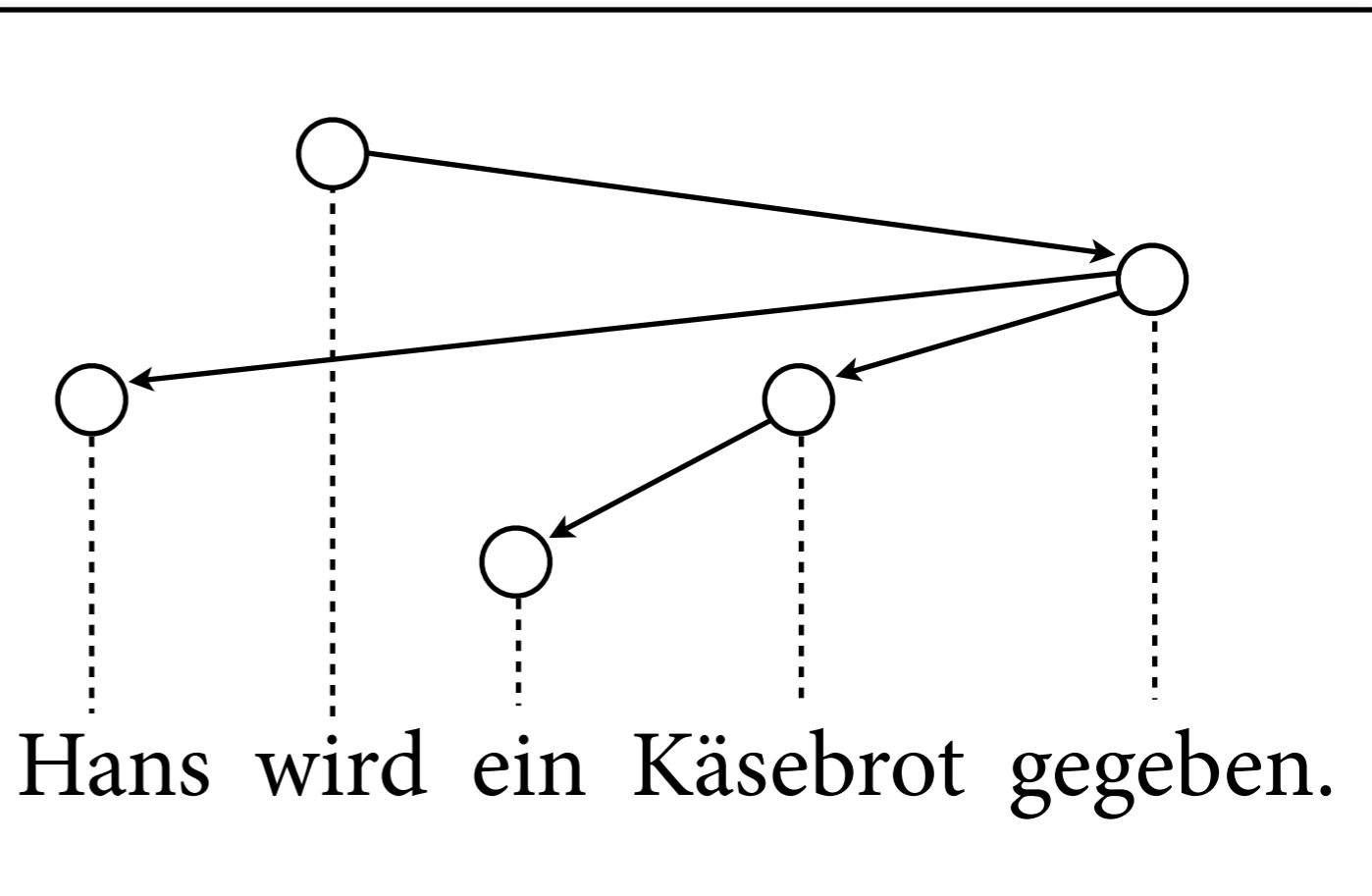
Nichtprojektive OAen

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

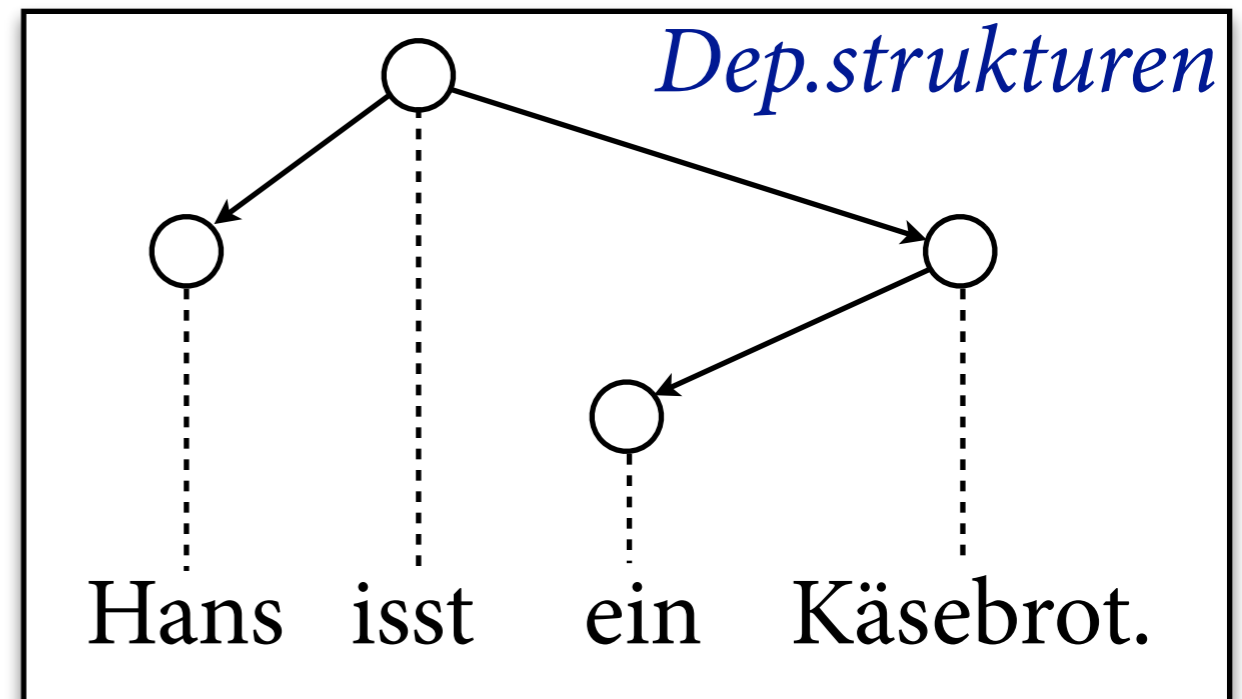
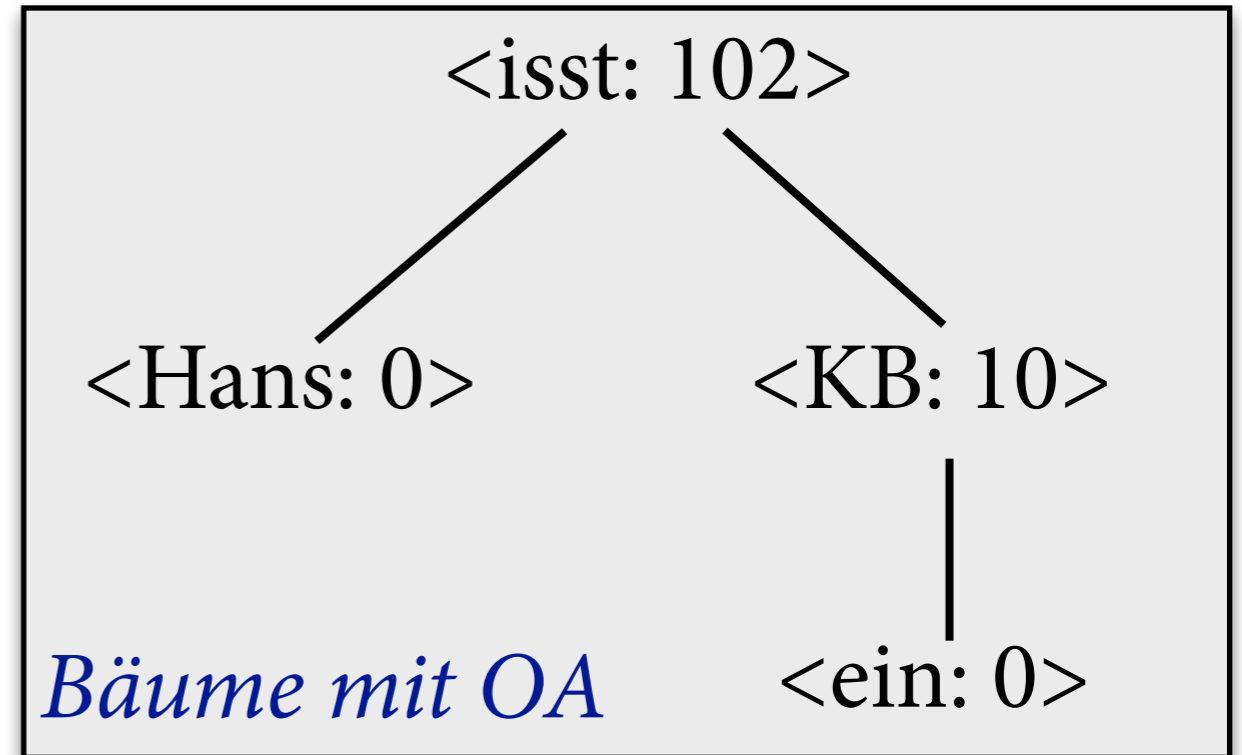
$S \rightarrow \langle \text{isst: 102} \rangle (\text{NP}, \text{VP})$

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

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

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

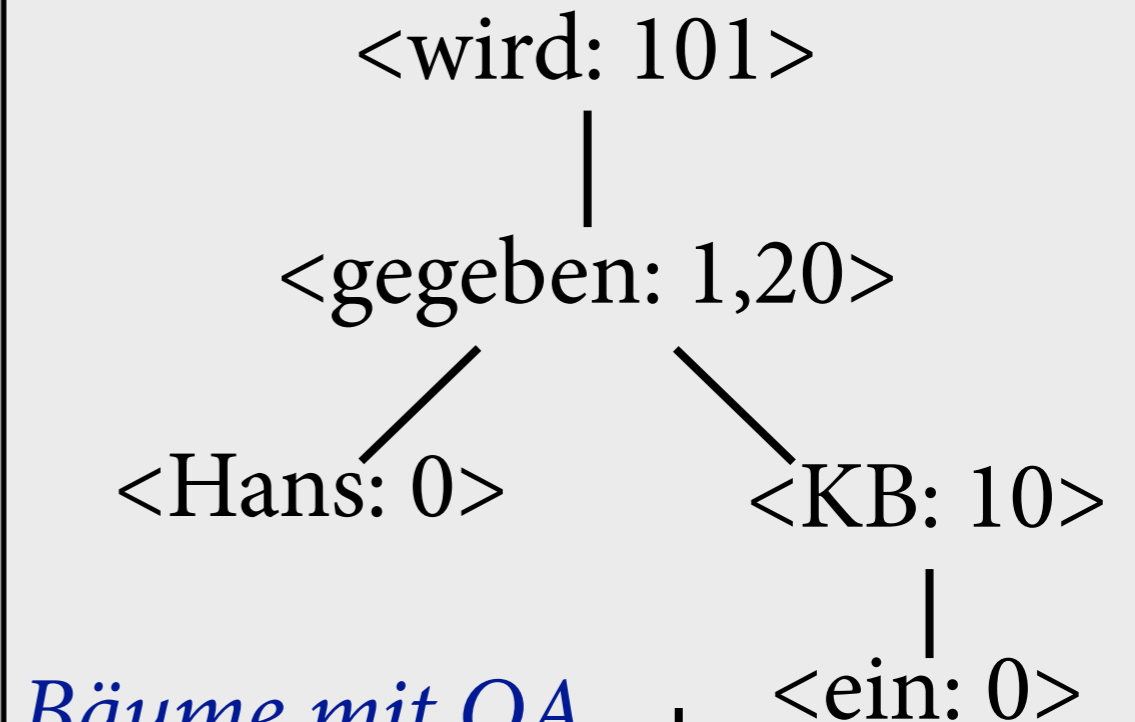
RDG-Grammatik



Beispiel

$S \rightarrow \langle \text{wird: 101} \rangle (\text{VP})$
 $\text{VP} \rightarrow \langle \text{geg: 1,20} \rangle (\text{NP}, \text{NP})$
 $\text{NP} \rightarrow \langle \text{Hans: 0} \rangle$
 $\text{NP} \rightarrow \langle \text{KB: 10} \rangle (\text{Det})$
 $\text{Det} \rightarrow \langle \text{ein: 0} \rangle$

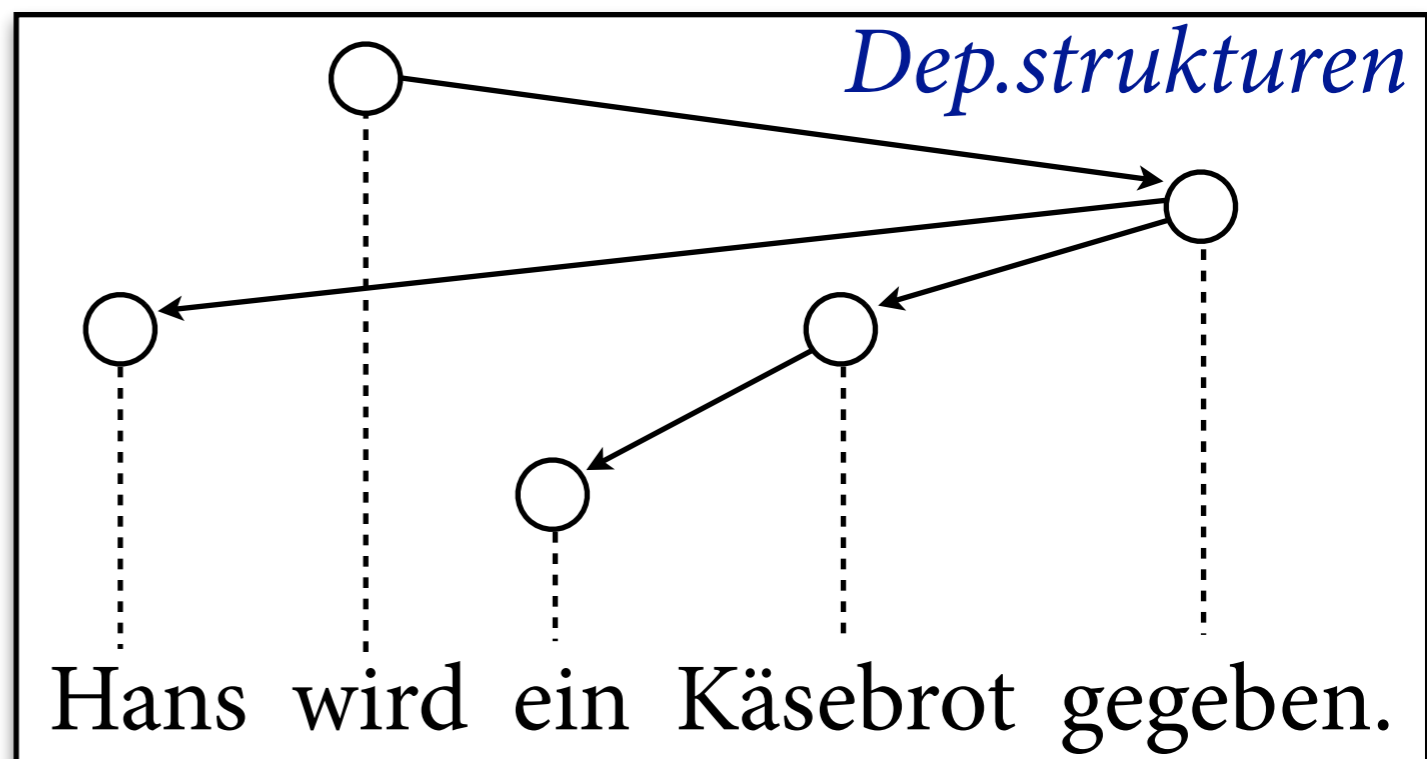
RDG-Grammatik



Bäume mit OA



Dep.strukturen



RDG als IRTG

$S \rightarrow \langle \text{wird: 101} \rangle (\text{VP})$
 $\text{VP} \rightarrow \langle \text{geg: 1,20} \rangle (\text{NP}, \text{NP})$
 $\text{NP} \rightarrow \langle \text{Hans: 0} \rangle$
 $\text{NP} \rightarrow \langle \text{KB: 10} \rangle (\text{Det})$
 $\text{Det} \rightarrow \langle \text{ein: 0} \rangle$

RDG-Grammatik

$S \rightarrow r_1(\text{VP}) \quad h(r_1) = \text{“wird:101”}(x_1)$
 $\text{VP} \rightarrow r_2(\text{NP}, \text{NP}) \quad h(r_2) = \text{“gegeben:1,20”}(x_1, x_2)$
 $\text{NP} \rightarrow r_3 \quad h(r_3) = \text{“Hans:0”}$
 $\text{NP} \rightarrow r_4(\text{Det}) \quad h(r_4) = \text{“KB:10”}(x_1)$
 $\text{Det} \rightarrow r_5 \quad h(r_5) = \text{“ein:0”}$

IRTG-Grammatik

RDG-String-Algebra:

Domäne: Strings + String-Tupel

$I(\text{gegeben:1,20}) (\text{“Hans”, “ein Käsebrot”})$
 $= \langle \text{“Hans”, “ein Käsebrot gegeben”} \rangle$

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