

Einführung in die Computerlinguistik

Syntax II - Syntaktische Verarbeitung

WS 2006/2007

Manfred Pinkal

Eigenschaften der syntaktischen Struktur [1]

- Er hat die Übungen gemacht.
- Der Student hat die Übungen gemacht.
- Der interessierte Student hat die Übungen gemacht.
- Der an computerlinguistischen Fragestellungen interessierte Student hat die Übungen gemacht.
- Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester hat die Übungen gemacht.
- Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, der im Hauptfach Informatik studiert, hat die Übungen gemacht.
- Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, der im Hauptfach, für das er sich nach langer Überlegung entschieden hat, Informatik studiert, hat die Übungen gemacht.

Eine kontextfreie Grammatik für deutsche Sätze

$S \rightarrow NP VI$ $NP \rightarrow ART NN (SREL)$

$SREL \rightarrow RPRO VI$ $S \rightarrow NP VT NP$

$SREL \rightarrow RPRO NP VT$

$VI \rightarrow \textit{schläft, arbeitet, wartet, ...}$

$VT \rightarrow \textit{wählt, studiert, liest, kennt, ...}$

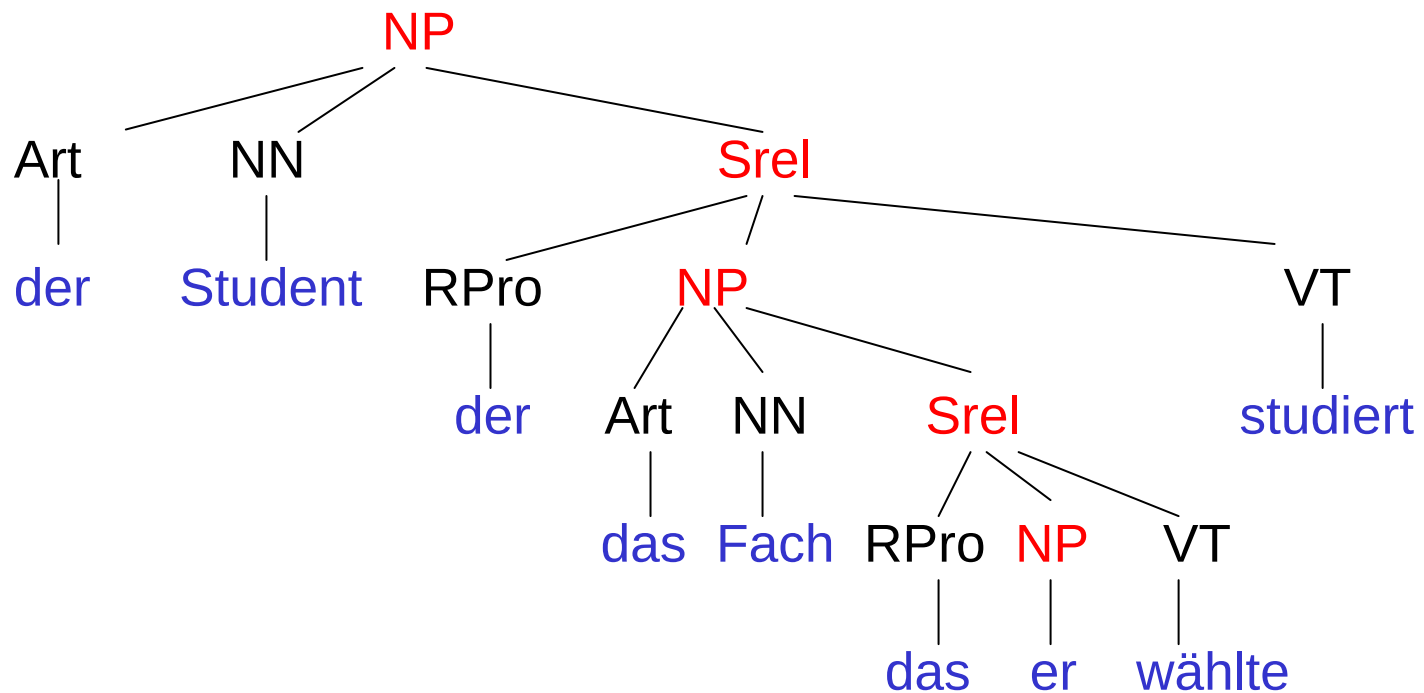
$NN \rightarrow \textit{Student, Fach, Dozentin, Professor, Buch, ...}$

$RPro \rightarrow \textit{der, die, das}$

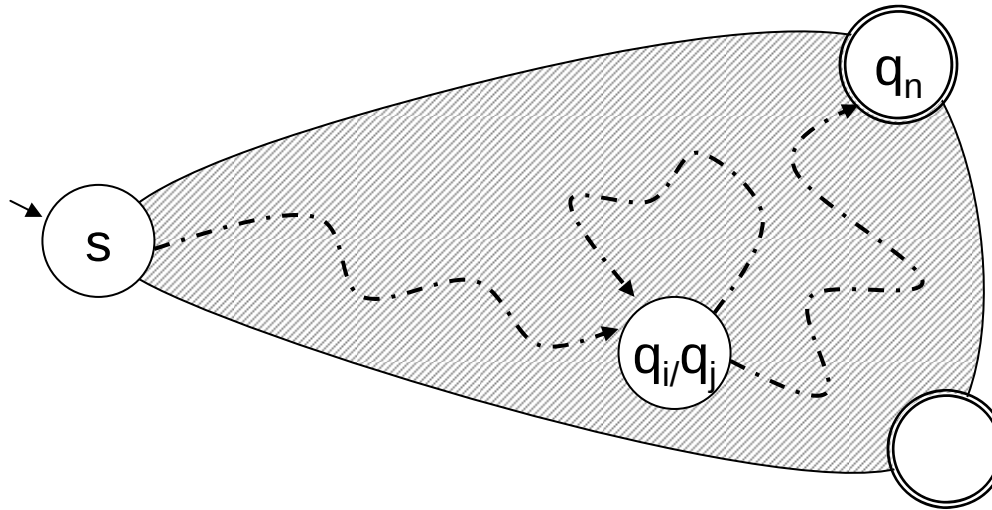
$ART \rightarrow \textit{der, die, das}$

Konstituentenstruktur

[_{NP} der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, [_{SRel} der [_{NP} das Fach, [_{SRel} das [_{NP} er] nach langer Überlegung gewählt hat]], eifrig studiert]



Das "Pumping Lemma"

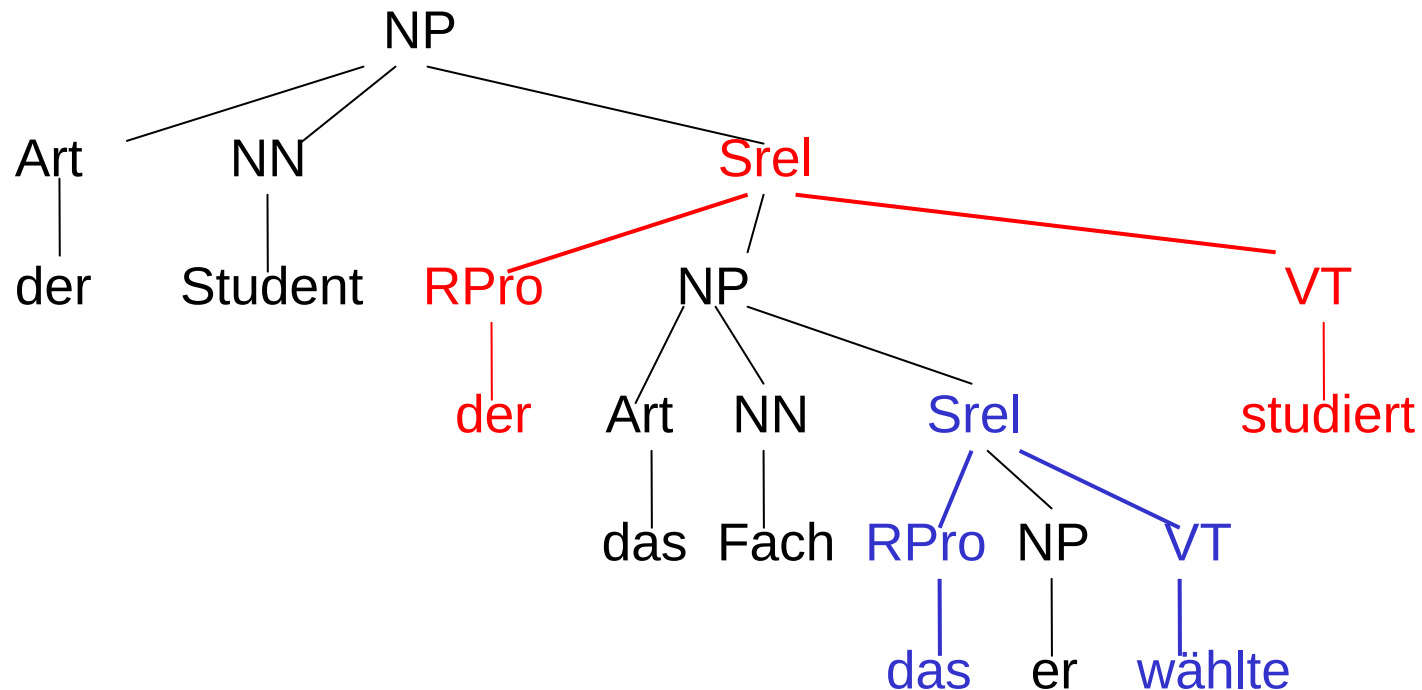


Endliches Gedächtnis als Problem

- Der endliche Automat kann Information nur in seinen (endlich vielen) Zuständen kodieren.
- Das Gedächtnis eines endlichen Automaten mit n Zuständen reicht deshalb höchstens $n-1$ Zeichen zurück.
- Kontextfreie Grammatiken erzeugen beliebig tief geschachtelte Strukturen, in denen beliebig weit voneinander entfernte Elemente voneinander abhängen können.

Konstituentenstruktur

[_{NP} *der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester*, [_{SRel} *der* [_{NP} *das Fach*, [_{SRel} *das* [_{NP} *er*] *nach langer Überlegung gewählt hat*]], eifrig *studiert*]]

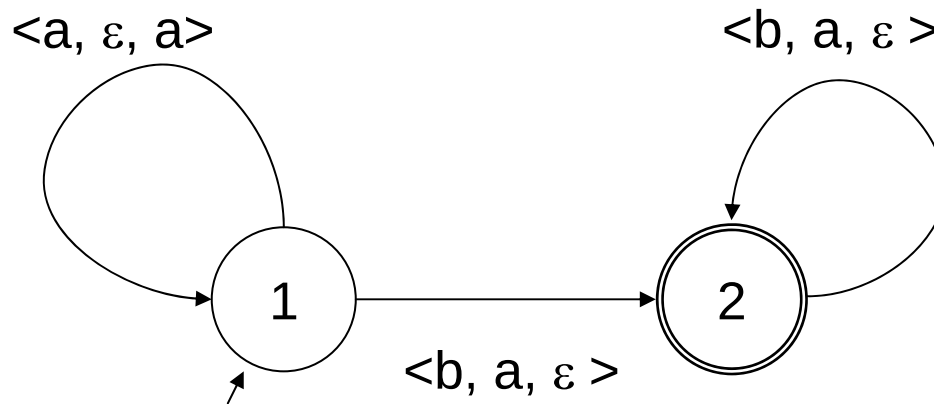


Die Lösung

- Wir erweitern Automaten durch einen zusätzlichen, im Prinzip unbegrenzten Speicher. Beim Übergang zwischen zwei Zuständen kann der Automat in den Speicher schreiben, Speicherinhalte lesen, modifizieren und löschen.
- Für die Verarbeitung kontextfreier Sprachen reicht als Speicher ein Stack/ Stapel, auf den der Automat Symbole von einer Seite ablegen und löschen kann. Wir sprechen von einem „Keller-Automaten“ (Push-down Automaton, PDA).

Kellerautomaten: Ein Beispiel

- Ein Kellerautomat, der $a^n b^n$ akzeptiert:



- $\langle u, v, w \rangle$ als Kanteninschrift steht für: **Lies Eingabe u** , **lösche v** vom Stack, **schreibe w** auf den Stack.
- Im Zustand 1 werden a 's gelesen und in den Stack geschrieben.
- Beim ersten b wechselt der Automat in den Zustand 2 und löscht für jedes gelesene b ein a vom Stack.
- Wenn die Eingabe abgearbeitet, der Stack leer und ein Endzustand erreicht ist, wird das Eingabewort akzeptiert.

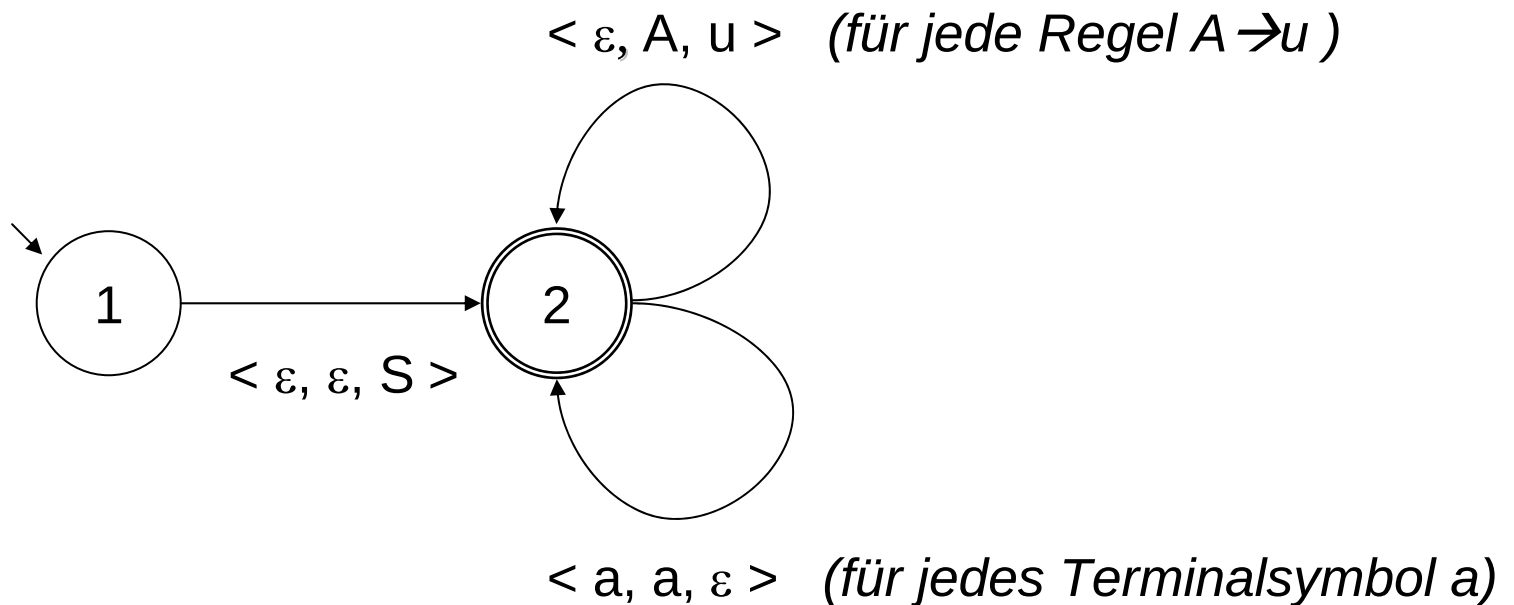
Keller-Automaten und kontextfreie Grammatiken

- Kontextfreie Grammatiken (CFGs) erlauben die einfache und elegante Darstellung syntaktischer Information.
- Keller-Automaten und kontextfreie Grammatiken sind gleichstarke Formalismen: Jede kontextfreie Sprache wird von einem Kellerautomaten akzeptiert, und umgekehrt.
- Frage: Wie kann die syntaktische Verarbeitung direkt auf CFGs zugreifen?
- Verfahren oder Systeme, die die Regeln einer Grammatik in den syntaktischen Analyseprozess umsetzen, nennen wir **Parser**.

Ein Parser für kontextfreie Grammatiken

- Initialisiere den Stack des Automaten mit dem Startsymbol.
- Wenn sich das oberste Stack-Element ein nicht-terminals Symbol A ist, wähle eine Grammatikregel $A \rightarrow u$ und ersetze das Stack-Symbol A durch u .
- Wenn das oberste Stack-Element ein Terminalsymbol a und mit dem aktuelle Eingabesymbol identisch ist, lösche a vom Stack und rücke in der Eingabe vor.

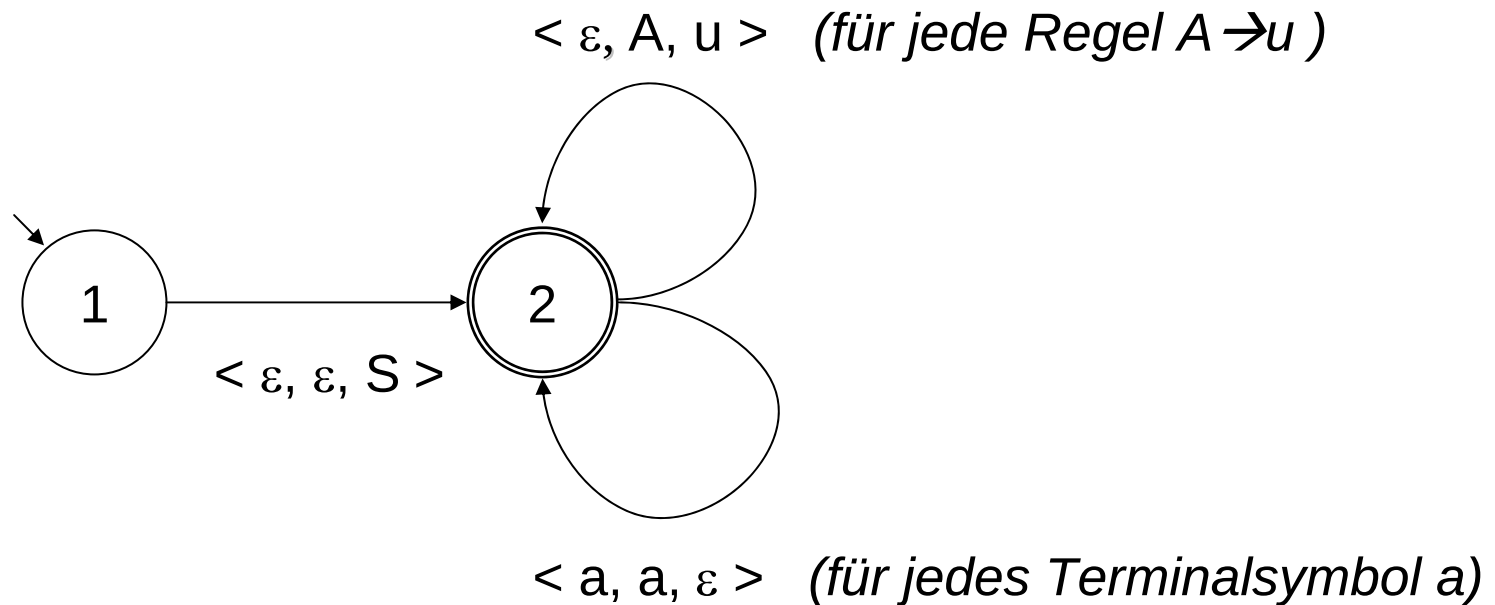
Ein Parser für kontextfreie Grammatiken



Beispiel: $a^n b^n$

$S \rightarrow aSb$

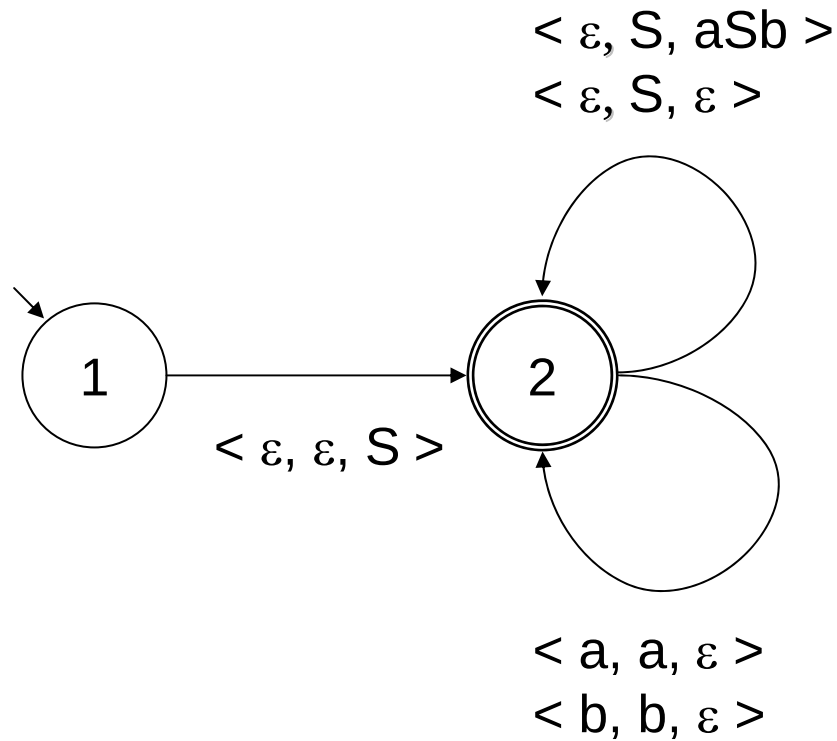
$S \rightarrow \varepsilon$



Beispiel: $a^n b^n$

$S \rightarrow aSb$

$S \rightarrow \varepsilon$



Beispiel: Deutsches Grammatikfragment

$S \rightarrow NP \text{ VI}$ $NP \rightarrow \text{ART} \text{ NN (SREL)}$

$\text{SREL} \rightarrow \text{RPRO} \text{ VI}$ $S \rightarrow NP \text{ VT} \text{ NP}$

$\text{SREL} \rightarrow \text{RPRO} \text{ NP VT}$

$\text{VI} \rightarrow \textit{schläft, arbeitet, wartet, ...}$

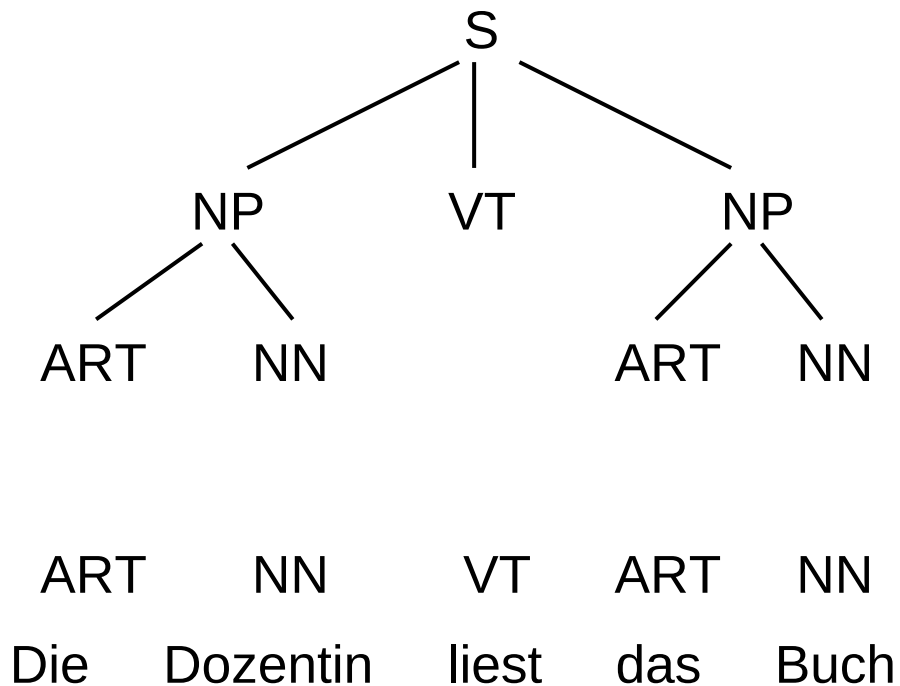
$\text{VT} \rightarrow \textit{wählt, studiert, liest, kennt, ...}$

$\text{NN} \rightarrow \textit{Student, Fach, Dozentin, Professor, Buch, ...}$

$\text{RPro} \rightarrow \textit{der, die, das}$

$\text{ART} \rightarrow \textit{der, die, das}$

Eine andere Sicht auf das Verfahren



Elementare Parsing-Strategien

- Das beschriebene Verfahren heißt **Top-Down-Parsing**: Der Algorithmus geht vom Start-/Satzsymbol aus, wendet Grammatikregeln versuchsweise von oben nach unten an und gleicht die erzeugten Terminalsymbole (Wörter, Lexikalische Kategorien) von links nach rechts mit der Eingabekette ab.
- **Bottom-Up-Parsing**: Die Eingabekette wird sukzessive eingelesen. Wenn die rechte Seite einer Grammatikregel passt, wird diese in umgekehrter Richtung angewandt, und auf diese Weise eine Konstituentenstruktur von unten nach oben aufgebaut.
- Top-Down-Parser und Bottom-Up-Parser sind **hochgradig nicht-deterministisch**. Der Top-Down-Parser, wie er vorgestellt wurde, erlaubt bei einem nicht-terminalen Symbol auf dem Stack, für das es n Regeln in der Grammatik gibt, z.B. n alternative Ersetzungsschritte.

Parsing und Determinismus

- Gibt es – wie bei den endlichen Automaten – zu jedem nicht-deterministischen Kellerautomaten/ kontextfreien Parser ein deterministisches Äquivalent?
- Leider nein:
- Formale Sprachen (Logik, Arithmetik, Programmiersprachen) sind typischerweise so definiert, dass sie deterministische Analyse erlauben (auch $a^n b^n$ ist deterministisch).
- Für natürliche Sprachen gibt es keine deterministische syntaktische Analyse.

Nicht-Determinismus in der Syntax

- Unterschiedliche Lesarten entsprechen unterschiedlichen Analysemöglichkeiten. Syntaktisch-strukturelle Mehrdeutigkeit:
 - *Peter sah den Mann mit dem Teleskop.*

Grammatische Mehrdeutigkeit, Beispiel

- Die folgende Grammatik erlaubt Präpositionalphrasen (PPs) als VP- (Verbphrasen-) und als NP-Modifikatoren

$S \rightarrow NP VP$

$VP \rightarrow VT NP$

$VP \rightarrow VI$

$NP \rightarrow ART NN$

$NP \rightarrow PN$

$NP \rightarrow NP PP$

$PP \rightarrow P NP$

$VP \rightarrow VP PP$

- Der Satz: *Peter sah den Mann mit dem Teleskop* hat in dieser Grammatik zwei Analysen, die zwei Bedeutungsvarianten entsprechen:

- $[_S \text{Peter} [_{VP} \text{sah} [_{NP} [_{NP} \text{den Mann}] [_{PP} \text{mit dem Teleskop}]]]]$
- $[_S \text{Peter} [_{VP} [_{VP} \text{sah} [_{NP} \text{den Mann}]]] [_{PP} \text{mit dem Teleskop}]]]$

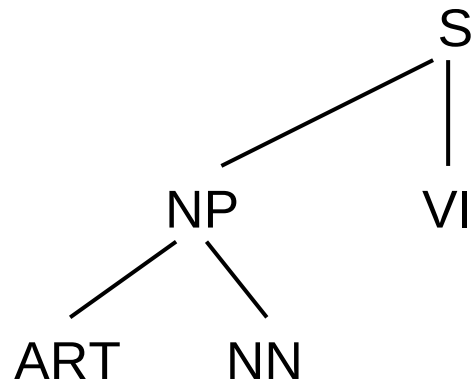
- Der Satz *Peter rief den Kollegen aus München mit dem Handy an* hat mindestens 4 Lesarten (welche?).

Nicht-Determinismus in der Syntax

- Unterschiedliche Lesarten entsprechen unterschiedlichen Analysemöglichkeiten. **Syntaktisch-strukturelle Mehrdeutigkeit:**
 - *Peter sah den Mann mit dem Teleskop.*
- Menschen verfolgen bei der syntaktischen Verarbeitung oft eine Primäranalyse, die revidiert werden muss („**Holzwegsätze**“):

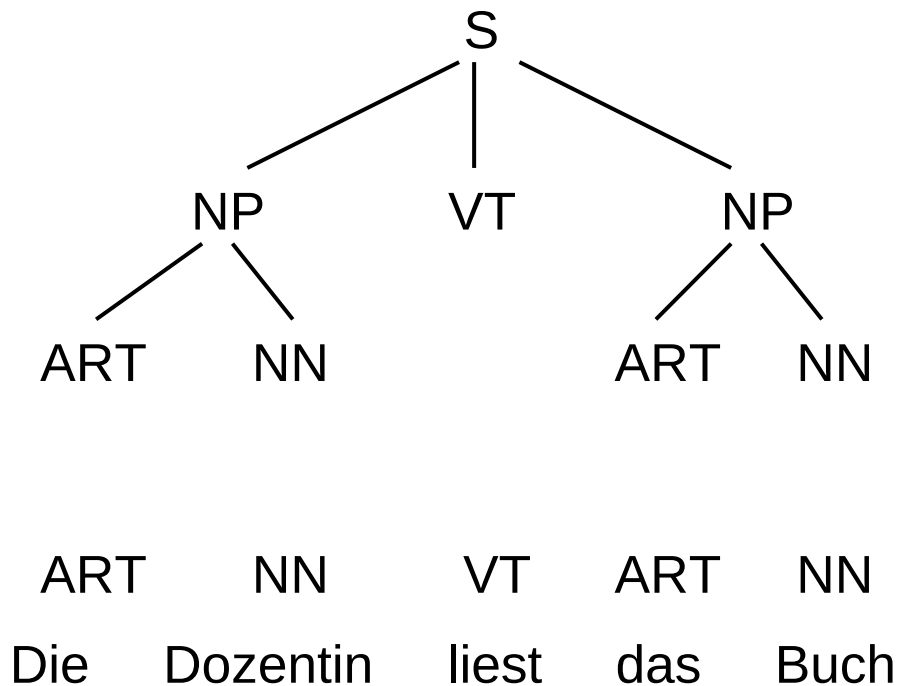
Er bezichtigte den Vater des Schreibens unkundiger Kinder

Ein Effizienzproblem



ART	NN	VT	ART	NN
Die	Dozentin	liest	das	Buch

Ein Effizienzproblem



Ein Effizienzproblem und seine Lösung

- Der Parser hat beim Abarbeiten des Verbs „festgestellt“, dass nicht die Regel $S \rightarrow NP VI$, sondern $S \rightarrow NP VT NP$ verwendet werden muss. Bis er zum kritischen Punkt gelangt, muss er aber die komplette NP-Analyse - mit allen Irrwegen - noch einmal identisch durchlaufen.
- Lösungsidee: Teilanalysen (erkannte mögliche Konstituenten) werden zwischengespeichert. Der Algorithmus greift nicht direkt auf Eingabesymbole (Wörter, lexikalische Kategorien) sondern, wo vorhanden, auf erkannte Konstituenten zurück.
- Die Repräsentation, die für die Speicherung verwendet wird, nennt man **Chart**, das Parsen mithilfe einer Chart „**Chart-Parsing**“.

Die Chart, Beispiel

① Der ① Detektiv ② im ③ Sportwagen ④ verfolgte ⑤ den ⑥ Gangster ⑦

Die Chart

NP

NP

PP

① Der ② *Detektiv* ③ *im* ④ *Sportwagen* ⑤ *verfolgte* ⑥ *den* ⑦ *Gangster*



Die Chart

NP → NP PP

NP → ART NN

PP → PAK NN

① Der ① Detektiv ② im ③ Sportwagen ④ verfolgte ⑤ den ⑥ Gangster ⑦



Die Chart

$S \rightarrow NP\ VT\ NP$

$NP \rightarrow NP\ PP$

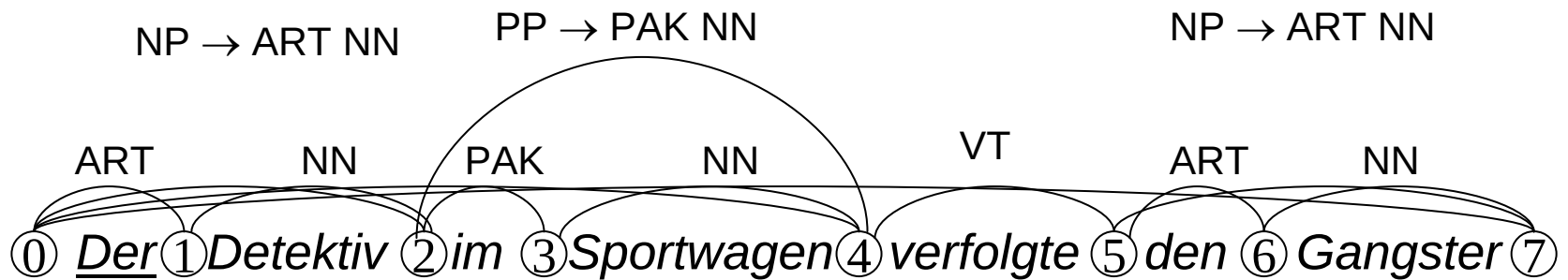


Chart-Parsing

- Chart-Parsing in der einen oder anderen Form ist ein Standardverfahren für die automatische Analyse von Konstituentenstrukturgrammatiken.
- Chart-Parser erlauben die syntaktische Analyse kontextfreier Grammatiken in polynomischer Zeit: Zeitbedarf in Abhängigkeit von der Länge der Eingabe: $O(n^2)$ für das Auffinden einer Analyse, $O(n^2)$ für das Auffinden aller Analysen.

Interaktion von grammatischen Merkmalen

- *Der [m,sg, nom]an computerlinguistischen Fragestellungen interessierte [m,sg, nom] Student [m,sg, nom] im ersten Semester, der [m,sg, nom] im Hauptfach, für das er [m,sg, nom] sich nach langer Überlegung entschieden hat [sg], Informatik studiert [sg], hat die Übungen gemacht.*

Interaktion von grammatischen Merkmalen

- **Kongruenz**: bestimmte grammatische Merkmale verschiedener Ausdrücke müssen übereinstimmen
- **Rektion**: ein Ausdruck (lexikalischer Kopf) verlangt das Vorliegen bestimmter Merkmale bei abhängigen Ausdrücken
- Beispiele für Kongruenz:
 - in Nominalausdrücken (Numerus, Genus, Kasus)
der [sg,nom,m]interessierte [sg,nom,m] Student [sg,nom,m]
dem[sg,dat,m] interessierten[sg,dat,m] Studenten[sg,dat,m]
 - Subjekt-Verb-Kongruenz (Person, Numerus)
Der Student[sg] arbeitet[sg] / Die Studenten[sg] arbeiten[sg]
- Beispiele für Rektion (Kasus)
 - *die Übungen [akk] abgeben / der Dozentin [dat] zuhören*
 - *mit den Kollegen [dat] / ohne die Kollegen [akk]*
 - *dem Fach [dat] zugetan / des Faches [gen] überdrüssig*

Interaktion von grammatischen Merkmalen

Nominalkongruenz

- *Der [m,sg, nom]an computerlinguistischen Fragestellungen interessierte [m,sg, nom] Student [m,sg, nom] im ersten Semester, der [m,sg, nom] im Hauptfach, für das er [m,sg, nom] sich nach langer Überlegung entschieden hat [sg], Informatik studiert [sg], hat die Übungen gemacht.*

Interaktion von grammatischen Merkmalen

Pronominalkongruenz

- *Der [m,sg, nom]an computerlinguistischen Fragestellungen interessierte [m,sg, nom] Student [m,sg, nom] im ersten Semester, der [m,sg, nom] im Hauptfach, für das er [m,sg, nom] sich nach langer Überlegung entschieden hat [sg], Informatik studiert [sg], hat die Übungen gemacht.*

Interaktion von grammatischen Merkmalen

Subjekt-Verb-Kongruenz

- *Der [m,sg, nom]an computerlinguistischen Fragestellungen interessierte [m,sg, nom] Student [m,3, sg, nom] im ersten Semester, der [m,3, sg, nom] im Hauptfach, für das er [m,3,sg, nom] sich nach langer Überlegung entschieden hat [3,sg], Informatik studiert [3, sg], hat [3, sg]die Übungen gemacht.*

Ein komplexerer Fall: Relativpronomen

Das Relativpronomen bekommt seine grammatischen Merkmale durch zwei unterschiedliche Mechanismen zugewiesen:

- Numerus , Genus-Kongruenz mit dem Kopf der Nominalphrase
- Kasusreaktion durch das Hauptverb des Relativsatzes

der Student [m,sg], den[m,sg,akk] die Computerlinguistik interessiert
der Student [m,sg], dem [m,sg,dat] die Computerlinguistik gefällt
die Studentin [f,sg], die [f,sg,akk] die Computerlinguistik interessiert
die Studentin [f,sg], der [f,sg,dat] die Computerlinguistik gefällt

Merkmalsverwaltung durch Unifikation

In der nächsten Woche!