

Eigenschaften der syntaktischen Struktur [1]

- Sätze setzen sich aus Satzteilen (Konstituenten) zusammen, die wiederum aus einfachen oder ihrerseits komplexen Satzteilen bestehen können. Sätze können deshalb beliebig lang und beliebig tief geschachtelt sein.

Konstituenten

- 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.

Syntaktische Kategorien [1]

Konstituenten-Typen werden „syntaktische Kategorien“ genannt;
Beispiele:

- Nominalphrasen (Nominalausdrücke): *er – der Student – der interessierte Student – die Übungen – computerlinguistischen Fragestellungen*
- Präpositionalphrasen (Präpositionalausdrücke): *an computerlinguistischen Fragestellungen – im ersten Semester, – nach langer Überlegung*
- Adjektivphrasen: *interessierte – an computerlinguistischen Fragestellungen interessierte*
- Verben, Verbkomplex: *hat – gemacht – entschieden hat*
- Satz: Haupt- und Nebensätze unterschiedlicher Art

Kategorie und Funktion

- Die syntaktische Kategorie ergibt sich aus dem internen Aufbau einer Konstituente, insbesondere aus der Wortart ihres „lexikalischen Kopfes“: Die Konstituenten
 - *der an computerlinguistischen Fragestellungen interessierte Student*
 - *an computerlinguistischen Fragestellungen interessiert*
 - *an computerlinguistischen Fragestellungen*sind Nominal-, Adjektiv- und Präpositionalphrase, weil der jeweilige Kopf Substantiv („Nomen“), Adjektiv, bzw. Präposition ist.
- Die grammatische Funktion dagegen bezeichnet die Rolle, die eine Konstituente im ganzen Satz spielt. Ein Nominalausdruck z.B. kann, je nach Stellung im Satz unter anderem die Funktion von Subjekt, (direktem oder indirektem) Objekt, (Genitiv-) Attribut oder Prädikatsnomen besitzen.

Eigenschaften der syntaktischen Struktur

- Sätze setzen sich aus Satzteilen (Konstituenten) zusammen, die wiederum aus einfachen oder ihrerseits komplexen Satzteilen bestehen können. Sätze können deshalb beliebig lang und beliebig tief geschachtelt sein.
- Die Syntax natürlicher Sprachen erlaubt variable Wortstellung: Wörter und Konstituenten mit der gleichen Funktion können an unterschiedlichen Stellen eines Satzes stehen. Unterschiedliche Sprachen erlauben sehr unterschiedliche Freiheitsgrade.

Variable Wortstellung

Peter hat der Dozentin das Übungsblatt heute ins Büro gebracht.

Das Übungsblatt hat Peter der Dozentin heute ins Büro gebracht.

Der Dozentin hat Peter heute das Übungsblatt ins Büro gebracht.

Ins Büro hat heute Peter der Dozentin das Übungsblatt gebracht.

Heute hat Peter das Übungsblatt der Dozentin ins Büro gebracht.

Ins Büro hat das Übungsblatt der Dozentin Peter heute gebracht.

** Ins Büro heute Peter das Übungsblatt hat gebracht der Dozentin.*

** Ins heute Büro der Peter Dozentin das hat Übungsblatt gebracht.*

Eigenschaften der syntaktischen Struktur [3]

- Sätze setzen sich aus Satzteilen (Konstituenten) zusammen, die wiederum aus einfachen oder ihrerseits komplexen Satzteilen bestehen können. Sätze können deshalb beliebig lang und beliebig tief geschachtelt sein.
- Die Syntax natürlicher Sprachen erlaubt variable Wortstellung: Wörter und Konstituenten mit der gleichen Funktion können an unterschiedlichen Stellen eines Satzes stehen. Unterschiedliche Sprachen erlauben sehr unterschiedliche Freiheitsgrade.
- Die grammatischen Eigenschaften unterschiedlicher Wörter und Konstituenten im Satz hängen voneinander ab – zum Teil auch in Fällen, in denen die Wörter und Konstituenten im Satz weit auseinander liegen.

Eigenschaften der syntaktischen Struktur [3]

- Sätze setzen sich aus Satzteilen (Konstituenten) zusammen, die wiederum aus einfachen oder ihrerseits komplexen Satzteilen bestehen können. Sätze können deshalb beliebig lang und beliebig tief geschachtelt sein.
- Die Syntax natürlicher Sprachen erlaubt variable Wortstellung: Wörter und Konstituenten mit der gleichen Funktion können an unterschiedlichen Stellen eines Satzes stehen. Unterschiedliche Sprachen erlauben sehr unterschiedliche Freiheitsgrade.
- Die grammatischen Eigenschaften unterschiedlicher Wörter und Konstituenten im Satz hängen voneinander ab – zum Teil auch in Fällen, in denen die Wörter und Konstituenten im Satz weit auseinander liegen.

Eigenschaften der syntaktischen Struktur [3]

- *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.*

Kontextfreie Grammatiken: Ein Beispiel

$S \rightarrow NP VI$

$S \rightarrow NP VT NP$

$NP \rightarrow ART NN$

$VI \rightarrow \textit{schläft}$

$VI \rightarrow \textit{arbeitet}$

$VT \rightarrow \textit{studiert}$

$VT \rightarrow \textit{wählt}$

$ART \rightarrow \textit{der}$

$NP \rightarrow ART NN SREL$

$SREL \rightarrow RPRO VI$

$SREL \rightarrow RPRO NP VT$

$NN \rightarrow \textit{Student}$

$NN \rightarrow \textit{Fach}$

$RPro \rightarrow \textit{der}$

$RPro \rightarrow \textit{das}$

$ART \rightarrow \textit{das}$

- Kontextfreie Grammatiken heißen in der Linguistik auch Konstituentenstruktur-Grammatiken oder Phrasenstruktur-Grammatiken

Eine kontextfreie Grammatik für deutsche Sätze

Kompaktere Schreibweise:

- Optionale Konstituenten werden in Klammern geschrieben.
- Lexikalische Symbole werden mit Komma aneinandergehängt.

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

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

$SREL \rightarrow RPRO NP VT$

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

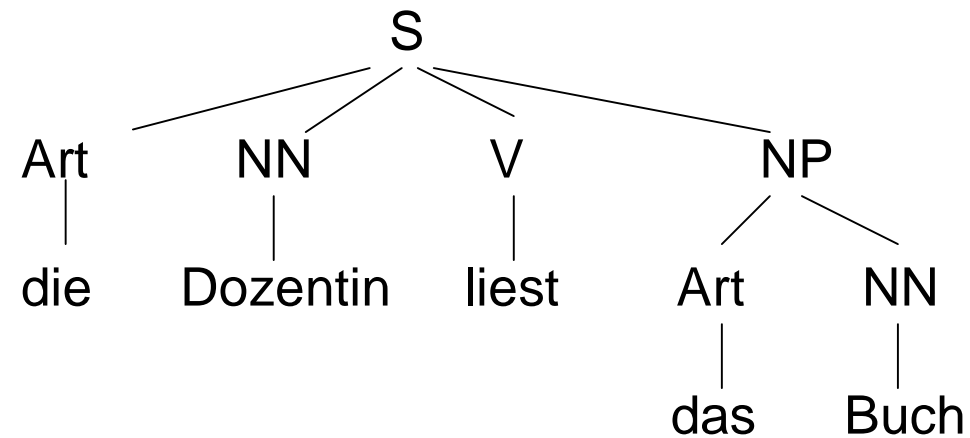
$VT \rightarrow \textit{wählt, studiert, liest, kennt, ...}$

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

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

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

Kontextfreie Grammatik: Ein Beispielsatz



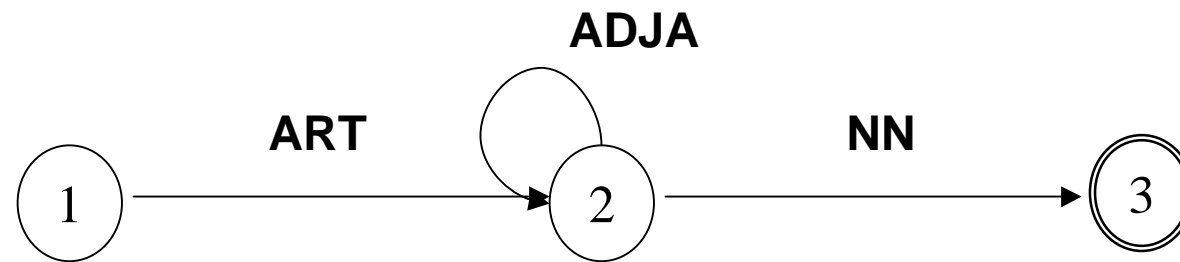
Lässt sich syntaktische Struktur mit endlichen Automaten beschreiben?

- Überlegung: DEA und NEA (und reguläre Ausdrücke) sind äquivalente Formalismen, die komplizierte Sprachstrukturen beschreiben können und, wegen der garantierten Überführbarkeit in DEAs, in linearer Zeit analysieren. Es wäre wünschenswert, wenn wir das Instrumentarium auch auf syntaktische Strukturen anwenden könnten. Frage: Können wir nicht nur die Morphologie, sondern auch die Syntax natürlicher Sprachen durch endliche Automaten analysieren?

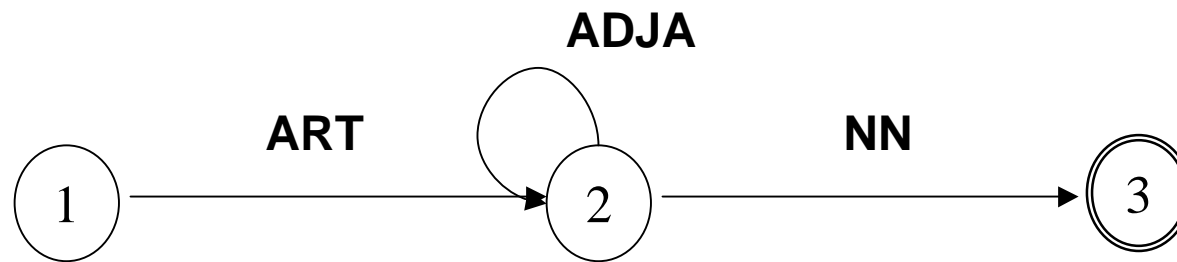
Reguläre Ausdrücke

- Reguläre Ausdrücke sind, neben NEA und DEA, ein dritter Formalismus, um Eigenschaften von Zeichenketten bzw. Sprachen zu beschreiben.
- Definition: Die Menge der regulären Ausdrücke zu einem Alphabet Σ ist die kleinste Menge, für die gilt:
 - \emptyset ist regulärer Ausdruck
 - Jedes $a \in \Sigma$ ist regulärer Ausdruck
 - Wenn α, β reguläre Ausdrücke sind, so auch
 - $\alpha + \beta$ (Alternative)
 - $\alpha \circ \beta$ (Konkatenation)
 - α^* (Iteration)

Endliche Automaten und reguläre Ausdrücke [1]



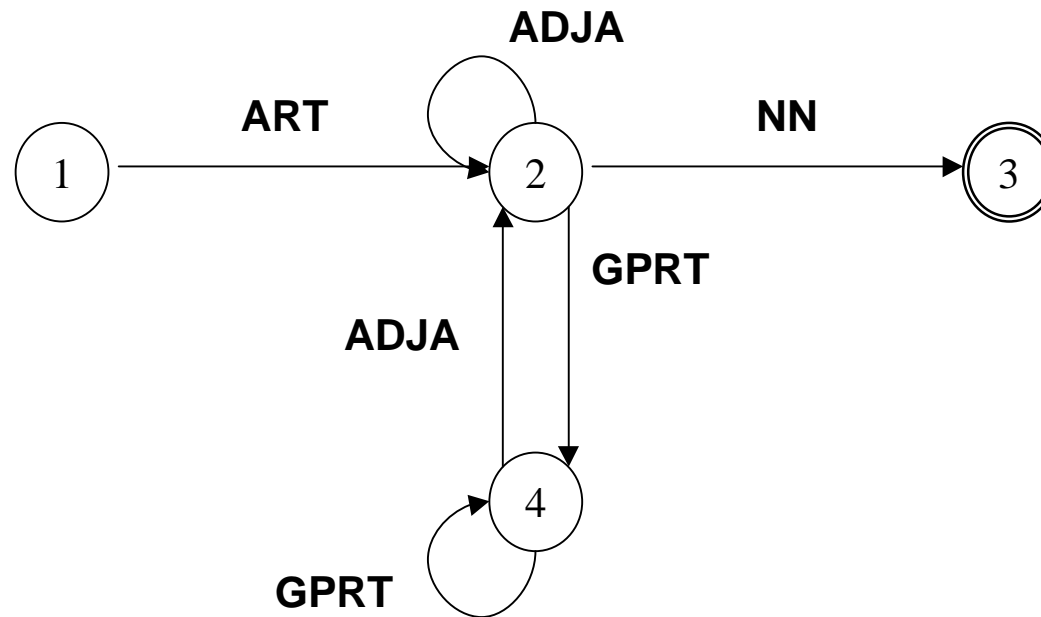
Endliche Automaten und reguläre Ausdrücke [1]



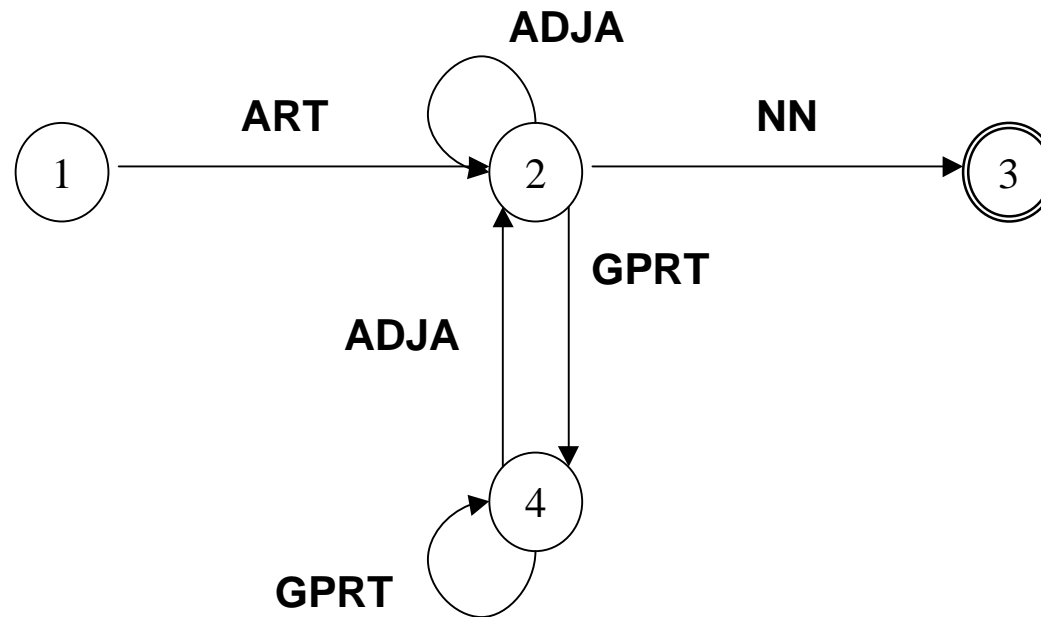
Regulärer Ausdruck:

$\text{ART} \circ \text{ADJA}^* \circ \text{NN}$

Endliche Automaten und reguläre Ausdrücke [2]



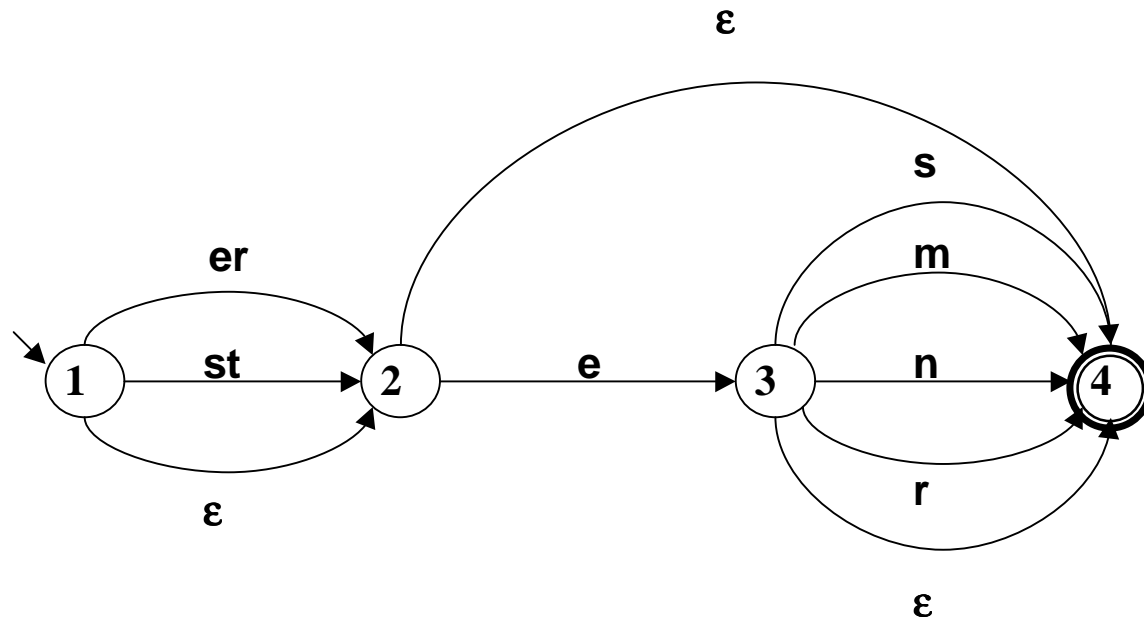
Endliche Automaten und reguläre Ausdrücke [2]



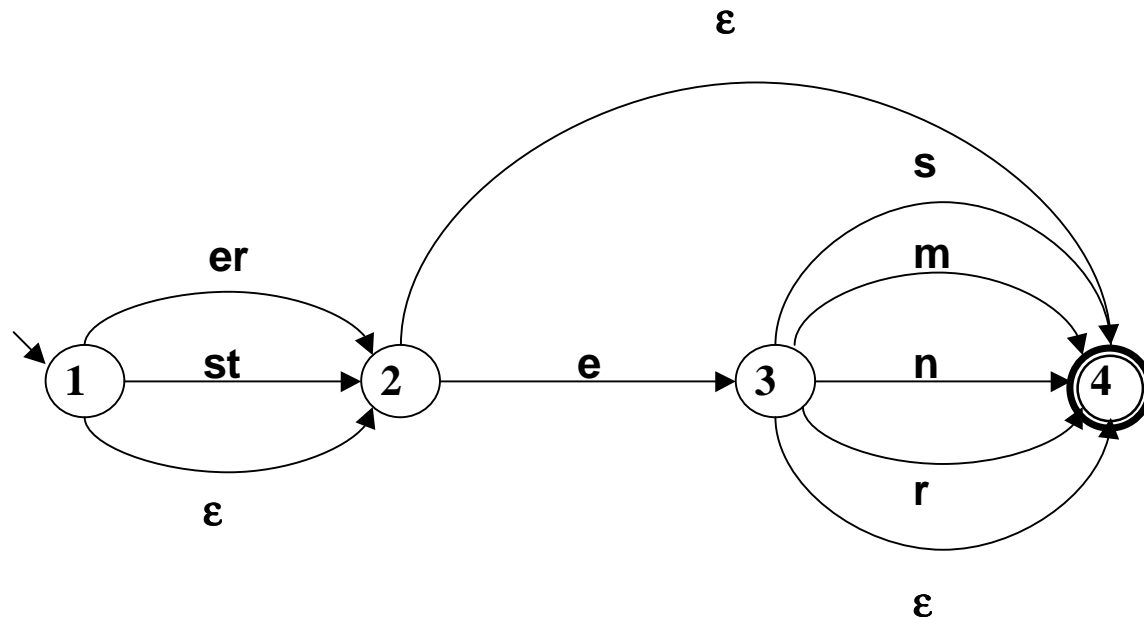
Regulärer Ausdruck:

$\text{ART} \circ (\text{ADJA}^* + (\text{GPRT} \circ \text{GPRT}^* \circ \text{ADJA})) \circ \text{NN}$

Endliche Automaten und reguläre Ausdrücke [3]



Endliche Automaten und reguläre Ausdrücke [3]



Regulärer Ausdruck:

$$(er+st+\emptyset^*) \circ (\emptyset^*+(e \circ (s+m+n+r+ \emptyset^*)))$$

Reguläre Ausdrücke: Formale Interpretation

- Reguläre Ausdrücke beschreiben Sprachen/ Mengen von Zeichenketten über einem gegebenen Alphabet.
- Die vom regulären Ausdruck Φ über Σ beschriebene Sprache nennen wir $L(\Phi)$. $L(\Phi)$ wird für beliebige reguläre Ausdrücke in der folgenden Weise rekursiv definiert:
 - $L(\emptyset) = \emptyset$
 - $L(a) = \{a\}$ für $a \in \Sigma$
 - $L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
 - $L(\alpha \circ \beta) = \{ww' \mid w \in L(\alpha) \text{ und } w' \in L(\beta)\}$
 - $L(\alpha^*) = \{w_1 \dots w_n \mid w_1, \dots, w_n \in L(\alpha), n \geq 0\}$
- Anmerkung: Es gilt $L(\emptyset^*) = \{\epsilon\}$

Reguläre Ausdrücke, endliche Automaten, reguläre Sprachen

- NEA, DEA und reguläre Ausdrücke sind Formalismen mit äquivalenter Beschreibungsstärke. Die Sprachen, die sich durch sie beschreiben lassen, heißen reguläre Sprachen.
- Vereinigung, Komplement, Durchschnitt, Konkatenation von Regulären Sprachen sind auch reguläre Sprachen.

Lässt sich syntaktische Struktur mit endlichen Automaten beschreiben?

- Können wir die Syntax natürlicher Sprachen durch endliche Automaten analysieren?
- Anders gefragt: Sind natürliche Sprachen reguläre Sprachen?
- Weitergehende Frage: Lassen sich alle Sprachen (über einem endlichen Alphabet) durch endliche Automaten beschreiben? Gibt es überhaupt Sprachen, die nicht regulär sind?

Es gibt nicht-reguläre Sprachen: Ein Existenzbeweis

Beweis:

- Wenn Σ ein endliches Alphabet ist, ist Σ^* (die Menge aller Worte über Σ) abzählbar unendlich.
- Die Menge aller Sprachen über dem Alphabet ist $\wp(\Sigma^*)$. Die Potenzmenge einer abzählbar unendlichen Menge ist überabzählbar. Also gibt es zu einem endlichen Alphabet Σ eine überabzählbar unendliche Menge von Sprachen.
- Jede reguläre Sprache über Sprache über Σ wird durch einen regulären Ausdruck über Σ definiert. Reguläre Ausdrücke über Σ sind Worte über dem endlichen Alphabet $\Sigma \cup \{\emptyset, +, \circ, *, (,)\}$. Es gibt nur abzählbar unendlich viele reguläre Ausdrücke über Σ , also nur abzählbar unendlich viele reguläre Sprachen.
- Also gibt es Sprachen über Σ , die nicht regulär sind.

Es gibt nicht-reguläre Sprachen: Ein Existenzbeweis

Beweis:

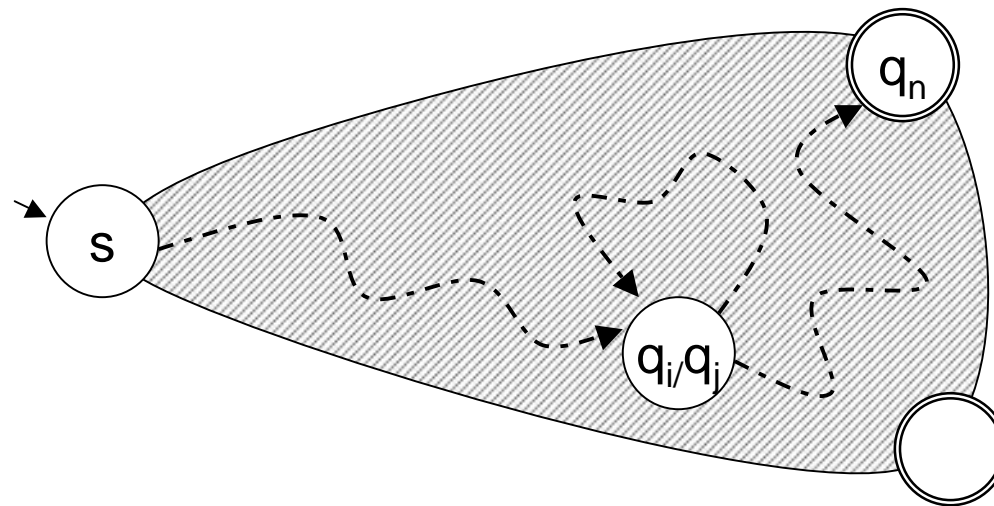
- Wenn Σ ein endliches Alphabet ist, ist Σ^* (die Menge aller Worte über Σ) abzählbar unendlich.
- Die Menge aller Sprachen über dem Alphabet ist $\wp(\Sigma^*)$. Die Potenzmenge einer abzählbar unendlichen Menge ist überabzählbar. Also gibt es zu einem endlichen Alphabet Σ eine überabzählbar unendliche Menge von Sprachen.
- Jede reguläre Sprache über Sprache über Σ wird durch einen regulären Ausdruck über Σ definiert. Reguläre Ausdrücke über Σ sind Worte über dem endlichen Alphabet $\Sigma \cup \{\emptyset, +, \cdot, *, (,)\}$. Es gibt nur abzählbar unendlich viele reguläre Ausdrücke über Σ , also nur abzählbar unendlich viele reguläre Sprachen.
- Also gibt es Sprachen über Σ , die nicht regulär sind.
- Aber was sind das für Sprachen?

Das "Pumping Lemma,, [1]

- Das Pumping Lemma (dt. „Pump-Lemma“ oder auch „uvw-Theorem“) gibt uns, anders als das vorher gezeigte Theorem, die Möglichkeit, konstruktiv zu beweisen, dass eine Sprache nicht regulär ist.
- Der Grundgedanke ist einfach: Jede reguläre Sprache L wird von einem endlichen Automaten akzeptiert. Ein endlicher Automat hat eine bestimmte, endliche Anzahl von Zuständen. Wenn ein Wort in L mehr Symbole hat als der Automat Zustände (genau genommen reichen schon mindestens so viele Symbole aus), dann muss beim Abarbeiten des Wortes ein Zustand mindestens zweimal vorkommen. Das bedeutet aber, dass beim Abarbeiten eine Schleife durchlaufen wird. Die könnte im Prinzip aber auch mehrfach durchlaufen (oder weggelassen) werden. Das heißt, für Wörter oberhalb einer bestimmten Länge durch beliebige Wiederholung eines Teilworts „aufgepumpt“ werden können und immer noch zur Sprache gehören.

Das "Pumping Lemma,, [2]

Sei $w = a_1 \dots a_n$ ein Wort, das vom DEA K mit k Zuständen erkannt wird, und $n = |w| \geq k$. Dann geht der Pfad vom Startzustand $q_0 = s$ zu einem Endzustand q_n , auf dem w gelesen wird, durch insgesamt $n+1$ Zustände. Das heißt, dass mindestens zwei Zustände q_i und q_j identisch sein müssen.



Das "Pumping Lemma,, [3]

Präzise Formulierung des Lemmas:

- Wenn L eine Sprache ist, die durch einen endlichen Automaten beschrieben werden kann ("reguläre Sprache"), dann gilt:
Wenn ein Wort $x \in L$ eine bestimmte Länge k erreicht oder überschreitet ($|x| \geq k$), dann läßt sich x so in drei Teile u , v und w zerlegen (mit $|v| \geq 1$), daß mit $x = uvw$ auch jedes $x' = uv^i w$ ($i=0$ oder $i>1$) Element von L ist.
- Um zu zeigen, dass eine Sprache L nicht regulär ist, genügt es, zu zeigen, dass L ausreichend lange Worte enthält. deren Teile nicht beliebig iterierbar sind.

Eine nicht reguläre Sprache [1]

Die Sprache $L = \{ a^n b^n \mid n \in \mathbb{N} \}$ (kurz: " $a^n b^n$ ") wird nicht von einem endlichen Automaten akzeptiert.

Beweis:

- Angenommen, die Sprache $L = \{ a^n b^n \mid n \in \mathbb{N} \}$ (kurz: " $a^n b^n$ ") wird von einem endlichen Automaten akzeptiert. Nach dem Pumping Lemma gibt es dann eine Zahl k , so daß für jedes Wort x mit $|x| \geq k$ eine Zerlegung in u , v und w möglich ist, so daß uw , $uvvw$, $uvvwv$, ... ebenfalls in L sind.
- Betrachten wir das Wort $a^k b^k$. Es gilt $|a^k b^k| \geq k$, das Wort muss also einen "duplizierbaren" Teil v besitzen. Um welchen Teil könnte es sich handeln?

Eine nicht reguläre Sprache [2]

- Drei Fälle sind denkbar:
 - Fall1: v liegt vollständig in der ersten Hälfte des Wortes, besteht also nur aus a 's. Dann müsste gelten, dass $uv^2w = a^{k+|v|}b^k \in L$: wegen $k+|v| \neq k$ unmöglich.
 - Fall2: v liegt vollständig in der zweiten Hälfte des Wortes, besteht also nur aus b 's. Dann müsste gelten, dass $uv^2w = a^kb^{k+|v|} \in L$: wegen $k+|v| \neq k$ unmöglich.
 - Fall 3: v überspannt die Mitte des Wortes, hat also die Form a^mb^m . Dann müsste gelten, dass $uv^2w = a^kb^ma^mb^k \in L$. Geht nicht, da a 's auf b 's folgen.
- Es gibt also für a^kb^k keine Zerlegung in uvw mit duplizierbarem Mittelteil. Also ist $L = a^n b^n$ nicht regulär.

Was steckt hinter dem Pumping Lemma?

Endliche Automaten haben eine fundamentale Einschränkung: Ihr „Gedächtnis“ ist endlich, durch die Anzahl ihrer Zustände beschränkt. Ein Automat mit k Zuständen kann sich nur an einen beschränkten Kontext „erinnern“: nach spätestens k gelesenen Symbolen ist er wieder in dem Zustand, den er vorher gehabt hat. Noch anders ausgedrückt: Der Automat kann nur bis k zählen. Um zum Beispiel $a^n b^n$ zu erkennen, müsste er sich aber beliebig lange Ketten von a 's merken können, weil er die Information anschließend beim Abarbeiten von b 's braucht.

Kontextfreie Grammatiken: Ein Beispiel

- $S \rightarrow aSb$, $S \rightarrow \varepsilon$ sind Produktionsregeln (Produktionen, Ersetzungsregeln)
- Beginne mit S (Startsymbol). Wenn in einer Kette das Symbol S vorkommt, ersetze es durch aSb (Regel 1) oder die leere Kette (Regel 2).
- Ersetze solange, bis nur noch a 's und b 's vorkommen („Terminalsymbole“)
- Eine gültige Ableitung der Beispielgrammatik:
 $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbbb$
- Die Beispielgrammatik erzeugt die Sprache $a^n b^n$

Kontextfreie Grammatik: Formale Definition

$G = \langle V, \Sigma, P, S \rangle$, wobei

- V nicht-leere Menge von Symbolen
- $\Sigma \subseteq V$ nicht-leere Menge von Terminalsymbolen
- $P \subseteq (V - \Sigma) \times V^*$ nicht-leere Menge von Produktionsregeln
- $S \in V - \Sigma$ das Startsymbol

Die Beispielgrammatik in alternativer Notation:

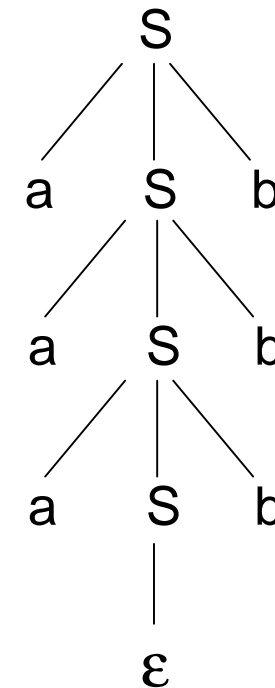
- $G1 = \langle \{a,b,S\}, \{a,b\}, \{ \langle S, aSb \rangle, \langle S, \varepsilon \rangle \}, S \rangle$
- Für $\langle A, \alpha \rangle \in P$ schreibt man üblicherweise $A \rightarrow \alpha$.
- Ableitungsschritt: Wenn w ein nicht-terminales Symbol A enthält und $A \rightarrow \alpha$ Produktion ist, ersetze A in w durch α .
- Die durch G erzeugte Sprache ist die Menge aller Worte über Σ^* , die sich aus S ableiten lassen.

Kontextfreie Grammatiken: Ein Beispiel

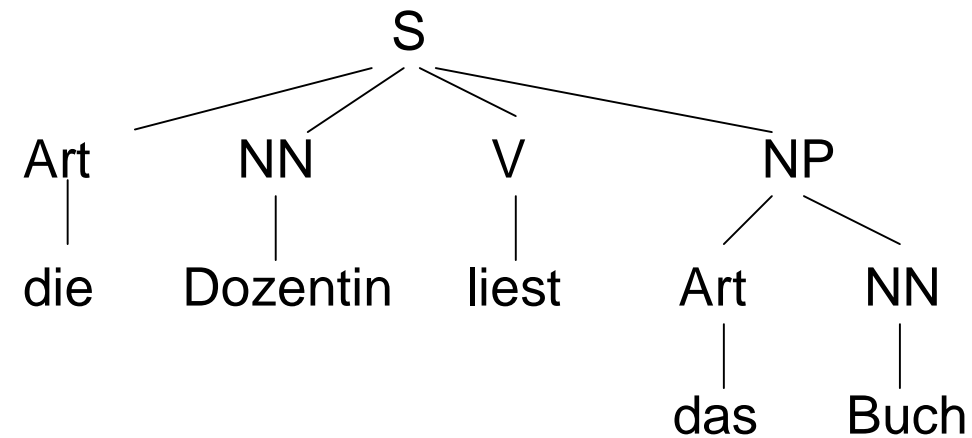
- $S \rightarrow aSb$, $S \rightarrow \varepsilon$ sind Produktionsregeln (Produktionen, Ersetzungsregeln)
- Beginne mit S (Startsymbol). Wenn in einer Kette das Symbol S vorkommt, ersetze es durch aSb (Regel 1) oder die leere Kette (Regel 2).
- Ersetze solange, bis nur noch a 's und b 's vorkommen („Terminalsymbole“)
- Eine gültige Ableitung der Beispielgrammatik:
 $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbbb$
- Die Beispielgrammatik erzeugt die Sprache $a^n b^n$

Kontextfreie Grammatiken: Ein Beispiel

- Das Wort aaabbb wird durch
 $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaasbbb \Rightarrow aaabbb$
abgeleitet.
- Die Ableitung kann durch den folgenden
Ableitungsbaum dargestellt werden:
- Die Blätter des Baums ergeben, von links nach
rechts gelesen, das abgeleitete Wort.
- Eine weitere alternative Schreibweise:
 $[_s a[_s a[_s a[_s \varepsilon] b] b] b]$
Kontextfreie Grammatiken ermöglichen die
Darstellung beliebig tief geschachtelter
Strukturen.
- „Ableitungsbaum“ heißt auch „Parse-Tree“:
Parsing ist automatische syntaktische Analyse



Kontextfreie Grammatik: Der alte Beispielsatz



Geschachtelte Strukturen in natürlicher Sprache

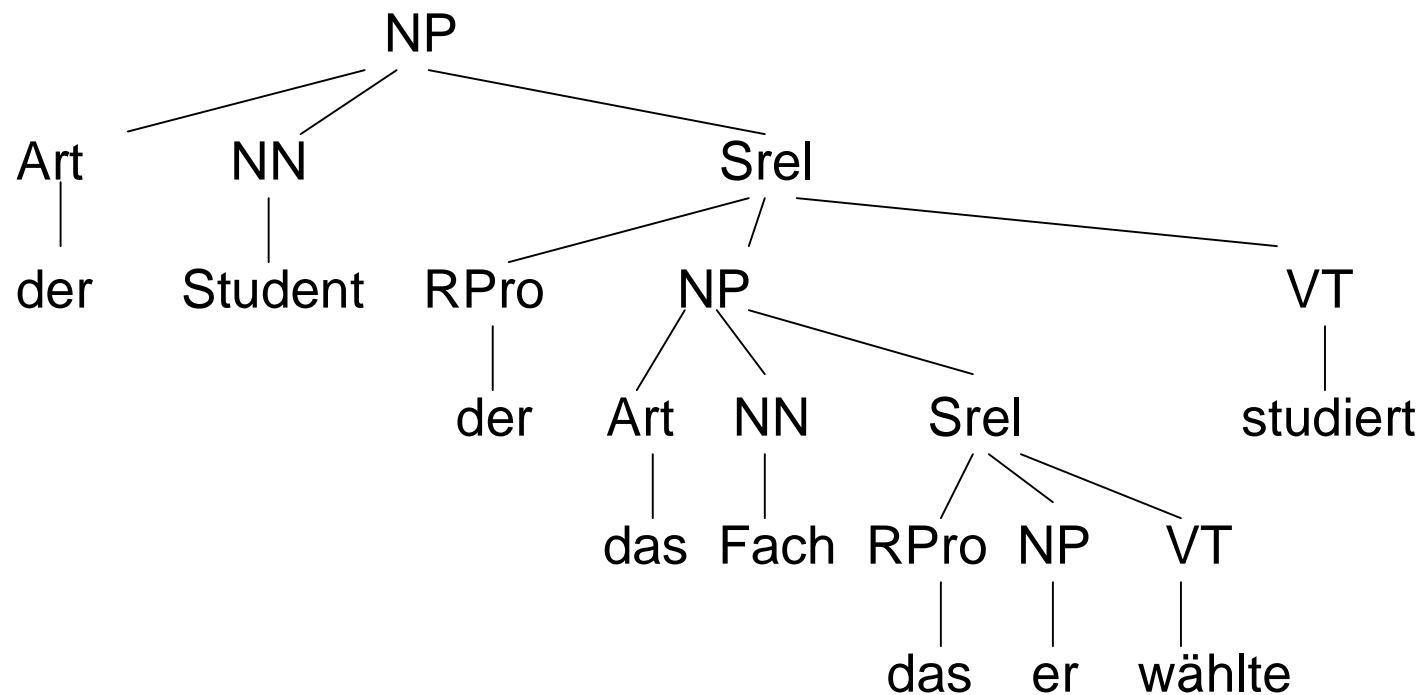
- Der Nominalausdruck „*der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, der im Hauptfach, für das er sich nach langer Überlegung entschieden hat, Informatik studiert*“ enthält
- den (Relativ-)Satz „*der im Hauptfach, für das er sich nach langer Überlegung entschieden hat, Informatik studiert*“; der enthält
- den Nominalausdruck „*Hauptfach, für das er sich nach langer Überlegung entschieden hat*“; der enthält
- den (Relativ-)Satz „*für das er sich nach langer Überlegung entschieden hat*“; der enthält
- unter anderem den Nominalausdruck „*er*“.

Geschachtelte Strukturen in natürlicher Sprache

*[_{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]]*

Geschachtelte Strukturen in natürlicher Sprache

[_{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]]



Konstituentenstruktur

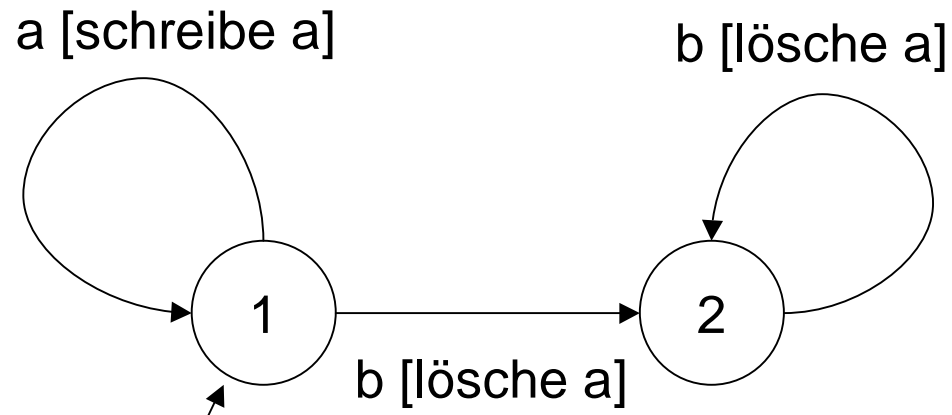
- Die Erzeugung eines Satzes mit einer kontextfreien Grammatik (und die Verarbeitung des Satzes mit einem Parser, der die Grammatik in ein Analysesystem umsetzt) liefert nicht nur die Information, ob der Satz zur Sprache gehört/"grammatisch ist". Als "Seiteneffekt" liefert sie den Ableitungsbaum/Parse Tree, der die "Konstituentenstruktur" und damit einen wesentlichen teil der syntaktischen Struktur beschreibt.
- Eine Grammatik ist eine angemessene Beschreibung der syntaktischen Struktur einer Sprache, wenn sie
 - genau die grammatisch akzeptablen Sätze der Sprache erzeugt
 - für die grammatischen Sätze plausible Konstituentenstrukturen erzeugt.

Syntaktische Verarbeitung

- Kontextfreie Grammatiken erlauben die einfache und elegante Beschreibung syntaktischer Strukturen (ähnlich wie Zustandsdiagramme/deterministische endliche Automaten das für morphologische Strukturen tun).
- Wie sehen Analysesysteme für kontextfreie Sprachen aus (syntaktische Parser)?
- Die einfachste Version ist der "Kellerautomat": Wir nehmen einen Automaten an, der zusätzlich zu Zuständen und Übergängen über einen Speicher als Stapel (Stack, Keller) verfügt. Der Automat liest Symbole der Eingabekette und kann bei jedem Übergang Symbole in den Stack schreiben oder vom Stack nehmen. Eine Eingabe wird akzeptiert, wenn ein Endzustand erreicht, das Eingabewort abgearbeitet, und der Stack leer ist.

Ein Beispiel

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



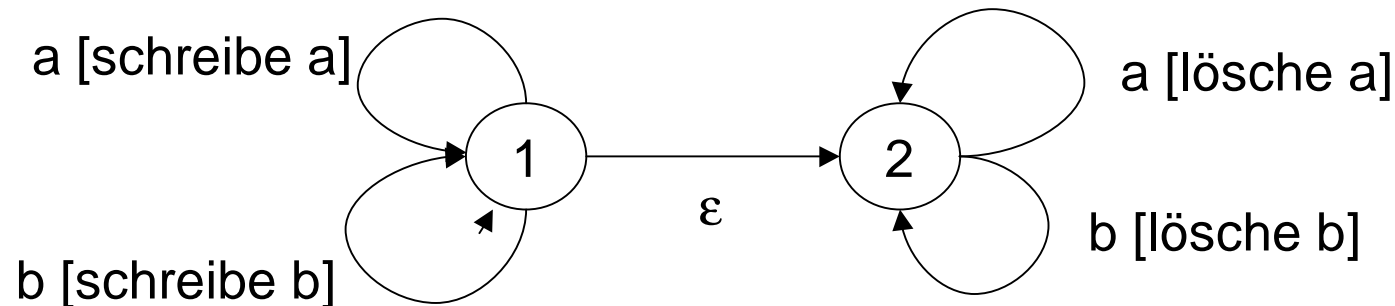
- Im Zustand 1 werden a's gelesen und in den Stack geschrieben. Beim ersten b wechselt der Automat in den Zustand 2 und löscht für jedes gelesene b ein a vom Stack. Eingabe und Stack sind beide leer, wenn die Eingabe gleichviele a's und b's enthielt. (Beide Zustände sind Endzustände.)

Deterministische syntaktische Verarbeitung?

- Der Automat aus Beispiel 1 ist deterministisch: Er führt in jedem Zustand bei jedem Eingabesymbol eine bestimmte Aktion aus, benötigt deshalb keine aufwändigen Suchverfahren und analysiert jede Eingabe in linearer Zeit, ganz wie ein DEA,
- Gibt es eine generelle Möglichkeit, kontextfreie Grammatiken (analog zum NEA-DEA-Überführungsalgorithmus) in ein Format zu übersetzen, das deterministisches Parsing erlaubt?
- Die Antwort ist Nein. Viele kontextfreie Sprachen sind inhärent nicht-deterministisch (dazu auch das Beispiel auf der nächsten Folie).

Ein einfaches Beispiel

- Die Sprache $L = \{ww^R \mid w, w^R \in \{a,b\}^*\}$ kann nicht deterministisch geparkt werden (w^R ist die Spiegelung von w).
- Auch dieser Automat hat zwei Zustände. Bis zur Mitte des Eingabewortes speichert er a's und b's im Stack, dann springt er in den zweiten Zustand, in dem er neu gelesene Symbole mit den gespeicherten Stack-Symbolen vergleicht.



- Problem: Woher weiß der Automat, dass er die Mitte des Wortes erreicht hat. Er muss raten, und wenn er falsch geraten hat, zurücksetzen und eine andere Option verfolgen.

Verarbeitung von CFGs für natürliche Sprachen

- Programmiersprachen, Sprachen der Logik und Mathematik sind meist so konzipiert, dass sie deterministische Analyse zulassen; eine einfache Möglichkeit, Nicht-Determinismus zu vermeiden, besteht in der Verwendung von Klammern.
- Natürliche Sprachen lassen keine deterministische Analyse zu, sie erfordern nicht-lineare Suchverfahren / Parsingalgorithmen, die im besten Fall quadratische Zeit erfordern (d.h., der Zeitbedarf für eine Eingabe der Länge n beträgt $c \cdot n^2$; c ist ein konstanter Faktor).

Konstituentenstruktur

- Die Erzeugung eines Satzes mit einer kontextfreien Grammatik (und die Verarbeitung des Satzes mit einem Parser, der die Grammatik in ein Analysesystem umsetzt) liefert nicht nur die Information, ob der Satz zur Sprache gehört/"grammatisch ist". Als "Seiteneffekt" liefert sie den Ableitungsbaum/"Parse Tree", der die "Konstituentenstruktur" und damit einen wesentlichen teil der syntaktischen Struktur beschreibt.
- Eine Grammatik ist eine angemessene Beschreibung der syntaktischen Struktur einer Sprache, wenn sie
 - genau die grammatisch akzeptablen Sätze der Sprache erzeugt
 - für die grammatischen Sätze plausible Konstituentenstrukturen erzeugt.

Grammatische Mehrdeutigkeit

- Natürlich-sprachliche Sätze sind oft grammatisch mehrdeutig. Die grammatische Mehrdeutigkeit ist relevant. Sie führt im allgemeinen zu semantischer Mehrdeutigkeit.
- Natürliche Sprachen benötigen deshalb kontextfreie Grammatiken, die Sätzen unterschiedliche syntaktische Strukturen zuweisen können (und deshalb übrigens per se nicht-deterministisch sind).

Grammatische Mehrdeutigkeit, Beispiele 1

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

$S \rightarrow NP VI$

$S \rightarrow NP VT NP$

$S \rightarrow S PP$

$NP \rightarrow ART NN$

$NP \rightarrow PN$

$NP \rightarrow NP PP$

$PP \rightarrow P NP$

- Der Satz: *Peter sah den Mann mit dem Teleskop* hat in dieser Grammatik zwei Analysen, die zwei Bedeutungsvarianten entsprechen:
 - $[_S \text{Peter} [_{VT} \text{sah} [_{NP} [_{NP} \text{den Mann}] [_{PP} \text{mit dem Teleskop}]]]]$
 - $[_S [_S \text{Peter} [_{VT} \text{sah} [_{NP} \text{den Mann}]]] [_{PP} \text{mit dem Teleskop}]]$
- Der Satz *Peter rief den Kollegen aus München mit dem Handy an* hat mindestens 4 Lesarten (welche?).

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*

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

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

Stand der Technik in der grammatischen Verarbeitung

- Im Prinzip gelöst durch:
Kontextfreie Grammatiken und Parsingtechniken + Unifikation
aber:
- Grammatiken sind unvollständig:
Es gibt zur Zeit keine Grammatik für eine natürliche Sprache, die die
möglichen Satzkonstruktionen hinreichend vollständig abdeckt.
- Vollständige grammatische Verarbeitung ist oft zu langsam.
- Alternative für bestimmte Zwecke:
Partielle/flache/statistische grammatische Analyse