

# Kontextsensitive Sprachen

Standardbeispiel:  $\{a^n b^n c^n\}$

$S \rightarrow a b c$   
 $S \rightarrow a A b B c c$   
 $B \rightarrow A B b c$   
 $B \rightarrow b$   
 $b A \rightarrow A b$   
 $a A \rightarrow a a$

Im Bereich der natürlichen Sprachen gibt es zahlreiche kontextsensitive Phänomene in der Semantik (z.B. Ellipsen), allerdings nur wenige in der Syntax. Beispiel: "Überkreuz-Dependenz" im Niederländischen:

*... dat Jan Piet Marie zag helpen swemmen.*  
*... daß Jan Piet Maria schwimmen helpen sah.*

Einige syntaktische Probleme sind zwar kontextsensitiv wesentlich übersichtlicher darstellbar als kontextfrei:

$S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}} \text{ PP}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}} \text{ PP } \text{ PP}$   
 $S \rightarrow \text{NP}_{\text{akk}} \text{ VP } \text{NP}_{\text{nom}}$   
 $S \rightarrow \text{NP}_{\text{akk}} \text{ VP } \text{NP}_{\text{nom}} \text{ NP}_{\text{dat}}$   
 $S \rightarrow \text{NP}_{\text{akk}} \text{ VP } \text{NP}_{\text{nom}} \text{ NP}_{\text{dat}} \text{ PP}$   
 $S \rightarrow \text{NP}_{\text{akk}} \text{ VP } \text{NP}_{\text{nom}} \text{ NP}_{\text{dat}} \text{ PP } \text{ PP}$

Kontextsensitive "Abkürzung":

$S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}} \text{ PP}$   
 $S \rightarrow \text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \text{ NP}_{\text{dat}} \text{ PP } \text{ PP}$   
 $\text{NP}_{\text{nom}} \text{ VP } \text{NP}_{\text{akk}} \rightarrow \text{NP}_{\text{akk}} \text{ VP } \text{NP}_{\text{nom}}$

Die Ableitung eines Ausdrucks durch eine kontextsensitive Grammatik entspricht jedoch nicht mehr dem Aufbau eines Strukturbaums; die Analyseverfahren für kontextfreie Grammatiken sind nicht mehr anwendbar.

Aus diesem Grund beschränkt man sich auf kontextfreie Grammatiken und ergänzt diese durch weitere Verarbeitungsschritte.

Beispiel: einfaches Grundgerüst für Sätze,

Wortstellungs- und Valenzrestriktionen werden via Unifikation bzw. durch zusätzliche Wortstellungsregeln überprüft

(Trennung von ID [immediate dominance]- und LP [linear precedence]-Regeln)

S → XP V<sub>fin</sub> XP\* V<sub>inf</sub>  
XP → NP | PP | NebS | AdvP

Valenz- und Kongruenzprüfung durch Merkmale:

V<sub>fin</sub> → *liebt*  
subj = [NP [kas=nomin, num=sg]]  
obj1 = [NP *es*]  
obj2 = [NebS [Infinitiv]]

oder Kongruenzprüfung durch zusätzliche Regel über Merkmale:

[subj [num]] = [V<sub>fin</sub> [num]]

LP-Regeln für die Wortstellung :

[NP *es*] < [NebS [Infinitiv]]

*Sie liebt es, schwimmen zu gehen.*

*Bei heißem Wetter liebt sie es, schwimmen zu gehen.*

*Es gefällt ihr, schwimmen zu gehen.*

*Jetzt heißt es schwimmen gehen.*

**Problem: Es gibt nur wenige allgemeingültige LP-Regeln.**

# Parsing: Top-down vs. Bottom-up

Gegeben:

- kontextfreie Grammatik
- zu analysierender Satz

Gesucht:

- sämtliche Strukturbeschreibungen des Satzes gemäß der Grammatik

Benötigt:

- Analyseverfahren, das diese Strukturbeschreibungen ermittelt

Alternativen:

- Ausgehend vom Startsymbol S werden beliebige Regeln ausgewählt. Passen sie nicht zum Eingabesatz, müssen diese Entscheidungen revidiert werden. (Top-down-Verfahren)
- Ausgehend vom Eingabesatz werden passende Regeln ausgewählt. Lassen sie sich nicht auf das Startsymbol zurückführen, müssen zusätzlich weitere Regeln ausprobiert werden. (Bottom-up-Verfahren)

Gemeinsamkeiten beider Verfahren:

- Üblicherweise wird ein Satz von links nach rechts analysiert.  
Beim Top-down-Verfahren werden daher zunächst die auf der rechten Seite einer Regel ganz links stehenden Symbole abgeleitet; beim Bottom-up-Verfahren beginnt man mit dem ersten Wort des Satzes.
- Bei beiden Verfahren sind üblicherweise Revisionen erforderlich (Backtracking).

Dies erfordert eine genaue Buchführung der bereits ausprobierten und der noch verfügbaren Regeln.

### Beispiel:

S → XP V XP  
S → XP V XP XP  
XP → NP  
XP → PP  
NP → Art N  
NP → N  
NP → Art N XP  
NP → N XP  
PP → Präp NP

*Das Bundesamt für Sicherheit in der  
Informationstechnik  
warnt vor einem Fehler.*

### Top-down-Analyse:

1. S → XP V XP  
2. XP → NP  
3. NP → Art N  
-- *Backtracking* --  
3. NP → Art N XP  
4. XP → NP  
-- *Backtracking* --  
4. XP → PP  
5. PP → Präp NP  
6. NP → N  
-- *Backtracking* --  
6. NP → N XP  
7. XP → NP  
-- *Backtracking* --  
7. XP → PP  
8. PP → Präp NP  
9. NP → Art N  
10. XP → NP  
-- *Backtracking* --  
10. XP → PP  
11. PP → Präp NP  
12. NP → Art N

## Strukturelle Mehrdeutigkeiten

Besitzt ein Satz mehrere syntaktische Interpretationen, so ermittelt der Parser zunächst eine einzige:

*Das Bundesamt für Sicherheit in der  
Informationstechnik  
warnt vor einem Fehler in NT4.*

- 1.-12. wie oben
- *Backtracking* --
- 12. NP → Art N XP
- 13. XP → NP
- *Backtracking* --
- 13. XP → PP
- 14. PP → Präp NP
- 15. NP → N

Analyseergebnis: *in NT4* als Attribut zu *Fehler*  
(Welche Struktur als erste ermittelt wird, hängt von der Sortierung der Regeln ab!)

Wird dieses Ergebnis von der semantischen Analyse nicht akzeptiert (oder sollen der semantischen Analyse alle möglichen syntaktischen Interpretationen zum Vergleich vorgelegt werden), wird "künstlich" ein weiterer Revisionsprozess gestartet.

In diesem Beispiel bedeutet dies ein Backtracking bis zurück zur ersten Regelanwendung und eine völlige Neuanalyse auf der Grundlage von

- 1. S → XP V XP XP

Diese vollständige Reanalyse hätte durch eine Umformulierung der Grammatik vermieden werden können (bei entsprechender "Verbiegung" der linguistischen Interpretation):

- S → XP V XPrek
- XPrek → XP
- XPrek → XP XPrek

...



## Behandlung von Rekursion:

Eine weitere Schwierigkeit bei der Top-down-Analyse stellen linksrekursive Regeln dar.

(Rechtsrekursive Regeln sind dagegen nur anwendbar, wenn ihr Anfang mit dem Eingabesatz übereinstimmt, was eine Endlosrekursion verhindert.)

Beispiel:

S → XP V XP  
S → XP V XP XP  
XP → NP  
XP → PP  
NP → Art N  
NP → N  
NP → NP XP  
PP → Präp NP

Mögliche Strategien:

- Eine rekursive Regel darf nur dann eingesetzt werden, wenn alle anderen Revisionsmöglichkeiten gescheitert sind.
- Eine rekursive Regel darf maximal dreimal rekursiv verwendet werden (Statistik!)

## Bottom-up-Analyse:

Naheliegende Strategie: Terminalsymbole durch die erste passende Regel auf Nichtterminalsymbole zurückführen; diese Nichtterminalsymbole entweder sofort oder besser nach vollständigem Durchlaufen des Satzes auf weitere Nichtterminalsymbole reduzieren

Anwendung dieser Strategie auf das o.g. Beispiel:

*Das Bundesamt für Sicherheit in der Informationstechnik warnt vor einem Fehler.*

1. NP → Art N	2. NP → N	3. NP → Art N	4. NP → Art N
5. XP → NP	6. PP → Präp NP	7. PP → Präp NP	8. PP → Präp NP
	9. XP → PP	10. XP → PP	11. XP → PP

ergibt die nicht weiter reduzierbare Struktur XP XP XP V XP

Problem: Backtracking in derselben Form wie bei der Top-down-Analyse führt nicht zu einer Lösung, weil die Anwendung der Regel

NP → Art N XP

die vorherige Erkennung der Konstituente XP voraussetzt!

Daher notwendig: gleichzeitige Verfolgung aller Analysemöglichkeiten

(von denen die meisten sich als Sackgasse erweisen, z.B. *Das Bundesamt* → Art NP)

Weitere Möglichkeit: Kombination von Top-down- und Bottom-up-Verfahren (sog. Left-corner-Parsing)

**kombiniert Vorteile, aber auch Nachteile beider Strategien**