Kontextfreie Grammatiken

Vorlesung “Computerlinguistische Techniken”

Alexander Koller

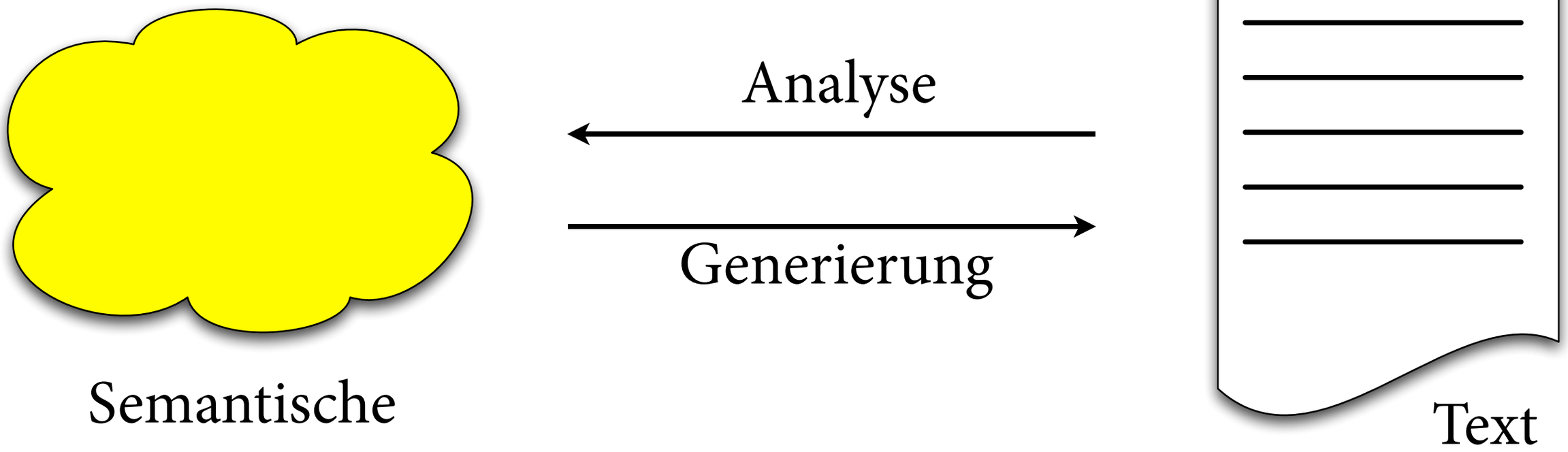
16. Oktober 2015

Übersicht

- Worum geht es in dieser Vorlesung?
- Übungen und Abschlussprojekt
- Kontextfreie Grammatiken

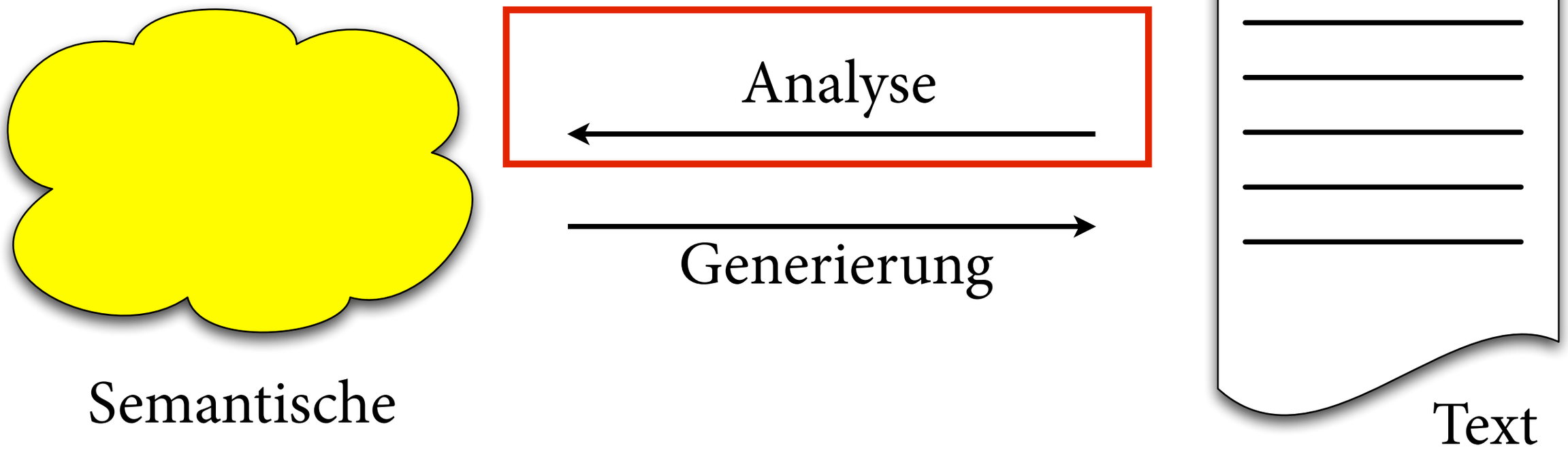
Computerlinguistische Techniken

- Grundlegende Probleme in der CL:



Computerlinguistische Techniken

- Grundlegende Probleme in der CL:



Analyse von Sprache

- Wie kann man die Struktur von sprachlichen Ausdrücken ausrechnen?

- Struktur nicht direkt sichtbar.

- Um Ausdrücken Struktur zuzuweisen, braucht man Wissen über Sprache.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

Gefallen findet er daran bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

		NN		
Gefallen	findet	er	daran	bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN		
Gefallen	findet	er	daran bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN	PPER	
Gefallen	findet	er	daran bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN	PPER	PAV	
Gefallen	findet	er	daran	bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN	PPER	PAV	ADV
Gefallen	findet	er	daran	bestimmt.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN	PPER	PAV	ADV
Gefallen	findet	er	daran	bestimmt.
Gefallen	ist	er		nicht.

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN VVFIN PPER PAV ADV
Gefallen findet er daran bestimmt.

Gefallen ist er nicht.
VVPP

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN VVFIN PPER PAV ADV
Gefallen findet er daran bestimmt.

Gefallen ist er nicht.
VVPP VAFIN

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN VVFIN PPER PAV ADV
Gefallen findet er daran bestimmt.

Gefallen ist er nicht.
VVPP VAFIN PPER

Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN VVFIN PPER PAV ADV
Gefallen findet er daran bestimmt.

Gefallen ist er nicht.
VVPP VAFIN PPER PTKNEG

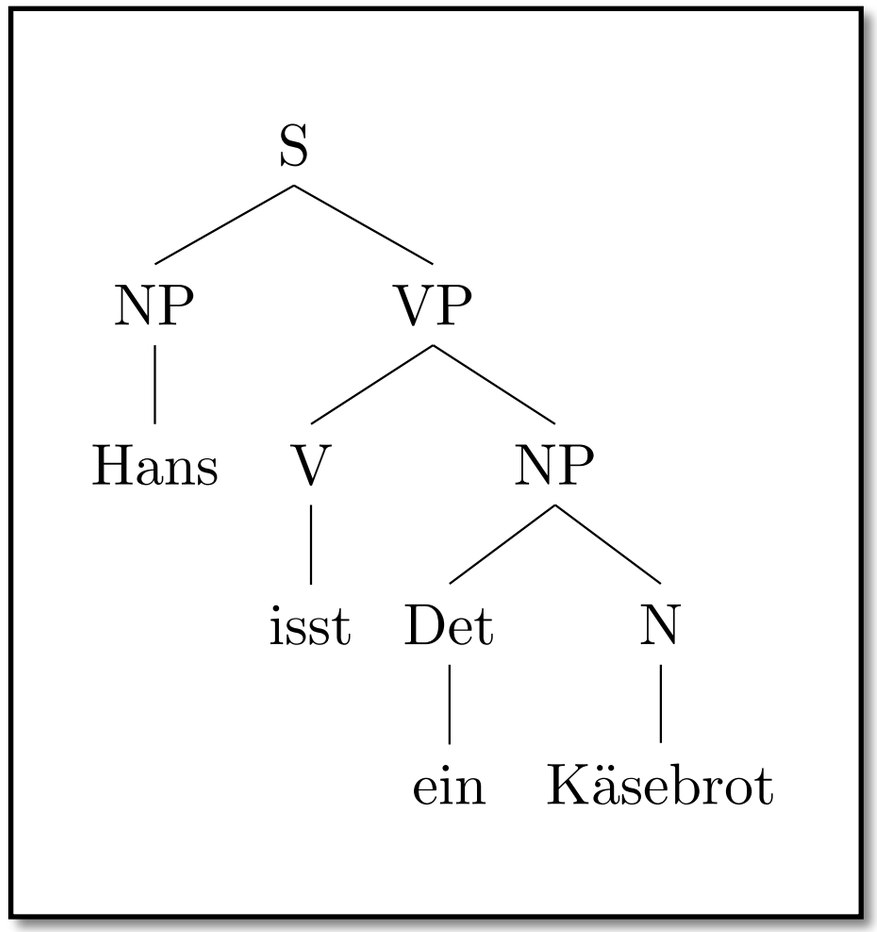
Beispiel: POS-Tagging

- Gegeben einen String von Wörtern, wie findet man die Wortarten der einzelnen Wörter heraus?

NN	VVFIN	PPER	PAV	ADV
Gefallen	findet	er	daran	bestimmt.
Gefallen	ist	er		nicht.
VVPP	VAFIN	PPER		PTKNEG

Beispiel: Parsing

- Was ist die syntaktische Struktur eines gegebenen Strings von Wörtern?



CL-Techniken: Ziele

- Ziel 1: Wie kann man die Struktur sprachlicher Ausdrücke berechnen?

Ambiguität

- Ein zentrales Problem: Sprachliche Ausdrücke sind sehr häufig *mehrdeutig*.
  - Lexikalische Ambiguität: “Gefallen ist” vs. “Gefallen findet”; “die Schule brennt” vs. “die Schule fällt aus”
  - Syntaktische Ambiguität:
    “I shot an elephant in my pajamas. How it got there, I have no idea.”
  - Referenzielle Ambiguität:
    “Hans verprügelte Peter. Das tat ihm weh.”

- Die einzelnen Analysen heißen *Lesarten*.

CL-Techniken: Ziele

- Ziel 1: Wie kann man die Struktur sprachlicher Ausdrücke berechnen?

- Ziel 2: Wie geht das effizient, auch wenn der sprachliche Ausdruck mehrdeutig ist?

CL-Techniken: Ziele

- Ziel 1: Wie kann man die Struktur sprachlicher Ausdrücke berechnen?

- Ziel 2: Wie geht das effizient, auch wenn der sprachliche Ausdruck mehrdeutig ist?

- Ziel 3: Wie erkennt man unter allen möglichen Lesarten die richtige?

Struktur der Vorlesung

- Teil 1: Parsingalgorithmen für kf. Grammatiken
  - Recursive Descent, Shift-Reduce, CKY, Earley
  - Parsing-Schemata
- Teil 2: Elementare statistische Verfahren
  - n-Gramm-Modelle
  - Hidden Markov Models
  - Probabilistische kontextfreie Grammatiken
- Teil 3: Weiterführende Themen
  - Semantik
  - Maschinelles Lernen

Vorlesungsstil

- Ich bin nicht nur zum Reden da, sondern auch, um Fragen zu beantworten.

- Bitte unterbrechen Sie mich, wenn Sie irgendetwas nicht verstehen.

- Es gibt keine dummen Fragen.

Übungen

- Regelmäßig sind Übungen zu Hause zu bearbeiten; Bearbeitungszeit ca. 10 Tage, Abgabe per Mail.

- Etwa jede dritte Sitzung ist Übungssitzung. Dort rechnen Sie Ihre Lösungen vor und diskutieren Probleme.

- Sie können Aufgaben untereinander diskutieren, müssen aber Lösung alleine formulieren und aufschreiben.

Prüfungsleistung

- Es gibt 8 Übungen.
  - 1. Semesterhälfte: Ü 1-4
  - 2. Semesterhälfte: Ü 5-8
  - Jede Übung gibt 100 Punkte. Für jeden Teilnehmer zähle ich nur die drei besten Übungen in jeder Semesterhälfte.

- Anforderungen an Übungen:
  - In jeder Semesterhälfte mindestens drei Übungen bearbeiten.
  - In den drei besten mindestens 150 Punkte bekommen.
  - Pro Semesterhälfte mindestens einmal eine Aufgabe in der Übungssitzung vorrechnen.

Prüfungsleistung

- 30% der Note bestehen aus Übungspunkten:
  - Durchschnitt aus den drei besten Übungen in jeder Semesterhälfte.
  - Durchschnitt 50 Punkte = Note 4,0.
- 70% der Note kommt aus dem Abschlussprojekt.
  - Bearbeitung in den Semesterferien

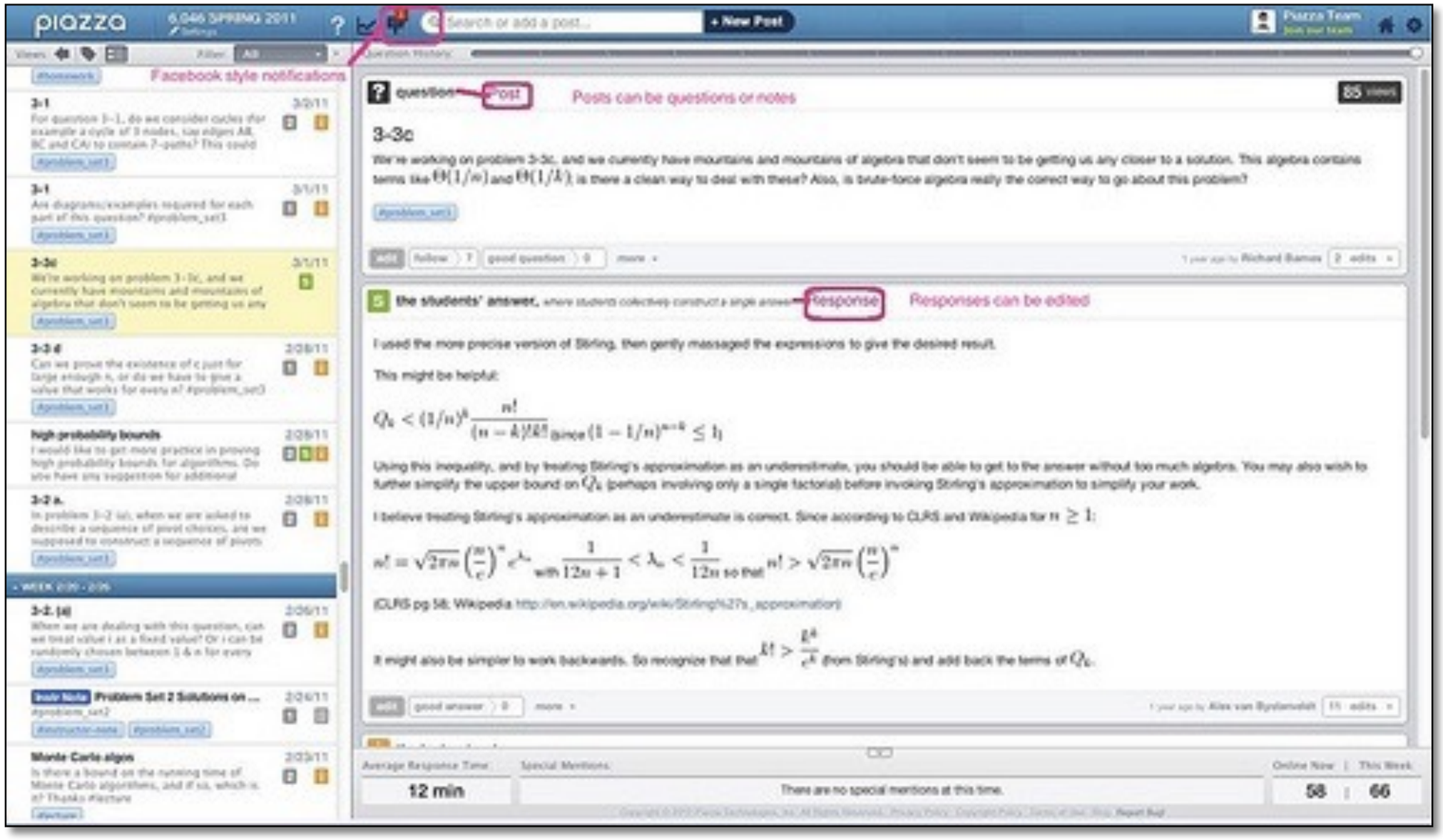
Tutorium

- Um Sie in Ihrer Arbeit zu unterstützen, bieten wir jede Woche ein freiwilliges Tutorium an.

- Zweck: Rückfragen bei inhaltlichen Unklarheiten, Hilfestellung bei der Bearbeitung der Übungen.

- Tutor: Marius Gerdes
  Macht nachher einen Termin mit Ihnen aus.

Piazza



https://piazza.com/class/iffgbwnws3l6x8

- Noch Fragen?

Teil 1

- Kontextfreie Grammatiken
- Verschiedene Parsingalgorithmen:
  - Backtracking: Recursive Descent, Shift-Reduce
  - Chartparsing: CKY, Earley
  - Parsingschemata
- Erweiterung auf kf. Grammatiken mit Features.

Literatur

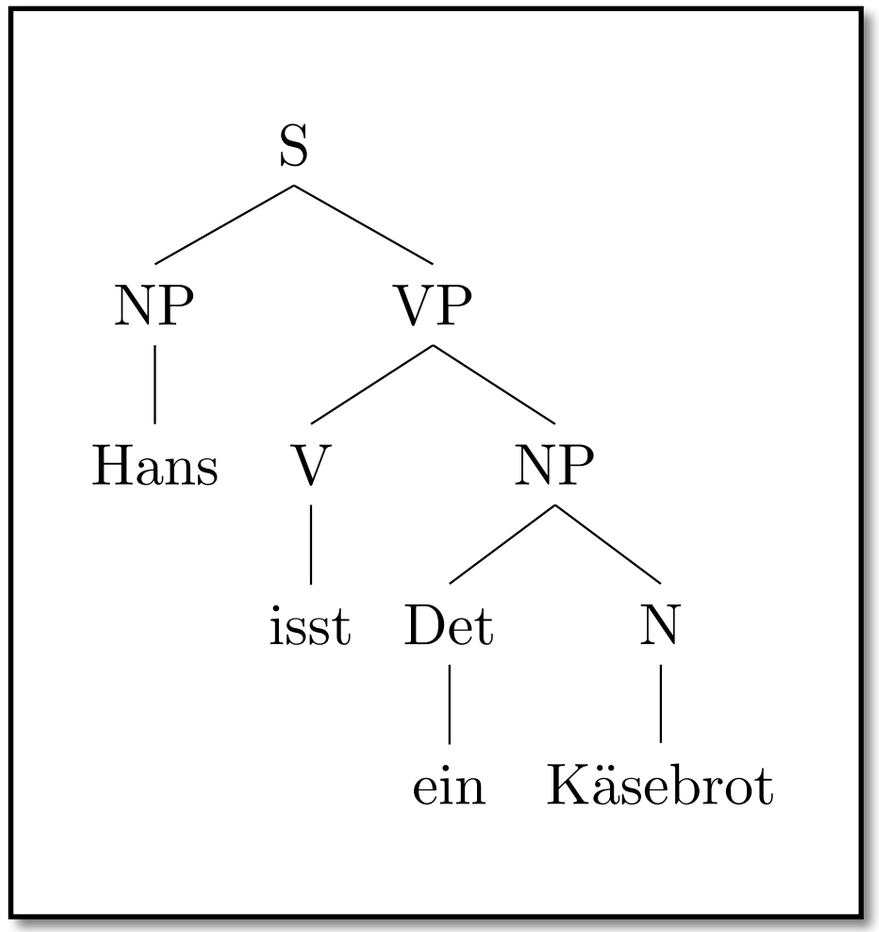
- Standard-Lehrbücher (empfehle ich jedem CL-Studenten zur Anschaffung):
  - Jurafsky & Martin, Speech and Language Processing
  - Manning & Schütze, Foundations of Statistical NLP

- Zum Thema Parsing gibt es auch deutsche Lehrbücher (hilft nur für die ersten paar Wochen).

- Im Internet finden Sie auch Folien und Tutorials von anderen Leuten.

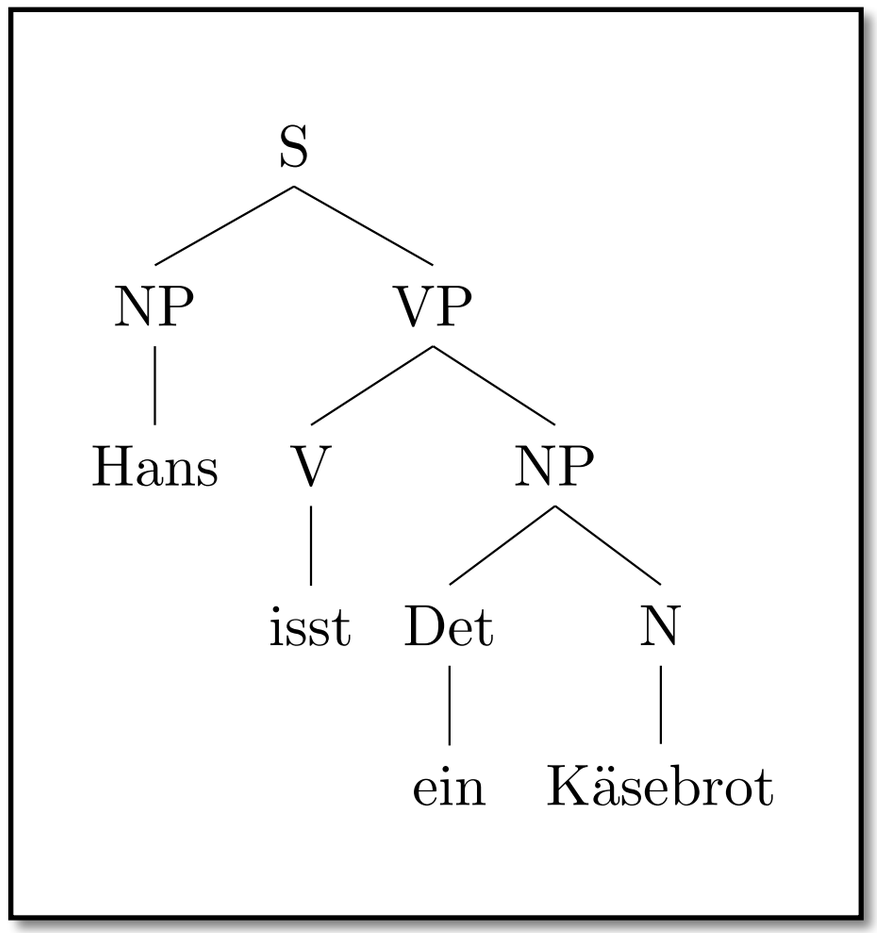
Kontextfreie Grammatiken

- Ein beliebter Ansatz für Syntax: Phrasenstruktur-Bäume.



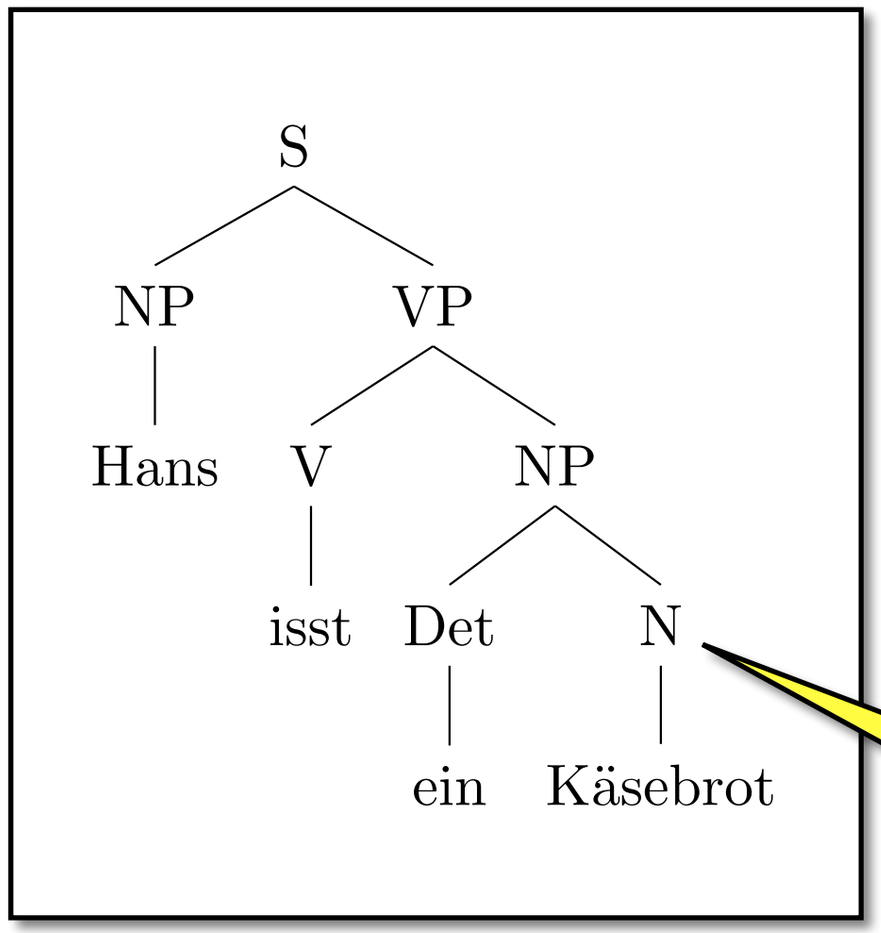
Kontextfreie Grammatiken

- Mit einer kontextfreien Grammatik (kfG) kann man “korrekte” PSG-Bäume beschreiben.



Kontextfreie Grammatiken

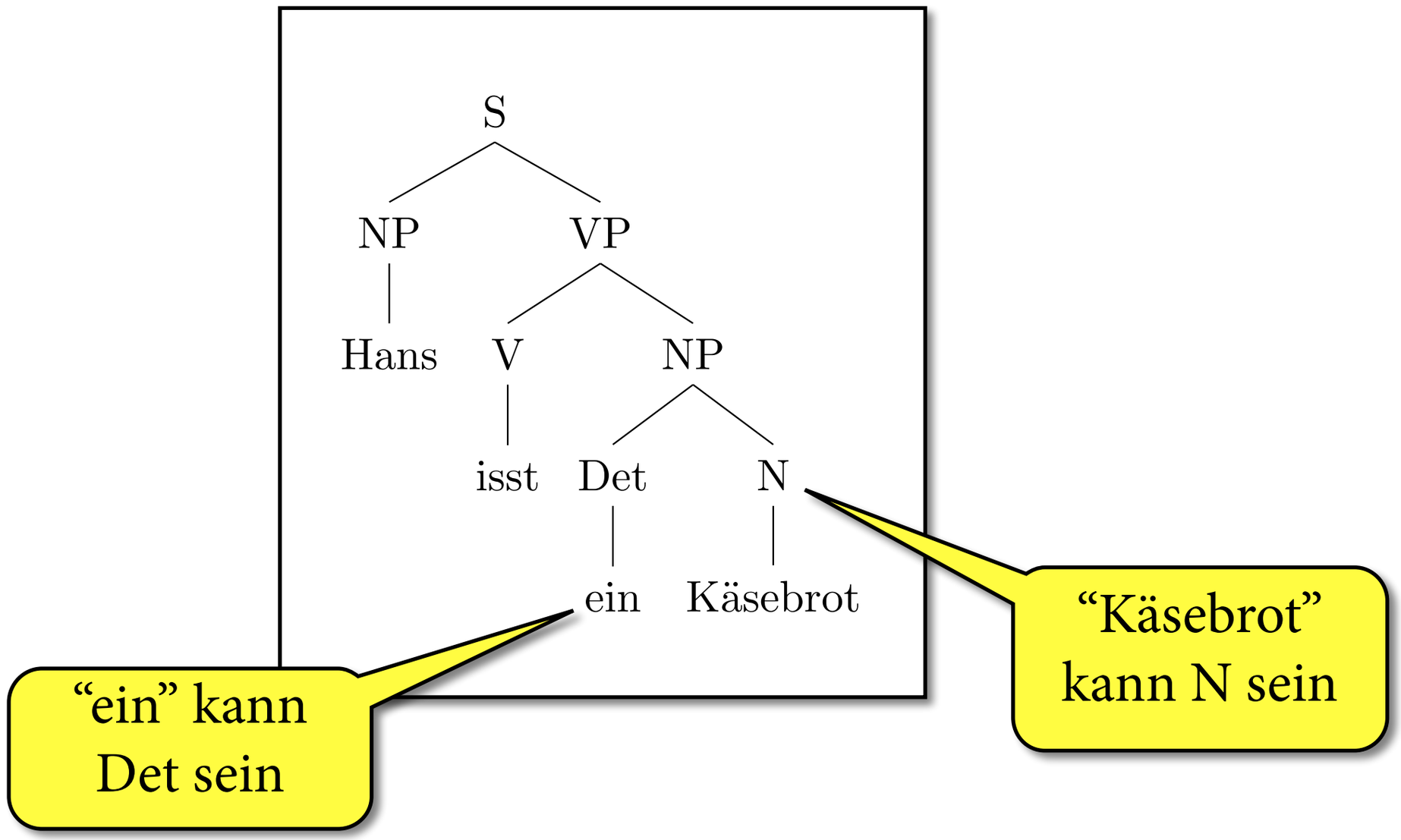
- Mit einer kontextfreien Grammatik (kfG) kann man “korrekte” PSG-Bäume beschreiben.



“Käsebrot” kann N sein

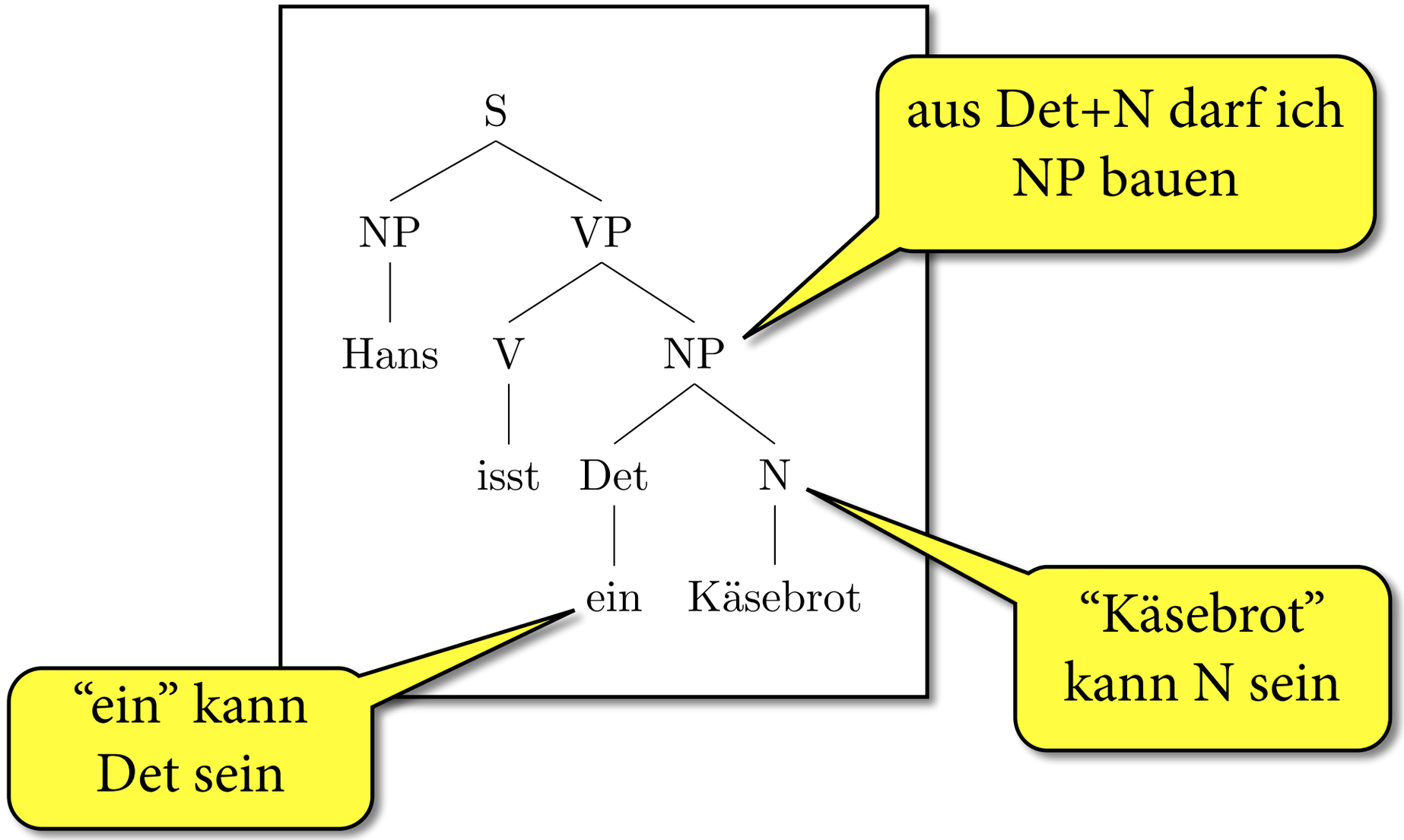
Kontextfreie Grammatiken

- Mit einer kontextfreien Grammatik (kfG) kann man “korrekte” PSG-Bäume beschreiben.



Kontextfreie Grammatiken

- Mit einer kontextfreien Grammatik (kfG) kann man “korrekte” PSG-Bäume beschreiben.



kfGs: Definition

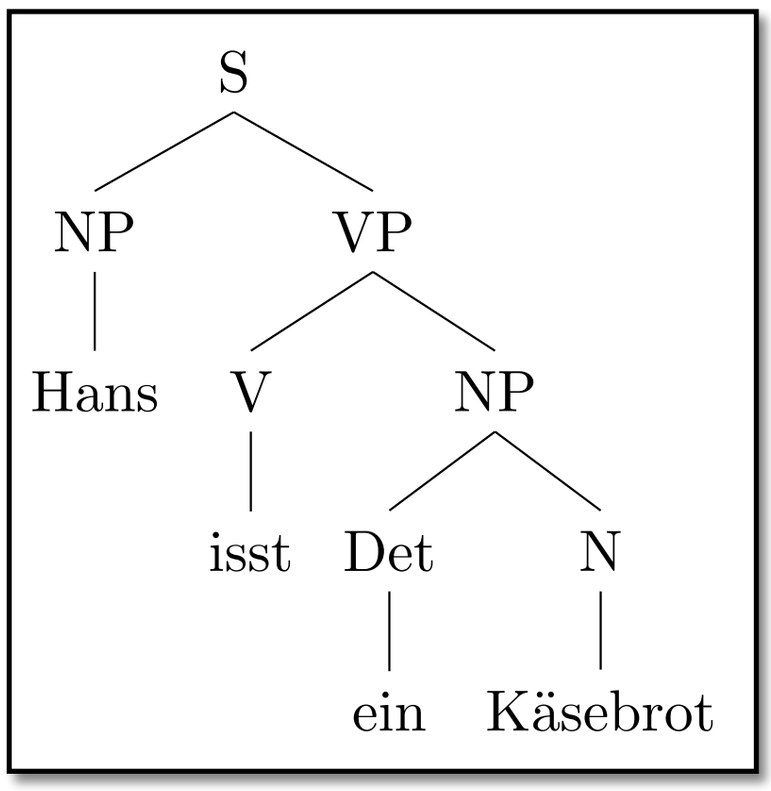
- Formal ist eine kfG G ein 4-Tupel (N,T,S,P):
  - N und T sind disjunkte endliche Mengen von Symbolen (T = Terminalsymbole; N = Nichtterminalsymbole)
  - $S \in N$ ist das Startsymbol
  - P ist eine endliche Menge von Produktionsregeln der Form $A \rightarrow w$, wobei A ein Nichtterminal und w ein Wort aus $(N \cup T)^*$ ist.

Ein Beispiel

T = {Hans, isst, Käsebrot, ein}
N = {S, NP, VP, V, N, Det}; Startsymbol: S

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP
V → isst
NP → Hans
Det → ein
N → Käsebrot



Ableitungen von kfGs

- Eine kfG G ist ein Werkzeug, um eine (evtl. unendliche) Sprache L(G) von Wörtern über T zu beschreiben.

- Dazu fängt man mit dem Startsymbol S an und wendet so lange Produktionsregeln an, bis keine Symbole aus N mehr da stehen.

Ableitungen von kfGs

- Ableitungsrelation $\Rightarrow$:

  $w_1 A w_2 \Rightarrow w_1 w w_2$ gdw $A \rightarrow w$ in P

  (w_1, w_2, w sind Wörter aus $(N \cup T)^*$)

- Reflexive, transitive Hülle $\Rightarrow^*$:

  $w \Rightarrow^* w_n$ falls $w \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_n$ (für $n \geq 0$)

- Sprache $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP

Beispiel

Produktionsregeln:

S → NP VP	V → isst	Det → ein
NP → Det N	NP → Hans	N → Käsebrot
VP → V NP		

S ⇒ NP VP ⇒ Hans VP

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP ⇒ Hans isst Det N

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP ⇒ Hans isst Det N

⇒ Hans isst ein N

Beispiel

Produktionsregeln:

S → NP VP
NP → Det N
VP → V NP

V → isst
NP → Hans

Det → ein
N → Käsebrot

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP ⇒ Hans isst Det N

⇒ Hans isst ein N ⇒ Hans isst ein Käsebrot

Beispiel

Produktionsregeln:

S → NP VP	V → isst	Det → ein
NP → Det N	NP → Hans	N → Käsebrot
VP → V NP		

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP ⇒ Hans isst Det N

⇒ Hans isst ein N ⇒ Hans isst ein Käsebrot

Also gilt S ⇒* “Hans isst ein Käsebrot”.

Beispiel

Produktionsregeln:

S → NP VP	V → isst	Det → ein
NP → Det N	NP → Hans	N → Käsebrot
VP → V NP		

S ⇒ NP VP ⇒ Hans VP ⇒ Hans V NP

⇒ Hans isst NP ⇒ Hans isst Det N

⇒ Hans isst ein N ⇒ Hans isst ein Käsebrot

Also gilt S ⇒* “Hans isst ein Käsebrot”.

Also ist “Hans isst ein Käsebrot” in L(G).

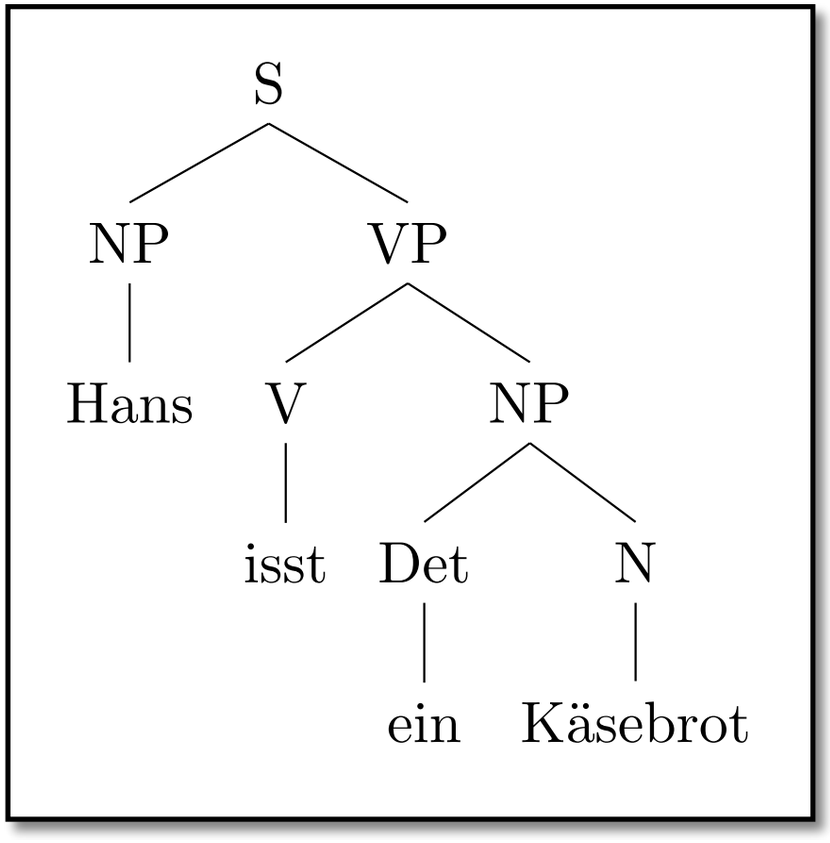
Parsebäume

- Die Struktur einer Ableitung gibt man gut lesbar in einem Parsebaum an.

Ableitung

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

Parsebaum



Parsing

- Gegeben seien eine kfG G und ein Wort w.

- Das *Wortproblem* ist die Frage, ob $w \in L(G)$ ist. Wortproblem wird von *Erkenner* gelöst.

- Das *Parsingproblem* ist das Problem, alle Parsebäume von w bzgl. G zu bestimmen. Parsingproblem wird von *Parser* gelöst.

- Jeder Parser löst Wortproblem mit.

Parsing: Beispiele

G

$S \rightarrow a\ S\ b$

$S \rightarrow a\ b$

Ist $aabb \in L(G)$?

Wenn ja, was sind die Parsebäume?

Parsing: Beispiele

G

$S \rightarrow a\ S\ b$

$S \rightarrow a\ b$

Ist $abab \in L(G)$?

Wenn ja, was sind die Parsebäume?

Parsing: Beispiele

G

$S \rightarrow S\ S$

$S \rightarrow a$

Ist aaa $\in$ L(G)?

Wenn ja, was sind die Parsebäume?

Parsing: Beispiele

G

$S \rightarrow S\ S$

$S \rightarrow a$

Ist aaa $\in$ L(G)?
Wenn ja, was sind die Parsebäume?

Ist aaaa $\in$ L(G)?
Wenn ja, was sind die Parsebäume?

Syntaktische Ambiguitäten

- Manche kfGs erlauben Ambiguität:
  Der gleiche String hat mehrere Parsebäume.
  - im worst case: exponentiell viele

- Das ist wichtig, weil Sätze wirklich syntaktisch mehrdeutig sein können.

- ... wird aber immer wieder sehr unbequem für uns sein.

Zusammenfassung

- Überblick
- Kontextfreie Grammatiken
- Wort- und Parsingproblem
- nächstes Mal: Top-Down und Bottom-Up Parsing.