

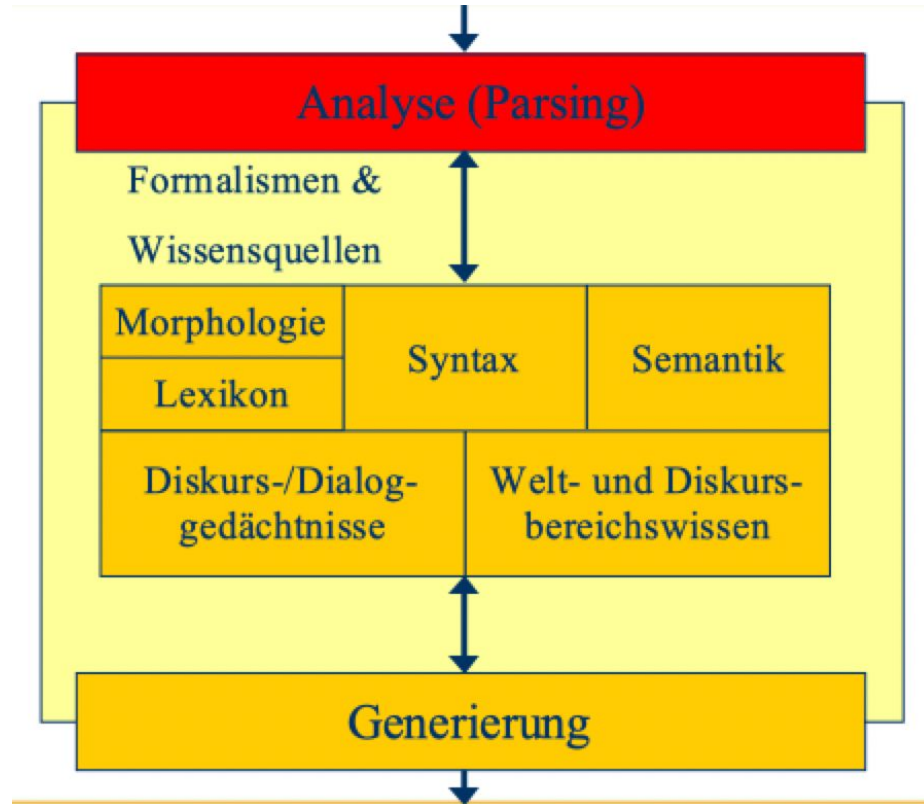
Computerlinguistik

E10: Parsingalgorithmen

Methoden

- **Begriffe, Anwendungen, Strategien, Überblick**
- **Elementare Parsingalgorithmen**
 - Top-Down-Parser
 - Bottom-Up-Parser
- **Komplexere Parser**
 - Chart-Parser
 - Statistische Parser
 - Deterministische Parser
 - Konnektionistische Parser

Architektur eines natürlichsprachlichen Systems



Parsing – Begriffe und Motivation

- **Wissensquellen** für Parser sind die Wissensquellen eines natürlichsprachlichen Systems
- **Motivation** für Parsing in der Computerlinguistik ist eine theoretisch fundierte und algorithmisch präzise Rekonstruktion des Sprachverstehens
- Parsing ist ein **algorithmisches Verfahren**, bei dem eine natürlichsprachliche Eingabekette auf eine strukturelle Beschreibung in einer formalen Repräsentationssprache abgebildet wird (vs. **Recognizer**)

Parsing – Begriffe und Motivation

- **Recognizer:** Entscheidet, ob Satz zu einer definierten Sprache gehört oder nicht

Das

Buch

liegt

auf

dem

Tisch

Parsing – Begriffe und Motivation

- **Recognizer:** Entscheidet, ob Satz zu einer definierten Sprache gehört oder nicht

Das

Buch

liegt

auf

dem

Tisch

→ True

Parsing – Begriffe und Motivation

➤ Parser: Strukturzuordnung

Das

Buch

liegt

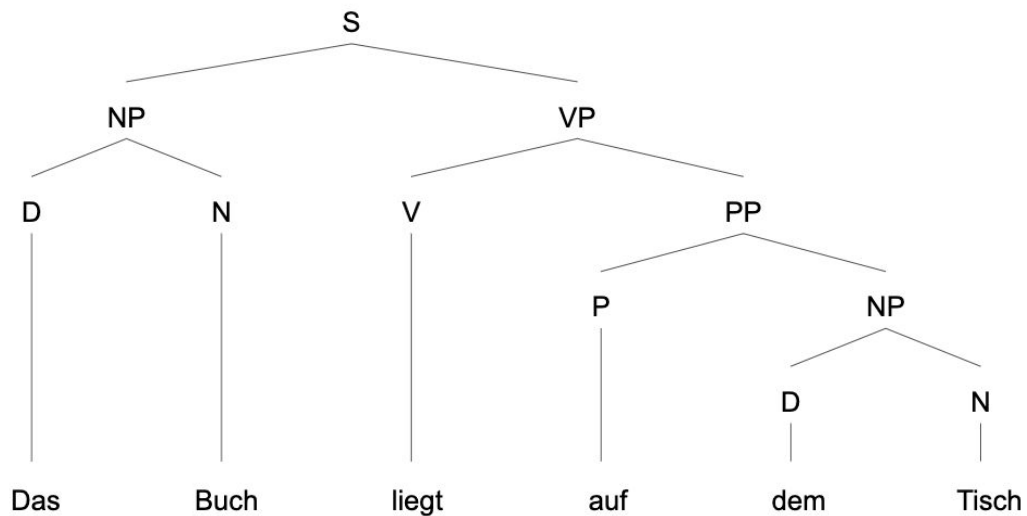
auf

dem

Tisch

Parsing – Begriffe und Motivation

- Parser: Strukturzuordnung



Überblick über die gängigsten Parser

- Elementare Parser: **Top-Down, Bottom-Up, Left-Corner**
- **Chart-Parser**
 - Earley-Algorithmus (Earley 1970)
 - Active Chart Parsing (Winograd 1983)
- Deterministische Parser
- **Statistische Parser**
- Konnektionistische Parser

Parsing-Strategien

- **Analyserichtung:** top-down vs. bottom-up
- **Suchstrategien:** breadth-first (parallel) vs. depth-first (sequenziell)
- **Verfahrensweise:** deterministisch vs. nicht-deterministisch
- **Verarbeitungsrichtung:** uni- vs. bidirektional
- **Kommunikation:** online vs. offline

Analyserichtung

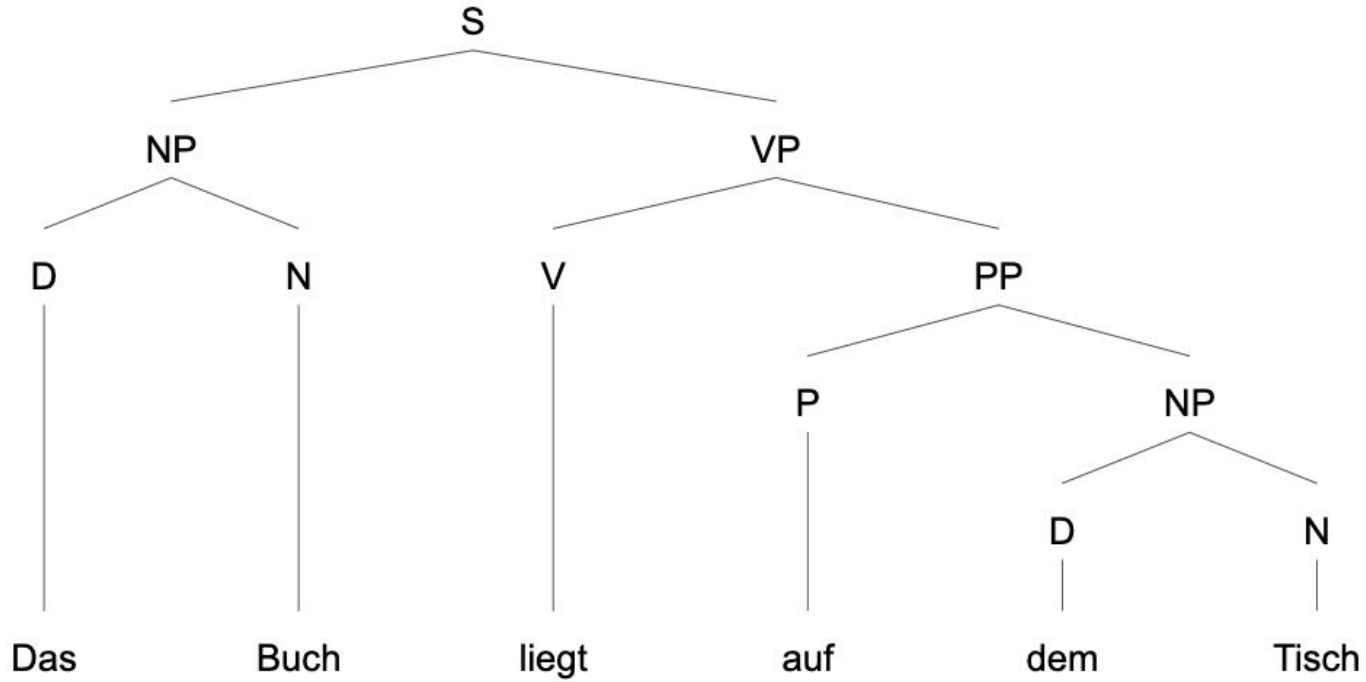
➤ Top-Down-Parsing

- zielgesteuert, von oben nach unten
- Regelexpansion: linke Regelseite → rechte Regelseite
- terminiert, wenn abgeleiteter Ausdruck dem Eingabesatz entspricht
- Probleme: Erzeugung evtl. nicht-terminierbarer Knoten; keine links-bzw. rechtsrekursiven Regeln zugelassen

➤ Bottom-Up-Parsing

- datengesteuert, von unten nach oben
- Regelreduktion: rechte Regelseite → linke Regelseite
- terminiert, wenn abgeleiteter Ausdruck dem Startsymbol entspricht
- Probleme: Bildung evtl. nicht reduzierbarer Konstituenten; keine Tilgungsregeln zugelassen

Analyserichtung



Suchstrategien

- **Breadth-First:** parallel: mehrere Alternativen der Regelanwendung werden simultan verfolgt
- **Depth-First:** sequenzielles Abarbeiten der Alternativen mit Protokoll der Auswahlentscheidungen und eventuellem Zurücksetzen (Backtracking)

Verfahrensweise

➤ Nichtdeterministisches Parsen:

- Grammatiken erlauben mehrere Alternativen
- Suchstrategien erforderlich

➤ Deterministisches Parsen:

- In jedem Zustand ist nur ein einziger Ableitungsschritt möglich
- Verwendet v.a. im Compilerbau
- Bei Einsatz für natürliche Sprache: Aufschieben der Entscheidung (look-ahead; wait & see)
- Vertreter: PARSIFAL, Tomita-Algorithmus

Top-Down Recognizer

Daten:	Lexikon und kontextfreie Syntax
Eingabe:	Satz (w) mit der Länge $n \geq 0$
Ausgabe:	TRUE / FALSE
Arbeitsstrukturen:	nächstes Wort (Anfangswert 1) Ableitung (Anfangswert Startsymbol [S])
Prozedur Main:	WENN (Ableitung leer und Wort == $n+1$) RETURN TRUE SONST Reduziere(Expandiere(Ableitung))
Prozedur Expandiere:	Wende Syntaxregel an, übergebe Ableitung
Prozedur Reduziere:	Wende Lexikonregel an, gehe zum nächsten Wort

Bottom-Up Recognizer

Daten:	Lexikon und kontextfreie Syntax
Eingabe:	STACK1 ([Satz w] mit der Länge $n \geq 1$) STACK2 ([])
Ausgabe:	TRUE / FALSE
Prozedur Main:	WENN (STACK2 nicht leer) DANN Reduce(STACK2) SONST WENN (STACK1 nicht leer) DANN Shift (STACK1, STACK2) SONST Return TRUE (wenn S auf STACK2) bzw. FALSE (im anderen Fall)
Prozedur Shift:	Schreibe Elemente von STACK1 in STACK2
Prozedur Reduce:	Ersetze Elemente in STACK2 nach Regeln der Syntax oder des Lexikons

Top-Down Parser

Daten:	Lexikon und kontextfreie Syntax
Eingabe:	Satz (w) mit der Länge $n \geq 0$
Ausgabe:	TRUE / FALSE
Arbeitsstrukturen:	nächstes Wort (Anfangswert 1) Ableitung (Anfangswert Startsymbol [S]) Position (Anfangswert 1)
Prozedur Main:	WENN (Ableitung == w) RETURN Ableitung SONST Reduziere(Expandiere(Ableitung))
Prozedur Expandiere:	Wende Syntaxregel an, ändere Variablen Ableitung und Position
Prozedur Reduziere:	Wende Lexikonregel an, ändere Variable Position

Literatur / Hausaufgabe

➤ **Zur Nachbereitung:**

- Naumann, Langer (1994): Kapitel 1 und 2 (S. 3-36)
- Geben Sie Syntax und Lexikon an, mit dem man den Satz „Das trüchtige Wildschwein schläft in dem Unterholz“ parsen kann. Führen Sie jeweils einen Top-Down- und einen Bottom-Up-Recognizing-Durchgang durch. Welche Methode kommt ohne Suchstrategie aus?

➤ **Zur Vorbereitung:**

- Naumann, Langer (1994): Kapitel 1 (S. 1-18)
- Texte und Abgabemodalitäten finden Sie im Ilias-Seminarordner.