

Syntax natürlicher Sprachen

Vorlesung 9: Parsing-Algorithmen

A. Wisiorek

Folien: Martin Schmitt

Centrum für Informations- und Sprachverarbeitung,
Ludwig-Maximilians-Universität München

14.12.2021

Arten von Parsing-Algorithmen

Top-Down

- **Recursive Descent**

- <https://www.nltk.org/book/ch08.html#recursive-descent-parsing>

- LL (Left-to-right Leftmost (derivation))

- LL(k)

- L(*)

- **Earley**

- <https://www.nltk.org/book/ch08-extras.html#the-earley-algorithm>

Bottom-Up

- Recursive Ascent

- GLR (Generalized Left-to-right Rightmost (derivation))

- **Shift-Reduce**

- <https://www.nltk.org/book/ch08.html#shift-reduce-parsing>

- CYK

1. Top-Down-Parsing: Recursive Descent

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus
- 4 Parsing mit Merkmalstrukturen
 - Modifizierter Earley Parser
 - Komplexität

Top-Down-Parsing (dt. *Abwärtsparsen*)

Parsing-Strategie, bei der man von der höchsten Ebene eines Syntaxbaums (Startsymbol der Grammatik) ausgeht und sich mithilfe der Ersetzungsregeln (Produktionsregeln) einer Grammatik bis zu den Terminalen (Lexemen) vorarbeitet.

Recursive Descent Parsing (dt. *rekursiver Abstieg*)

- Form von Top-Down-Parsing
- probiert jede anwendbare Regel aus
- benutzt *Backtracking* im Problemfall
- am intuitivsten „händisch“ zu programmieren
- kann je nach Grammatik zu exponentieller Laufzeit führen (oder sogar zu unendlich langer Laufzeit)
- **2 Operationen: PREDICT (*EXPAND*) + SCAN (*MATCH*)**

Recursive Descent Parser: Beispiel

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Chomsky kennt das Buch

Initialisierung mit Startsymbol

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROPN$

④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROPN \rightarrow \text{Chomsky}$

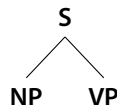
⑧ $V \rightarrow \text{kennt}$

S

Chomsky kennt das Buch

PREDICT (Ableitung)

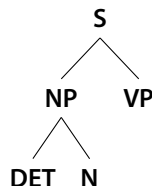
- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

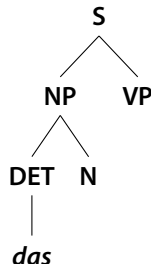
PREDICT (zunächst jeweils 1. Regel für eine LHS)

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

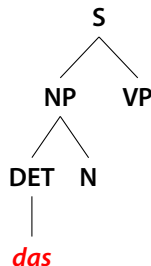
- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ **$DET \rightarrow \text{das}$**
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

SCAN (Abgleich mit Satz)

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

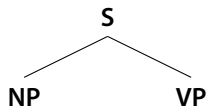


Chomsky kennt das Buch

⚡ (kein Match!)

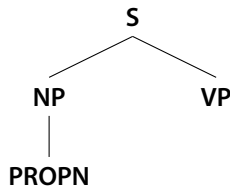
Recursive Descent Parser: Backtracking

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



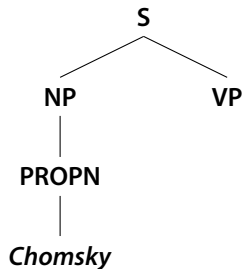
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



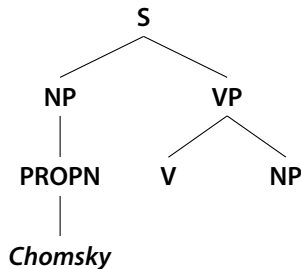
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



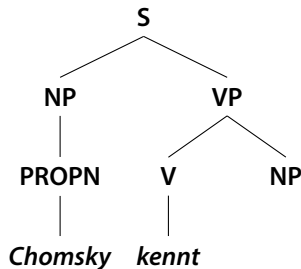
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



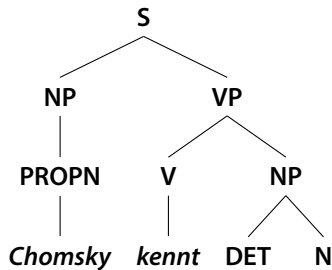
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



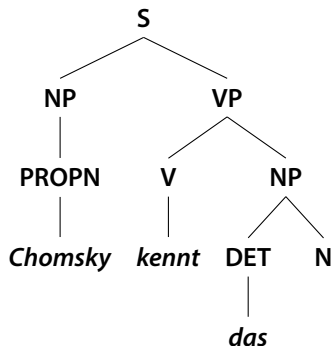
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



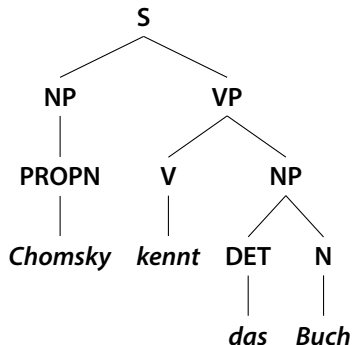
Chomsky kennt das Buch

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ **DET \rightarrow das**
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

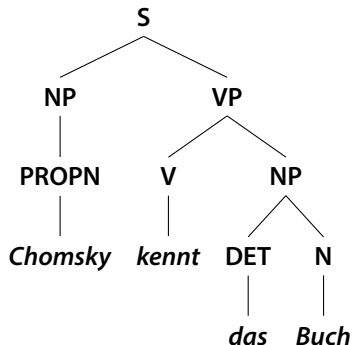
- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

Recursive Descent Parser: erfolgreicher Parse

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch



Probleme

- Es kann zu jeder Zeit für ein Nichtterminal viele verschiedene Ersetzungsregeln geben.
- Im schlimmsten Fall müssen alle diese Regeln ausprobiert werden (exponentieller Blow-up).
- Viele Teilstrukturen werden erzeugt, obwohl sie nie erfolgreich sein können.
 - Bsp.: Eingabesatz enthält gar nicht die passenden Wörter.

Gefahr der Endlosschleife

- *Links-rekursive* Produktionsregeln führen (bei naiver Ausführung) zu unendlicher Laufzeit!
- Beispiel: $NP \rightarrow NP PP$

2. Bottom-Up-Parsing: Shift Reduce

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce**
- 3 Chart Parsing: Earley Algorithmus
- 4 Parsing mit Merkmalstrukturen
 - Modifizierter Earley Parser
 - Komplexität

Bottom-Up-Parsing (dt. *Aufwärtsparsen*)

Parsing-Strategie, bei der man von den kleinsten vorgefundenen Einheiten (Token, Lexeme, Terminale) ausgeht und versucht, diese nach und nach zu größeren syntaktischen Strukturen zu verbinden, bis man beim Startsymbol der Grammatik angelangt ist.

Shift Reduce Parsing (dt. *Verschieben – Zurückführen*)

- Form von Bottom-Up-Parsing (*datengeleitetes Parsing*)
- gebraucht die Datenstruktur **Stack** (dt. **Stapel**)
- **verschiebt** Token auf den Stapel, um sie auf Grammatikregeln **zurückzuführen**
- **2 Operationen: SHIFT + REDUCE**

Shift Reduce Parser: Beispiel

Grammatik

Stapel

Ableitungsbaum

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROPN$

④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROPN \rightarrow \text{Chomsky}$

⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

Initialisierung mit leerem Stack

Grammatik

Stapel

Ableitungsbaum

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROPN$

④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROPN \rightarrow \text{Chomsky}$

⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

SHIFT (Input auf Stack verschieben)

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Chomsky

Chomsky

Chomsky kennt das Buch

REDUCE (Ersatz top-Stack-Items mit LHS von Regel, deren RHS diese matchen)

Grammatik

Stapel

Ableitungsbaum

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROPN$

④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROPN \rightarrow \text{Chomsky}$

⑧ $V \rightarrow \text{kennt}$

PROPN

PROPN

Chomsky

Chomsky kennt das Buch

REDUCE

Grammatik

Stapel

Ableitungsbaum

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROP$

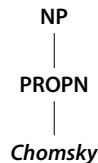
④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROP \rightarrow \text{Chomsky}$

⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

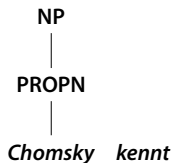
SHIFT (kein REDUCE mehr möglich: keine Regel mit NP als RHS = right-hand-side)

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

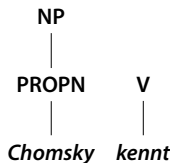
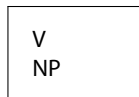
REDUCE (Ergebnis auf Stack: NP-V; V = letztes Element = top-Stack-Item)

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



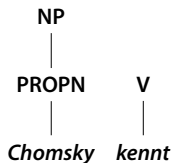
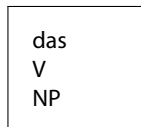
Chomsky kennt das Buch

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



das

Chomsky kennt das Buch

REDUCE

Grammatik

Stapel

Ableitungsbaum

① $S \rightarrow NP VP$

② $NP \rightarrow DET N$

③ $NP \rightarrow PROPN$

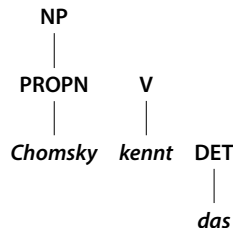
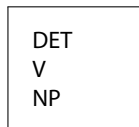
④ $VP \rightarrow V NP$

⑤ $DET \rightarrow \text{das}$

⑥ $N \rightarrow \text{Buch}$

⑦ $PROPN \rightarrow \text{Chomsky}$

⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

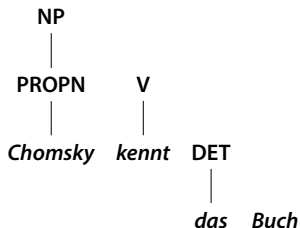
Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Buch
DET
V
NP



Chomsky kennt das Buch

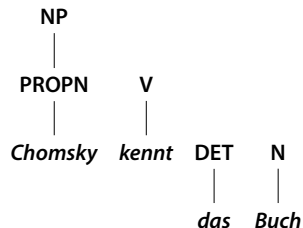
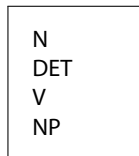
REDUCE

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



Chomsky kennt das Buch

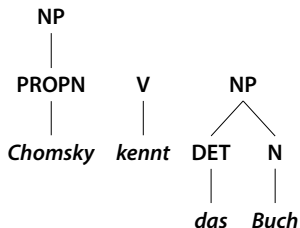
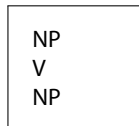
REDUCE

Grammatik

Stapel

Ableitungsbaum

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$



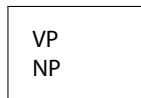
Chomsky kennt das Buch

REDUCE

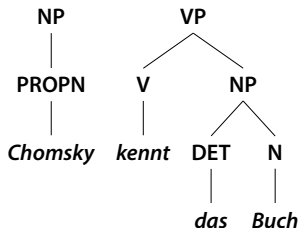
Grammatik

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ **$VP \rightarrow V NP$**
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Stapel



Ableitungsbaum



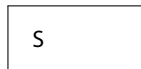
Chomsky kennt das Buch

REDUCE

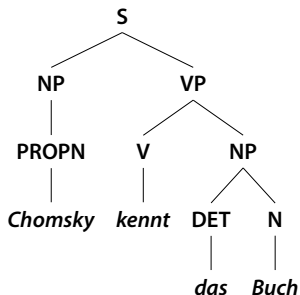
Grammatik

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Stapel



Ableitungsbaum



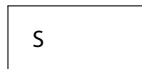
Chomsky kennt das Buch

Shift Reduce Parser: erfolgreicher Parse

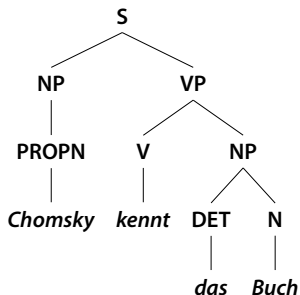
Grammatik

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Stapel



Ableitungsbaum



Chomsky kennt das Buch



Vorteile

- arbeitet abhängig von der Eingabe
- ist daher effizienter als ein Top-Down-Parser

Probleme

- erzeugt auch Teilstrukturen, die zu keinem Ergebnis führen
 - benötigt also im Allgemeinen auch Backtracking
- potentiell exponentielle Laufzeit

Top-Down vs. Bottom-Up

Top-Down

- startet die Analyse beim Startsymbol
- alterniert zwischen Regelanwendung (*Predict*) und Abgleich mit der Eingabe (*Scan*)
- geht besser mit POS Ambiguitäten um
- baut Strukturen öfter als benötigt
- verbringt viel Zeit mit unmöglichen Ableitungen

Bottom-Up

- startet die Analyse beim Beginn der Eingabe
- alterniert zwischen Einlesen der Eingabe (*Shift*) und „Rückwärtsanwendung“ der Regeln (*Reduce*)
- muss alle lexikalische Ambiguitäten berücksichtigen
- baut benötigte Strukturen nur einmal
- verbringt viel Zeit mit unnötigen Strukturen

3. Chart Parsing: Earley Algorithmus

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus**
- 4 Parsing mit Merkmalstrukturen
 - Modifizierter Earley Parser
 - Komplexität

Chart Parsing

- *Dynamische Programmierung* vermeidet doppelte Berechnungen.
- Zwischenergebnisse werden in Datenstruktur (*Chart*) gespeichert
- s. auch: <https://www.nltk.org/book/ch08-extras.html#chart-parsing>

Earley Parsing

- Top-Down-Parser (ohne Backtracking)
 - Algorithmus kann eigentlich nur Grammatikalität entscheiden.
- Zur Baumerstellung müssen zusätzliche Verweise gespeichert werden.
- Komplexität: $\mathcal{O}(n^3)$
 - funktioniert nur mit ε -freien Grammatiken!
 - **3 Operationen: PREDICT + SCAN + COMPLETE**

ε -Regel

- Regel der Form $A \rightarrow \varepsilon$ (Nichtterminal A wird gelöscht)
- Im nltk-Format $A \rightarrow$ (z. B. für optionale Elemente)

Eliminierungsalgorithmus

- 1 Wähle ein Nichtterminal A mit einer ε -Regel
- 2 Entferne die ε -Regel
- 3 Für jede Regel p mit A auf der rechten Seite:
dupliziere die Regel für jede mögliche Kombination mit/ohne A
($2^{\text{Anzahl der Vorkommen von } A \text{ in } p}$ neue Regeln)
- 4 Falls es immer noch ε -Regeln gibt, gehe zurück zu Schritt 1.

Beispiel (Leeres Subjekt bei Imperativ)

- 1 | $S \rightarrow NP \ VP$
- 2 | $NP \rightarrow \epsilon$
- 3 | $NP \rightarrow DET \ N$
- 4 | $VP \rightarrow V$
- 5 | $V \rightarrow \text{"schlaf"} \mid \text{"schläft"}$
- 6 | $DET \rightarrow \text{"der"}$
- 7 | $N \rightarrow \text{"Hund"}$

Zur Vermeidung von Übergenerierung fehlen noch entsprechende Bedingungen! (s. Hausaufgabe 8)

Beispiel (nach Eliminierung)

```
1 | S → NP VP
2 | S → VP
3 | NP → DET N
4 | VP → V
5 | V → "schlaf" | "schläft"
6 | DET → "der"
7 | N → "Hund"
```

Zur Vermeidung von Übergenerierung fehlen noch entsprechende Bedingungen! (s. Hausaufgabe 8)

Gegeben

Eingabesequenz $s = s_1, \dots, s_n$; Grammatik $G = (T, N, P, S)$

Datenstrukturen

- Position := Tokengrenze
(z. B. zwischen s_1 und s_2 etc.)
- Zu jeder Pos. Menge Q von *Zuständen*
- Zustand := $(X \rightarrow \alpha \cdot \beta, i)$
bestehend aus
 - der aktuellen Produktionsregel $X \rightarrow \alpha\beta \in P$,
 - der aktuellen Position in dieser Regel (der Punkt \cdot),
 - der Ursprungsposition i in der Eingabe, an der das Abgleichen dieser Regel begann.

Operationen

P Prediction (dt. *Voraussage*)

falls $(A \rightarrow \dots \cdot B \dots, j) \in Q_i$ mit $B \in N$, dann für jede Regel $B \rightarrow \alpha \in P$:

setze $(B \rightarrow \cdot \alpha, i) \in Q_i$

S Scanning (dt. *Überprüfung*)

falls $(A \rightarrow \dots \cdot a \dots, j) \in Q_i$ mit $a \in T$ und $a = s_{i+1}$, dann

setze $(A \rightarrow \dots a \cdot \dots, j) \in Q_{i+1}$

C Completion (dt. *Vervollständigung*)

falls $(A \rightarrow \dots \cdot, j) \in Q_i$, dann für alle Zustände

$(B \rightarrow \dots \cdot A \dots, k) \in Q_j$:

setze $(B \rightarrow \dots A \cdot \dots, k) \in Q_i$

Earley Algorithmus III

PREDICT

wenn . vor Nichtterminal (N)

SCAN

wenn . vor Terminal (T)

COMPLETE

wenn . letzte Position

Algorithmus

- 1 Initialisiere Q_0 mit dem Zustand $(S' \rightarrow \cdot S, 0)$ mit S' frisches nichtterminales Symbol
- 2 Führe je nach Situation eine der drei Operationen (P, S, C) aus, bis keine weiteren Zustände mehr hinzugefügt werden können.
- 3 Wiederhole Schritt 2 bis keine neuen Zustände mehr hinzugefügt werden können.
- 4 Akzeptiere die Eingabesequenz s genau dann, wenn $(S' \rightarrow S \cdot, 0) \in Q_{|s|}$

\Rightarrow Beispiel auf der nächsten Folie

Earley Parser: Beispiel (Initialisierung)

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q_0

$(S' \rightarrow \cdot S, 0)$

Q_1

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q₀

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

Q₁

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q₀

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

Q₁

0 Chomsky 1 kennt 2 das 3 Buch 4

Q_0 : PREDICT (jeweils alle Möglichkeiten für eine LHS)

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q_0

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

Q_1

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q₀

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

Q₁

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

Q₀

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

$(PROPN \rightarrow \cdot \text{Chomsky}, 0)$

Q₁

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

- ① $S \rightarrow NP VP$
- ② $NP \rightarrow DET N$
- ③ $NP \rightarrow PROPN$
- ④ $VP \rightarrow V NP$
- ⑤ $DET \rightarrow \text{das}$
- ⑥ $N \rightarrow \text{Buch}$
- ⑦ $PROPN \rightarrow \text{Chomsky}$
- ⑧ $V \rightarrow \text{kennt}$

Pos. Zustände

 Q_0 $(S' \rightarrow \cdot S, 0)$ $(S \rightarrow \cdot NP VP, 0)$ $(NP \rightarrow \cdot DET N, 0)$ $(NP \rightarrow \cdot PROPN, 0)$ $(DET \rightarrow \cdot \text{das}, 0)$ $(PROPN \rightarrow \cdot \text{Chomsky}, 0)$ Q_1 $(PROPN \rightarrow \text{Chomsky} \cdot, 0)$

0 **Chomsky** 1 kennt 2 das 3 Buch 4

Grammatik:(4) $VP \rightarrow V NP$ (8) $V \rightarrow \text{kennt}$ Q₀: $(S' \rightarrow \cdot S, 0)$ $(S \rightarrow \cdot NP VP, 0)$ $(NP \rightarrow \cdot DET N, 0)$ $(NP \rightarrow \cdot PROPN, 0)$ $(DET \rightarrow \cdot \text{das}, 0)$ $(PROPN \rightarrow \cdot \text{Chomsky}, 0)$

Pos. Zustände

Q₁ $(PROPN \rightarrow \text{Chomsky} \cdot, 0)$ Q₂o Chomsky₁ kennt₂ das₃ Buch₄

Q₁: COMPLETION

Grammatik:

(4) $VP \rightarrow V NP$

(8) $V \rightarrow \text{kennt}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROP N, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

$(PROP N \rightarrow \cdot \text{Chomsky}, 0)$

Pos. Zustände

Q₁

$(PROP N \rightarrow \text{Chomsky} \cdot, 0)$

$(NP \rightarrow PROP N \cdot, 0)$

Q₂

0 Chomsky 1 kennt 2 das 3 Buch 4

Q₁: COMPLETION

Grammatik:

(4) $VP \rightarrow V NP$

(8) $V \rightarrow \text{kennt}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

$(PROPN \rightarrow \cdot \text{Chomsky}, 0)$

Pos. Zustände

Q₁

$(PROPN \rightarrow \text{Chomsky } \cdot, 0)$

$(NP \rightarrow PROPN \cdot, 0)$

$(S \rightarrow NP \cdot VP, 0)$

Q₂

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(4) $VP \rightarrow V NP$

(8) $V \rightarrow \text{kennt}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

$(PROPN \rightarrow \cdot \text{Chomsky}, 0)$

Pos. Zustände

Q₁

$(PROPN \rightarrow \text{Chomsky} \cdot, 0)$

$(NP \rightarrow PROPN \cdot, 0)$

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q₂

o Chomsky₁ kennt₂ das₃ Buch₄

Grammatik:

(4) $VP \rightarrow V NP$

(8) $V \rightarrow \text{kennt}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

$(S \rightarrow \cdot NP VP, 0)$

$(NP \rightarrow \cdot DET N, 0)$

$(NP \rightarrow \cdot PROPN, 0)$

$(DET \rightarrow \cdot \text{das}, 0)$

$(PROPN \rightarrow \cdot \text{Chomsky}, 0)$

Pos. Zustände

Q₁

$(PROPN \rightarrow \text{Chomsky} \cdot, 0)$

$(NP \rightarrow PROPN \cdot, 0)$

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

$(V \rightarrow \cdot \text{kennt}, 1)$

Q₂

o Chomsky₁ kennt₂ das₃ Buch₄

Grammatik:(4) $VP \rightarrow V NP$ (8) $V \rightarrow \text{kennt}$ Q_0 : $(S' \rightarrow \cdot S, 0)$ $(S \rightarrow \cdot NP VP, 0)$ $(NP \rightarrow \cdot DET N, 0)$ $(NP \rightarrow \cdot PROPN, 0)$ $(DET \rightarrow \cdot \text{das}, 0)$ $(PROPN \rightarrow \cdot \text{Chomsky}, 0)$ Pos. Zustände Q_1 $(PROPN \rightarrow \text{Chomsky} \cdot, 0)$ $(NP \rightarrow PROPN \cdot, 0)$ $(S \rightarrow NP \cdot VP, 0)$ $(VP \rightarrow \cdot V NP, 1)$ $(V \rightarrow \cdot \text{kennt}, 1)$ Q_2 $(V \rightarrow \text{kennt} \cdot, 1)$ o Chomsky₁ kennt₂ das₃ Buch₄

Grammatik:

(2) NP → DET N

(3) NP → PROPN

(5) DET → das

(7) PROPN → Chomsky

Q₁:

(PROPN → Chomsky ·, 0)

(NP → PROPN ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos. Zustände

Q₂

(V → kennt ·, 1)

Q₃o Chomsky₁ kennt₂ das₃ Buch₄

Q₂: COMPLETION

Grammatik:

(2) NP → DET N

(3) NP → PROPN

(5) DET → das

(7) PROPN → Chomsky

Q₁:

(PROPN → Chomsky ·, 0)

(NP → PROPN ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos.	Zustände
Q ₂	
	(V → kennt ·, 1)
	(VP → V · NP, 1)
Q ₃	

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(2) NP → DET N

(3) NP → PROPN

(5) DET → das

(7) PROPN → Chomsky

Q₁:

(PROPN → Chomsky ·, 0)

(NP → PROPN ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos. Zustände

Q₂

(V → kennt ·, 1)

(VP → V · NP, 1)

(NP → · DET N, 2)

Q₃

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(2) NP → DET N

(3) NP → PROP_N

(5) DET → das

(7) PROP_N → Chomsky

Q₁:

(PROP_N → Chomsky ·, 0)

(NP → PROP_N ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos. Zustände

Q₂

(V → kennt ·, 1)

(VP → V · NP, 1)

(NP → · DET N, 2)

(NP → · PROP_N, 2)

Q₃

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(2) NP → DET N

(3) NP → PROPN

(5) DET → das

(7) PROPN → Chomsky

Q₁:

(PROPN → Chomsky ·, 0)

(NP → PROPN ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos. Zustände

Q₂

(V → kennt ·, 1)

(VP → V · NP, 1)

(NP → · DET N, 2)

(NP → · PROPN, 2)

(DET → · das, 2)

Q₃

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(2) NP → DET N

(3) NP → PROPN

(5) DET → das

(7) **PROPN → Chomsky**

Q₁:

(PROPN → Chomsky ·, 0)

(NP → PROPN ·, 0)

(S → NP · VP, 0)

(VP → · V NP, 1)

(V → · kennt, 1)

Pos. Zustände

Q₂

(V → kennt ·, 1)

(VP → V · NP, 1)

(NP → · DET N, 2)

(NP → · PROPN, 2)

(DET → · das, 2)

(PROPN → · Chomsky, 2)

Q₃

o Chomsky₁ kennt₂ das₃ Buch₄

Grammatik:(2) NP \rightarrow DET N(3) NP \rightarrow PROPN(5) DET \rightarrow das(7) PROPN \rightarrow ChomskyQ₁:(PROPN \rightarrow Chomsky \cdot , 0)(NP \rightarrow PROPN \cdot , 0)(S \rightarrow NP \cdot VP, 0)(VP $\rightarrow \cdot$ V NP, 1)(V $\rightarrow \cdot$ kennt, 1)Pos. ZuständeQ₂(V \rightarrow kennt \cdot , 1)(VP \rightarrow V \cdot NP, 1)(NP $\rightarrow \cdot$ DET N, 2)(NP $\rightarrow \cdot$ PROPN, 2)(DET $\rightarrow \cdot$ das, 2)(PROPN $\rightarrow \cdot$ Chomsky, 2)Q₃(DET \rightarrow das \cdot , 2)o Chomsky₁ kennt₂ **das**₃ Buch₄

Grammatik:(6) $N \rightarrow \text{Buch}$ Q₀: $(S' \rightarrow \cdot S, 0)$ Q₁: $(S \rightarrow NP \cdot VP, 0)$ $(VP \rightarrow \cdot V NP, 1)$ Q₂: $(VP \rightarrow V \cdot NP, 1)$ $(NP \rightarrow \cdot DET N, 2)$ $(DET \rightarrow \cdot \text{das}, 2)$ Pos. ZuständeQ₃ $(DET \rightarrow \text{das} \cdot, 2)$ Q₄o Chomsky₁ kennt₂ das₃ Buch₄

Q₃: COMPLETE

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

Q₁:

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q₂:

$(VP \rightarrow V \cdot NP, 1)$

$(NP \rightarrow \cdot \text{DET } N, 2)$

$(DET \rightarrow \cdot \text{das}, 2)$

Pos. Zustände

Q₃

$(DET \rightarrow \text{das} \cdot, 2)$

$(NP \rightarrow DET \cdot N, 2)$

Q₄

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(6) **N** → **Buch**

Q₀:

(S' → · S, 0)

Q₁:

(S → NP · VP, 0)

(VP → · V NP, 1)

Q₂:

(VP → V · NP, 1)

(NP → · DET N, 2)

(DET → · das, 2)

Pos. Zustände

Q₃

(DET → das ·, 2)

(NP → DET · N, 2)

(N → · Buch, 3)

Q₄

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:(6) $N \rightarrow \text{Buch}$ Q_0 : $(S' \rightarrow \cdot S, 0)$ Q_1 : $(S \rightarrow NP \cdot VP, 0)$ $(VP \rightarrow \cdot V NP, 1)$ Q_2 : $(VP \rightarrow V \cdot NP, 1)$ $(NP \rightarrow \cdot DET N, 2)$ $(DET \rightarrow \cdot \text{das}, 2)$ Pos. Zustände Q_3 $(DET \rightarrow \text{das} \cdot, 2)$ $(NP \rightarrow DET \cdot N, 2)$ $(N \rightarrow \cdot \text{Buch}, 3)$ Q_4 $(N \rightarrow \text{Buch} \cdot, 3)$ o Chomsky₁ kennt₂ das₃ Buch₄

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

Q₁:

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q₂:

$(VP \rightarrow V \cdot NP, 1)$

$(NP \rightarrow \cdot DET N, 2)$

$(DET \rightarrow \cdot \text{das}, 2)$

Pos. Zustände

Q₃

$(DET \rightarrow \text{das} \cdot, 2)$

$(NP \rightarrow DET \cdot N, 2)$

$(N \rightarrow \cdot \text{Buch}, 3)$

Q₄

$(N \rightarrow \text{Buch} \cdot, 3)$

$(NP \rightarrow DET N \cdot, 2)$

0 Chomsky 1 kennt 2 das 3 Buch 4

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

Q₁:

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q₂:

$(VP \rightarrow V \cdot NP, 1)$

$(NP \rightarrow \cdot DET N, 2)$

$(DET \rightarrow \cdot \text{das}, 2)$

Pos. Zustände

Q₃

$(DET \rightarrow \text{das} \cdot, 2)$

$(NP \rightarrow DET \cdot N, 2)$

$(N \rightarrow \cdot \text{Buch}, 3)$

Q₄

$(N \rightarrow \text{Buch} \cdot, 3)$

$(NP \rightarrow DET N \cdot, 2)$

$(VP \rightarrow V NP \cdot, 1)$

0 Chomsky 1 kennt 2 das 3 Buch 4

Q₄: COMPLETE

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q₀:

$(S' \rightarrow \cdot S, 0)$

Q₁:

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q₂:

$(VP \rightarrow V \cdot NP, 1)$

$(NP \rightarrow \cdot DET N, 2)$

$(DET \rightarrow \cdot \text{das}, 2)$

Pos. Zustände

Q₃

$(DET \rightarrow \text{das} \cdot, 2)$

$(NP \rightarrow DET \cdot N, 2)$

$(N \rightarrow \cdot \text{Buch}, 3)$

Q₄

$(N \rightarrow \text{Buch} \cdot, 3)$

$(NP \rightarrow DET N \cdot, 2)$

$(VP \rightarrow V NP \cdot, 1)$

$(S \rightarrow NP VP \cdot, 0)$

0 Chomsky 1 kennt 2 das 3 Buch 4

Q₄: COMPLETE

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q₀:

($S' \rightarrow \cdot S, 0$)

Q₁:

($S \rightarrow NP \cdot VP, 0$)

($VP \rightarrow \cdot V NP, 1$)

Q₂:

($VP \rightarrow V \cdot NP, 1$)

($NP \rightarrow \cdot DET N, 2$)

($DET \rightarrow \cdot \text{das}, 2$)

Pos. Zustände

Q₃

($DET \rightarrow \text{das} \cdot, 2$)

($NP \rightarrow DET \cdot N, 2$)

($N \rightarrow \cdot \text{Buch}, 3$)

Q₄

($N \rightarrow \text{Buch} \cdot, 3$)

($NP \rightarrow DET N \cdot, 2$)

($VP \rightarrow V NP \cdot, 1$)

($S \rightarrow NP VP \cdot, 0$)

($S' \rightarrow S \cdot, 0$)

o Chomsky₁ kennt₂ das₃ Buch₄

Earley Parser: erfolgreicher Parse

Grammatik:

(6) $N \rightarrow \text{Buch}$

Q_0 :

$(S' \rightarrow \cdot S, 0)$

Q_1 :

$(S \rightarrow NP \cdot VP, 0)$

$(VP \rightarrow \cdot V NP, 1)$

Q_2 :

$(VP \rightarrow V \cdot NP, 1)$

$(NP \rightarrow \cdot DET N, 2)$

$(DET \rightarrow \cdot \text{das}, 2)$

Pos. Zustände

Q_3

$(DET \rightarrow \text{das} \cdot, 2)$

$(NP \rightarrow DET \cdot N, 2)$

$(N \rightarrow \cdot \text{Buch}, 3)$

Q_4

$(N \rightarrow \text{Buch} \cdot, 3)$

$(NP \rightarrow DET N \cdot, 2)$

$(VP \rightarrow V NP \cdot, 1)$

$(S \rightarrow NP VP \cdot, 0)$

$(S' \rightarrow S \cdot, 0) \checkmark$

0 Chomsky 1 kennt 2 das 3 Buch 4

Top-Down-Parsing mit Extras

- Zwischenergebnisse werden in Datenstruktur (Chart) gespeichert (→ Chart-Parsing, Dynamische Programmierung)
 - Zustände werden mit Positionen in der Eingabesequenz abgeglichen (Elemente des Bottom-Up-Parsings)
- Komplizierter als *Recursive Descent* und *Shift Reduce*
- Dafür wesentlich schneller

Komplexität

- Laufzeit in $\mathcal{O}(n^3)$ im schlimmsten Fall
- Für unambige Grammatiken sogar $\mathcal{O}(n^2)$
- Für bestimmte Typen von Grammatiken (LR) sogar $\mathcal{O}(n)$
- Funktioniert am besten mit *links-rekursiven* Regeln

4. Parsing mit Merkmalstrukturen

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus
- 4 **Parsing mit Merkmalstrukturen**
 - **Modifizierter Earley Parser**
 - **Komplexität**

4.1. Modifizierter Earley Parser

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus
- 4 Parsing mit Merkmalstrukturen
 - **Modifizierter Earley Parser**
 - Komplexität

1. Möglichkeit

- Parsen wie bisher und am Ende versuchen, zu unifizieren
- Unschön: **Zahl von möglichen Analysen wird nicht so früh wie möglich beschränkt**

→ Optimierungspotential

2. Möglichkeit

- Merkmalstruktur zu jedem Earley-Zustand hinzufügen
- Complete-Operation unifiziert die Merkmalstrukturen der beiden Zustände
- Predict-Operation fügt neuen Zustand nur hinzu, wenn er von keinem vorhandenen *subsumiert* wird
- Nicht-destruktive Unifikation einsetzen! (Kopien machen!)

4.2. Komplexität

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus
- 4 Parsing mit Merkmalstrukturen
 - Modifizierter Earley Parser
 - **Komplexität**

Unterschied gegenüber ursprünglichem Earley Parser

- Zustandsmenge nach Zuständen durchsuchen, deren Merkmalstrukturen mit gegebener Merkmalstruktur unifizieren
- Häufiges Kopieren von Merkmalstrukturen (nicht-destruktive Unifikation)

Komplexität

- im Allgemeinen ist Unifikationsparsen „relativ teuer“
- NP-vollständig in manchen Versionen
- mit sehr umfangreichen Constraints sogar Turing-vollständig (!)
- Zahlreiche Varianten existieren
 - Quasi-destruktive Unifikation (Hideto Tomabechi)
 - Tractable HPSG (Gerald Penn)

Rückblick auf heutige Themen

- 1 Top-Down-Parsing: Recursive Descent
- 2 Bottom-Up-Parsing: Shift Reduce
- 3 Chart Parsing: Earley Algorithmus
- 4 Parsing mit Merkmalstrukturen
 - Modifizierter Earley Parser
 - Komplexität